# ARRL Periodicals Archive – Search Results

## A membership benefit of ARRL and the ARRL Technical Information Service

**ARRL Members:** You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

**Need a higher quality reprint or scan?** Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

**QST on CD-ROM**: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

**Non-Members:** Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.

### Copyright/Reprint Notice

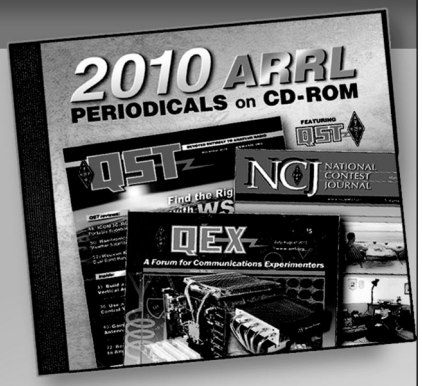
In general, all ARRL content is copyrighted. ARRL articles, pages, or documents--printed and online--are not in the public domain. Therefore, they may not be freely distributed or copied. Additionally, no part of this document may be copied, sold to third parties, or otherwise commercially exploited without the explicit prior written consent of ARRL. You cannot post this document to a Web site or otherwise distribute it to others through any electronic medium.

For permission to quote or reprint material from ARRL, send a request including the issue date, a description of the material requested, and a description of where you intend to use the reprinted material to the ARRL Editorial & Production Department: permission@arrl.org.

**QST Issue:** Feb 1981
**Title:** Add-Ons For Greater Dipper Versatility
**Author:** Robert Johns, W3JIP

Click Here to Report a Problem with this File

ADVERTISEMENT




**2010 ARRL Periodicals**
on CD-ROM

**ARRL's popular journals are available on a compact, fully-searchable CD-ROM. Every word and photo published throughout 2010 is included!**

- ***QST*** **The official membership journal of ARRL**
- ***NCJ*** **National Contest Journal**
- ***QEX*** **Forum for Communications Experimenters**

**SEARCH** the full text of every article by entering titles, call signs, names—almost any word. **SEE** every word, photo (including color images), drawing and table in technical and general-interest features, columns and product reviews, plus all advertisements. **PRINT** what you see, or copy it into other applications.

**System Requirements**: Microsoft Windows™ and Macintosh systems, using the industry standard Adobe® Acrobat® Reader® software. The Acrobat Reader is a free download at **www.adobe.com**.

**2010 ARRL Periodicals on CD-ROM**

ARRL Order No. 2001
**Only $24.95***
*plus shipping and handling

Additional sets available:
2009 Ed., ARRL Order No. 1486, $24.95
2008 Ed., ARRL Order No. 9406, $24.95
2007 Ed., ARRL Order No. 1204, $19.95
2006 Ed., ARRL Order No. 9841, $19.95
2005 Ed., ARRL Order No. 9574, $19.95
2004 Ed., ARRL Order No. 9396, $19.95
2003 Ed., ARRL Order No. 9124, $19.95
2002 Ed., ARRL Order No. 8802, $19.95
2001 Ed., ARRL Order No. 8632, $19.95

ARRL The national association for AMATEUR RADIO™
SHOP DIRECT *or call for a dealer near you.*
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)

Feb 1981 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

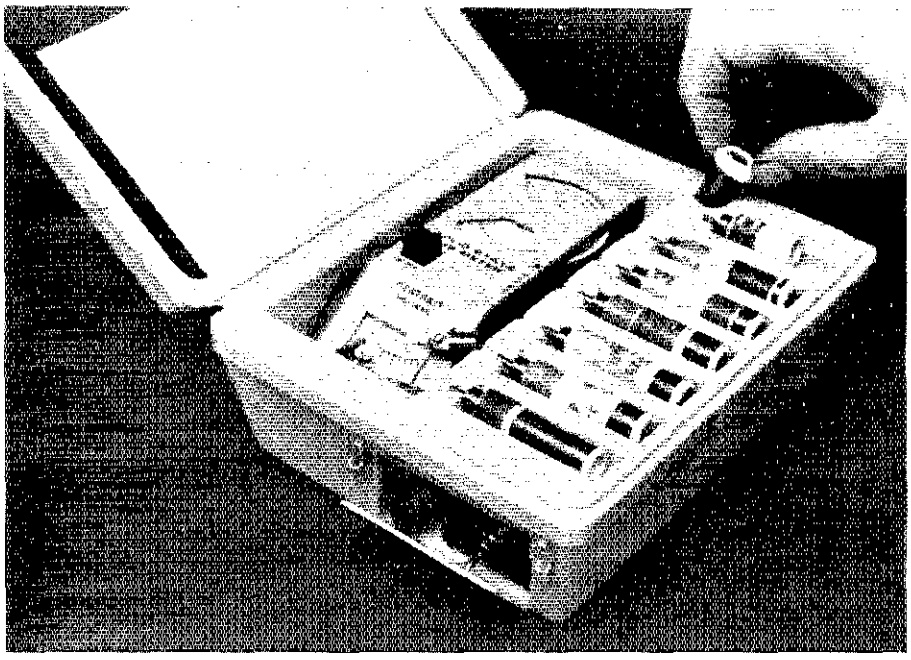
# Add-Ons for Greater Dipper Versatility

## Your dip oscillator may be in for a new ball game! These simple gimmicks and gadgets extend its usefulness and improve dial accuracy.

By Robert H. Johns,* W3JIP

Simple accessories can be added to your dip oscillator for increased usefulness. They do not require any modification of the basic instrument. All are "outboard additions" that complement the original equipment by providing improved dial accuracy and a means for capacitance coupling to inaccessible circuits. Furthermore, the add-ons make tuning easier by slowing the dipper tuning rate, an advantage when the device is performing as a signal generator. Another benefit is provision of convenient connections and components for measuring inductance and capacitance.

The accessories described apply directly to the Heathkit HD-1250 dip meter. The ideas, however, may be applied to other models, too.



The case of the Heath HD-1250 dip meter has plenty of extra space for accessories that belong with the dipper. A photocopy of the *Handbook* "L and C vs. frequency" graph is taped to the lid. Loose wire additions such as the capacitance probe and the phono jack are tucked in the instrument compartment. The door cut in the side permits access to the space under the coil tray for the plastic box and other coils. Solder lugs held by small hardware hold the door closed. The 5-μH standard coil has a plastic hat, white silicone rubber caulking compound, which prevents it from slipping through the mounting hole to the right of the dipper coils.

### Accuracy

I found the dial calibration surprisingly accurate, especially for a kit. Dipper scales are never exact, and no one expects them to be. As I prepared a chart of how many kilohertz to add or subtract from the dial indication, I noticed that each amateur band required the use of a different coil except for 15 and 20 meters. The thought came to mind, "Why not adjust each coil so that it is 'right on' for the bands where the dipper is used most?" Not wishing to modify the coils themselves (they are nicely encased in plastic), I trimmed the inductance of each by the addition of tiny amounts of core material, either powdered ferrite or chips of copper.

First, I listened to the dipper on a calibrated receiver to determine the correction needed. If the dipper indicated 6.9 MHz when the receiver is getting the signal at 7.0 MHz, the coil is too small (too much C needed to resonate it at 7.0 MHz). This means a little more inductance needs to be added, which can be done by inserting pieces of ferrite from a junk-box coil. See Fig. 1. Use a hammer or pliers to break up the ferrite. Before adding the ferrite powder, plug the coil with some sort of plastic that will harden. Fill to a depth of 10 mm (0.4 inch) inside the coil form. Silicone rubber caulking or bathtub sealant has good electrical properties. Either makes a good embedding medium for the ferrite and can be dug out of the coil form if you later decide to remove it. As you add powder and little bits of ferrite, they stick in the silicone. You can follow the frequency change with your receiver and add inductance until both read the same frequency. Make sure that the dipper coil is plugged all the way into its socket since the capacitance involved at this point is part of the tuned oscillating circuit. Otherwise, a little slip will throw off the exact calibration you are aiming for.

Should the coil be too large (the dial

*R. H. Johns-Scientific Instruments, 3379 Paper Mill Rd., Huntingdon Valley, PA 19006

Feb 1981 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

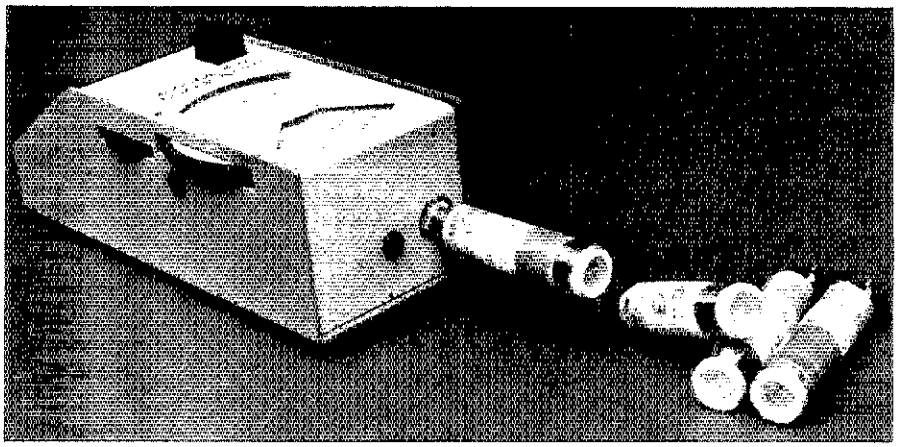


Fig. 1 — The Heath dip meter with plug-in coils. The silicone rubber that may be seen inside the coil has ferrite dust or copper chips embedded in it to trim the coil's inductance so that the dipper reads correct frequencies on amateur bands.

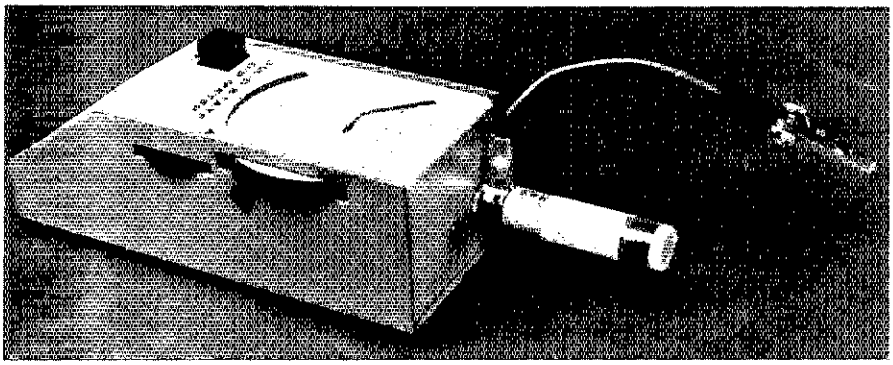


Fig. 2 — The capacitance coupling probe. The clip lead is connected to the circuit under test and is coupled to the dip meter through the small capacitance between the insulated collar and the phono plug of the coil.

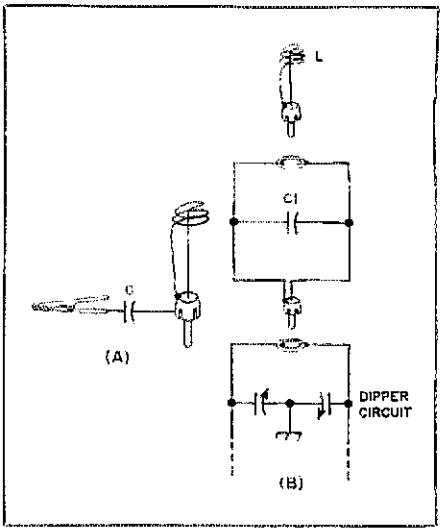



Fig. 3 — At A is shown the coupling capacitor, C, connected between the clip and the coil. Actually, this capacitor is simply the capacitance between the phono plug outer sleeve and the aluminum collar that fits around it but doesn't touch it. At B, a 150-pF or 100-pF capacitor, C1, is connected in parallel with the plug-in coil, L, and also the balanced tuning capacitors inside the dip meter. The capacitors are mounted in a plastic box and connections made by plugging the box in between the coil and the dipper.

indicating 7.1 MHz when the receiver is set at 7.0 MHz), the inductance may be decreased by placing small chips of copper in the coil. Each one acts as a shorted turn of a transformer secondary where the dipper coil is the primary. Cut pieces about a millimeter (0.04 inch) long from heavy copper wire and add them a few at a time to the silicone plug in the coil form. When enough core material has been added to a coil, squeeze some additional silicone into the coil form to cover and secure it. The silicone will set overnight and is easier to trim with a sharp knife than by trying to smooth out the uncured sticky stuff.

There is a little compromise concerning the 15/20-meter coil. That compromise in calibration amounts to the width of a calibration line. In other words, the calibration is about 1 mm (0.04 inch) off. That's very good, however.

### Capacitance Coupling

After reading the recent article by Fred Brown, W6HPH,[1] I thought that I would try capacitance coupling to an unknown circuit as he describes. I agree with his conclusion that it works very well. I'm sorry that I did not know about this years ago! With capacitance coupling, you can dip an unknown circuit without having to place the dipper physically next to it so that the coils couple magnetically. The unknown can be a toroid or an LC circuit below a chassis or in a tight place. A single wire is clipped to the unknown and then lightly coupled (a few picofarads) to one side of the dipper tank circuit. A "no-holes" modification for the Heath dipper is accomplished by wrapping a turn or two of the insulated single wire around the outer surface of the phono plug of the dipper coil. This outer surface of the phono connector is not a ground. Rather, it is one side of the balanced oscillator circuit.

A more stable capacitance probe is shown in Fig. 2. An aluminum ring is bent and shaped to be a snug slip fit around the phono plug. The single wire is connected and black electrical tape wrapped around the ring for insulation. The capacitance between this ring and the phono plug is enough to couple the unknown circuit to the dipper so that a nice dip is observed at resonance. The ring may be cut and bent from aluminum sheeting, but I made the one shown from aluminum tape of the type sold in automotive stores. It can be cut with scissors, works easily and can be folded into layers to become as thick as desired. There are many uses for it in the ham shack.

This capacitance probe also provides uniform coupling to a circuit when measuring its Q, as suggested in the Heath manual. In brief, one measures the rf voltage across the test circuit and notes the change in frequency between the 3-dB-down points, where the voltage falls to 0.707 times the peak voltage. The Q of the unknown circuit is equal to the center frequency divided by the bandwidth. This measurement of Q with a dipper was new to me. It works out very well, provided that the coupling between the dipper and the unknown circuit is kept constant. The bandspread provided by the next circuit addition is also a help in measuring the 3-dB bandwidth since these frequencies are quite close together for high-Q circuits in the hf range.

### Signal Generator

Although use of a dip meter as a signal source for an impedance bridge or as a signal generator in receiver work is common, the fast tuning rate is a handicap. It is tricky to get a dipper to stop in the passband of a receiver. Since the amateur bands are at the low end of the tuning ranges of the Heath dipper, a simple way to slow down the tuning rate is to add a capacitor in parallel with the tuning capacitor in the dip oscillator as shown in Fig. 3. No change need be made to either the dipper or the coil if the capacitor is placed between them. Placement must allow the coil to be plugged into the capacitor box and the capacitor box plugged into the instrument. With the 150-pF

[1] Brown "A 1980 Dipper," *QST*, March 1980, p. 11.

Feb 1981 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

padder in the circuit, the normal tuning range of the dipper from about 15 pF to 70 pF becomes about 165 to 220 pF. Accordingly, the frequency of the oscillator is both lower for a given coil and does not change over such a wide range. The green coil of the Heath dipper, which tunes from 12.5 to 26 MHz, oscillates from 6.7 to 7.7 MHz with the 150-pF capacitor in parallel with it. This capacitor serves to spread out the bands, 160 through 10 meters. The 50-pF capacitor gives 15- and 17-meter coverage with one coil and 30-meter coverage with another.

Capacitors, phono plugs and jacks are mounted in a small plastic box with the bandspread calibrations shown on the outside (Fig. 4). The calibrations are marked on colored tapes (Radio Shack 64-2340) that correspond to the colors of the coils. Which way to plug the box in is taken care of by the calibration scales; when you read the 40-meter scale, the box is plugged in with the proper capacitor connected.

The oscillator stalls at the high ends of the 160- and 80-meter ranges, probably from lack of feedback. This is easy to recognize, for the meter drops to zero and restricts the ranges only a little.

With the modification indicated, the dipper signal is easy to tune in on a receiver. Also, the $\Delta f$ for Q measurements in or near the ham bands is easier to estimate from the bandspread scales, despite the inaccuracies in reading the calibration through the logging scale. The measurement of the rf voltage across the unknown circuit is made easier by using a digital voltmeter and a simple probe. The 0.1 millivolt sensitivity of these instruments makes this job a snap!

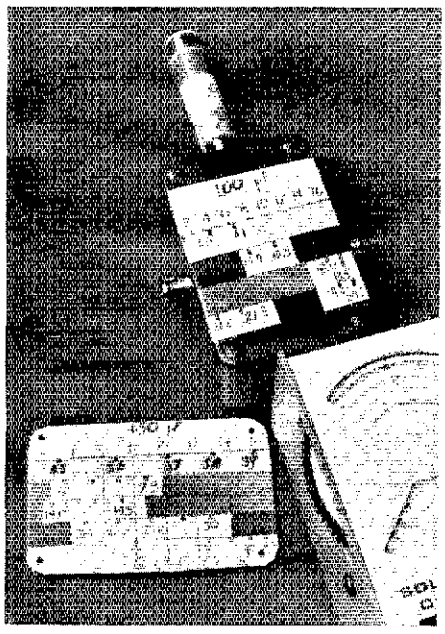


Fig. 4 — The 100-pF bandspread capacitor is connected between the coil and the dipper and the bandspread scales are taped on the plastic box. To use the 150-pF capacitor, the box is turned over and rotated 90°, and the other set of phono connectors is used.

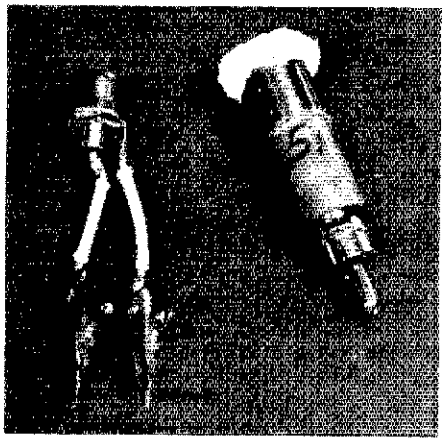


Fig. 5 — Accessories for measuring inductance and capacitance. The phono jack with clip leads will connect to the 100-pF standard in the plastic box and clip onto the unknown inductance. After locating the resonant frequency with the dipper, the value of the unknown coil may be read from charts in the ARRL *Handbook*. The standard 5-μH coil is about the same size as the dipper coils and can be plugged into the phono jack for connection to an unknown capacitor.

### Measuring Unknown Coils and Capacitors

The box with a 50-pF capacitor inside fits right in with the graphs and the methods outlined in the measurements chapter in the ARRL *Handbook*. Unknown coils may be measured by connecting them to the 50-pF capacitor in the box via a phono jack equipped with short clip leads as in Fig. 5 and the resonant frequency of the combination found with a dipper. The graph of inductance vs. frequency for this capacitor will furnish the inductance of the unknown coil with no calculation.

Any of the dipper coils can be used as a standard in finding the capacitance of unknown capacitors; their inductance is given in the manual. This can get confusing, since the standard coil that is plugged into the phono jack and connected to the unknown capacitor is sometimes needed in the dipper to scan for the resonant frequency. I prefer to make up a small 5-μH standard that can be stored in the plastic dipper box along with other accessories. It could have been put in the plastic box, but I was afraid of stray coupling and false resonance. The small coil in Fig. 5 is wound on a length of half-inch PVC tubing which is obtainable from the plumbing department of many hardware stores. A good phono plug (Radio Shack no. 274-339) fits snugly in this tubing.

It should be clear that any of these additions to your dip meter could be made independently of the others and according to your interests and needs. The dipper is one of the most useful devices in the shack and it can become even more versatile! QST

# Strays

### TA PROFILES

☐ We are pleased to have Jim Stewart, WA4MVI on our team of ARRL Technical Advisors, serving as our radio propagation/predictions and EME specialist. In 1978 Jim had a book published on this subject, entitled *VHF Radio Propagation*.

First licensed at age 15, Jim presently has an Extra Class license, and is the proud holder of a WAS certificate on 6 and 2 meters, plus a WAC certificate on 2 meters. His principal interests in Amateur Radio include EME, vhf/uhf, propagation, radio astronomy and OSCAR programs. He is also involved in flying, skiing and sports cars.

As a chemistry major, Jim received his BS degree from Lander College. He resides in Hendersonville, North Carolina, and is employed in air traffic control and as a commercial pilot. — *Marion Anderson, WB1FSB*



TA Jim Stewart, WA4MVI

### HAPPY BIRTHDAY AMSAT-OSCAR 8!

☐ March 5 marks the completion of three years of successful operation for AMSAT-OSCAR 8. The satellite will have provided flawless service to Amateur Radio operators, students and science teachers worldwide during 15,280 orbits.

To commemorate this anniversary, ARRL will issue a special QSL card. During the period March 1 through 7, send a signal-reception report to AMSAT-OSCAR 8 Third Anniversary, ARRL Club and Training Dept., 225 Main St., Newington, CT 06111. Provide as much information as possible, including calls heard, telemetry and frequency. Check the OSCAR Operating Schedule elsewhere in this issue or write ARRL Hq. for details on how to listen for AMSAT-OSCAR 8. — *Bernie Glassmeyer, W9KDR, ARRL Satellite Coordinator*

# ARRL Periodicals Archive – Search Results

## A membership benefit of ARRL and the ARRL Technical Information Service

**ARRL Members:** You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

**Need a higher quality reprint or scan?** Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

**QST on CD-ROM**: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

**Non-Members:** Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.

### Copyright/Reprint Notice

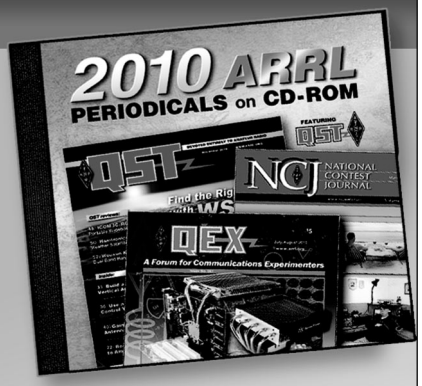
In general, all ARRL content is copyrighted. ARRL articles, pages, or documents--printed and online--are not in the public domain. Therefore, they may not be freely distributed or copied. Additionally, no part of this document may be copied, sold to third parties, or otherwise commercially exploited without the explicit prior written consent of ARRL. You cannot post this document to a Web site or otherwise distribute it to others through any electronic medium.

For permission to quote or reprint material from ARRL, send a request including the issue date, a description of the material requested, and a description of where you intend to use the reprinted material to the ARRL Editorial & Production Department: permission@arrl.org.

**QST Issue:** Jul 1993
**Title:** Updated Tandem Match, An
**Author:** John Grebenkemper, KI6WX

Click Here to Report a Problem with this File

ADVERTISEMENT




**2010 ARRL Periodicals**
on CD-ROM

**ARRL's popular journals are available on a compact, fully-searchable CD-ROM. Every word and photo published throughout 2010 is included!**

- ***QST*** **The official membership journal of ARRL**
- ***NCJ*** **National Contest Journal**
- ***QEX*** **Forum for Communications Experimenters**

**SEARCH** the full text of every article by entering titles, call signs, names—almost any word. **SEE** every word, photo (including color images), drawing and table in technical and general-interest features, columns and product reviews, plus all advertisements. **PRINT** what you see, or copy it into other applications.

**System Requirements**: Microsoft Windows™ and Macintosh systems, using the industry standard Adobe® Acrobat® Reader® software. The Acrobat Reader is a free download at **www.adobe.com**.

**2010 ARRL Periodicals on CD-ROM**

ARRL Order No. 2001
**Only $24.95***
*plus shipping and handling

Additional sets available:
2009 Ed., ARRL Order No. 1486, $24.95
2008 Ed., ARRL Order No. 9406, $24.95
2007 Ed., ARRL Order No. 1204, $19.95
2006 Ed., ARRL Order No. 9841, $19.95
2005 Ed., ARRL Order No. 9574, $19.95
2004 Ed., ARRL Order No. 9396, $19.95
2003 Ed., ARRL Order No. 9124, $19.95
2002 Ed., ARRL Order No. 8802, $19.95
2001 Ed., ARRL Order No. 8632, $19.95

ARRL The national association for AMATEUR RADIO™
SHOP DIRECT *or call for a dealer near you.*
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)

Jul 1993 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

# Technical Correspondence

Conducted By Paul Pagel, N1FB
Associate Technical Editor

The publishers of *QST* assume no responsibility for statements made herein by correspondents.

## AN UPDATED TANDEM MATCH

◊ The Tandem Match[1] is a directional wattmeter as accurate as any commercial unit over the 1.8- to 54-MHz frequency range. It also provides a direct linear readout of SWR and power. Judging by the mail I've received since the original unit appeared, the unit continues to be a popular construction project despite the lack of a PC board. That lack exists no longer: A PC board is available from FAR Circuits.[2]

Several corrections to the original article appear in January 1988 *QST*. The project also appears in the 16th edition of the *ARRL Antenna Book*. The schematic in that book contains an error; there should be a 0.01-µF capacitor, not a wire, between the center pin of J1 and D6 and D7. The FAR Circuits PC board includes several minor modifications to the original circuit. An updated schematic is available from me or the ARRL Technical Department Secretary for a business-size SASE, and is provided by FAR Circuits with each PC board.

The primary revision to the original circuit is support of op amps other than the TLC27x4 family used originally. To use the circuit as it was originally designed (with the 9-V batteries and TLC27L4 or TLC27M4s), jumpers are connected at W1, W2 and W3. By omitting these jumpers, any quad FET-input op amp can be used in lieu of the TLC27x4s. Possible substitutes include the TL064, TL074, TL084, LF347, and LF444. These op amps probably won't work in the original circuit because of their higher current drain and their different input- and output-voltage ranges. Omitting W1, W2 and W3 disables the automatic turn-on circuit, and the circuitry associated with Q1, Q2 and Q3 need not be built. Instead, a +15- to +18-volt supply is connected to V+ and a –5- to –15-volt supply is connected to V–. If you want to use this unit to measure power levels greater than 1 kW, V+ must be 18 volts.[3] The V+ and V– supplies could be derived from an external ac-line-operated power supply or three 9-volt batteries.

The original circuit specified use of TLC27L4 or TLC27M4 op amps. I had a great deal of difficulty with circuit instability when I used a TLC27L4 at U1, especially when RF is applied to the input. The problem may be related to the very low supply current (10 µA) required by this op amp. I recommend building the circuit using TLC27M4s or TLC274s, which use a much higher supply current. This increases the overall current drain to 3.8 mA, or 17 mA, respectively.

Other circuit changes are minor. The feedback resistors around the 1N5711 diodes are now 100 kΩ instead of 180 kΩ and 1 MΩ. The feedback capacitors from the op-amp output to inverting input have been increased in value from 0.001 µF to 0.01 µF. Several additional 470-pF bypass capacitors have been added to the circuit. Please note: I don't recommend using plastic-case 2N2222/2N2907 transistors at Q1, Q2 or Q3. Their higher leakage current could cause false triggering of the automatic turn-on circuit.

Radio Shack no longer sells the 50-µA meter specified in the original article. They do, however, offer a 0- to 15-volt dc meter (#270-1754) which, in reality, is a 1-mA full-scale meter movement packaged with a 15-kΩ series resistor. This meter can be used in the circuit by reducing the value of the resistor in series with the meter from 100 kΩ to 5 kΩ. The 1N34 diodes are available at Radio Shack (#276-1123); 1N271s are almost identical and can also be used. Mount the 1N34 diodes slightly above the circuit board so that the heat from soldering does not cause a diode failure. (Use a clip-on heat sink [such as Radio Shack's #276-1567], a pair of pliers or a hemostat on the diode lead above the board to draw the heat from the diode body.—*Ed.*)

The Tandem Match PC board is double-sided with plated-through holes. The upper (component) side has mainly the chassis and circuit ground planes, although there are a few signal traces. Check the board carefully to ensure that none of the ground planes pass too close to a circuit lead. I found it necessary to scrape away a bit of foil from a few places around the IC holes; this is easy to do with an X-ACTO knife. All of the trimmer pots must be square multiturn units with top adjustment screws.

The PC board accepts ¼-watt resistors and radial-lead capacitors. Mount the ferrite beads so that they don't touch any PC board traces; the beads have sufficient leakage to cause problems in the high-impedance portions of this circuit. Before mounting the SO-239 connectors on the PC board, enlarge the location holes to ⅝" diameter. The components connected to the SO-239s are soldered directly between the center pin and the board traces. Be sure to follow the instructions in the original article (or the *ARRL Antenna Book*) as to the mounting of the four 200-Ω resistors that serve as a 50-Ω termination resistor, and the 1N5711 detector diodes. I soldered the ground end of the 200-Ω resistors directly to the chassis ground plane. Despite what the original article says, I've found that the detector diodes need *not* be a matched pair.

*You must enclose the PC board within a metal cabinet.* Otherwise, RF can get into the circuit and cause problems. These problems usually manifest themselves as an SWR reading of less than 1, or a negative voltage at TP3 or TP6. This problem can also occur if you have a large common-mode current on the coaxial cable (current flowing on the coax shield). A solid metal enclosure makes the circuit less susceptible to RFI.

Parts can be obtained through various distributors including Digi-Key, Surplus Sales of Nebraska, Active Components, and Newark Electronics (see the ARRL Parts Suppliers List in any recent *Handbook*). I wasn't able to find a single distributor that stocks all of the parts. So, I contacted a local distributor, Anchor Electronics,[4] owned and operated by John Burgoon, AB6JQ. John has agreed to stock the semiconductor parts used in the revised Tandem Match, including the TLC27M4CN, 1N5711, LM336-2.5 and CA3146E. Anchor also stocks the other parts, including the trimmer pots and diodes.—*John Grebenkemper, KI6WX, ARRL Technical Advisor, 19490 Miller Ct, Saratoga, CA 95070.*

[1] J. Grebenkemper, "The Tandem Match," *QST*, Jan 1987, pp 18-26. Also, see Feedback, *QST*, Jan 1988, p 49. The Tandem Match also appears in various editions of the *Handbook* and *The ARRL Antenna Book*.

[2] A PC board is available from FAR Circuits, 18 N 640 Field Court, Dundee, IL 60118; price, $19 plus $1.50 shipping. PC-board template packages are available from FAR Circuits.

[3] The amount of power you can measure is dependent on the common-mode range of the op amp. Let's assume the common mode is 3 volts below $V_{CC}$. This means we can have an input voltage of 15 volts, that is equivalent of 2.25 watts being fed into the detector (which is slightly more than the 2 watts dissipation of the 50-Ω termination resistor). With a 30-dB coupler, this corresponds to 2.25 kW through the wattmeter. If the supply voltage is 15, we're limited to measuring 1.4 kW with this same common-mode range.

[4] Anchor Electronics, 2040 Walsh Ave, Santa Clara, CA 95050, tel: 408-727-3693, fax: 408-727-4424. Minimum order: $20.

**Note:** All correspondence addressed to the Technical Correspondence column should bear the name, call sign and complete address of the sender. Please include a daytime telephone number at which you can be reached if necessary.

Keep the author(s) in the communications loop. Whether praising or criticizing a work, copy the author(s) on comments sent to Technical Correspondence. QST

# Feedback

◊ Refer to "Sporadic-E Causes," *QST*, Technical Correspondence, April 1993, page 73. In the second paragraph, fourth sentence, insert the words "peaks and" between the words "whose" and "troughs" so that the partial sentence reads: "...over at least a decade and whose peaks and troughs were closely related...—*Neil Spokes, AB4YK*

# Test Procedures Manual

By Bob Allison, WB1GCM, ARRL Test Engineer
Michael Tracy, KC1SX, ARRL Technical Advisor
Mike Gruber, W1MG, ARRL EMC/RFI Engineer

**Revision history:**
Initial document: September 1990
Revision A, October 1992, updates and corrections
Revision B, January 1996, major revision, additional tests added
Revision C, April 1997, updates and corrections
Revision D, November 1997, new graphics, updates and corrections
Revision E, August 1998, expanded test information added
Revision F, June 2000, BER test box changes added
Revision G, January 2002, new schematics added to Appendix A
Revision H, June 2004, updates for new test equipment added
Revision I, January 2009, major revision, additional tests added
Revision J, June 2009, updates and corrections
Revision K, January 2010 updates and corrections

American Radio Relay League, Inc., 1990-2010

## TABLE OF CONTENTS


| | |
|---|---|
| I. PURPOSE AND SCOPE | 3 |
| II. LIST OF FIGURES | 4 |
| III. LIST OF REQUIRED TEST INSTRUMENTS | 5 |
| IV. TRANSMITTER TESTS | 6 |
| 4.1 OUTPUT POWER TEST | 7 |
| 4.2 TRANSMIT FREQUENCY RANGE TEST | 9 |
| 4.3 CW TRANSMIT-FREQUENCY ACCURACY TEST | 11 |
| 4.4 SPECTRAL PURITY TEST | 13 |
| 4.5 TWO-TONE TRANSMIT IMD TEST | 16 |
| 4.6 SSB CARRIER AND UNWANTED SIDEBAND SUPPRESSION AT 14.250 MHz | 18 |
| 4.7 CW KEYING WAVEFORM TEST | 21 |
| 4.8 PTT TO SSB/FM RF OUTPUT TEST | 23 |
| 4.9 TRANSMIT/RECEIVE TURN-AROUND TEST | 27 |
| 4.10 KEYER SPEED AND SIDETONE FREQUENCY TEST | 30 |
| 4.11 COMPOSITE NOISE TEST | 32 |
| V. RECEIVER TESTS | 35 |
| 5.1 NOISE FLOOR TEST | 36 |
| 5.2 AM RECEIVE SENSITIVITY TEST | 39 |
| 5.3 FM SINAD AND QUIETING TEST | 41 |
| 5.4 RECEIVE FREQUENCY RANGE TEST | 44 |
| 5.5 FIRST IF AND IMAGE REJECTION TEST | 46 |
| 5.6 ANTENNA PORT ISOLATION TEST | 49 |
| 5.7 BLOCKING GAIN COMPRESSION | 51 |
| 5.7A RECIPROCAL MIXING | 54 |
| 5.8 TWO-TONE 2ND AND 3RD ORDER DYNAMIC RANGE TEST | 57 |
| 5.9 FM ADJACENT-CHANNEL SELECTIVITY TEST | 61 |
| 5.10 FM TWO-TONE 3RD ORDER DYNAMIC RANGE TEST | 64 |
| 5.11 AUDIO POWER OUTPUT TEST | 68 |
| 5.12 AUDIO AND IF FREQUENCY RESPONSE TEST | 70 |
| 5.13 SQUELCH SENSITIVITY TEST | 73 |
| 5.14 S METER TEST | 75 |
| 5.15 IN BAND IMD TEST | 77 |
| 5.16 NOTCH FILTER TEST | 80 |
| 5.17 DSP NOISE REDUCTION TEST | 86 |
| 5.18 EQUIVALENT RECTANGULAR BANDWIDTH TEST | 89 |
| 5.19 NOISE FIGURE CALCULATION | 89 |
| VI. NON-STANDARD AND SPECIAL PURPOSE TESTS | 90 |
| 6.1 LOW VOLTAGE AND TEMPERATURE CHAMBER TEST | 90 |
| 6.3 RECEIVER BIT-ERROR RATE TEST | 93 |
| 6.4 TRANSMITTER BIT-ERROR RATE TEST | 95 |
| 6.6 POWER METER TEST | 98 |
| DATA SHEETS | 103 |
| 6.7 HF LINEAR AMPLIFIER TEST | 129 |
| 6.7B LINEAR AMPLIFIER IMD TEST | 131 |
| LINEAR AMPLIFIER DATA SHEET | 135 |
| ABOUT THE LINEAR AMPLIFIER IMD TEST FIXTURE | 136 |
| VII. APPENDIX A: ARRL CUSTOM TEST CIRCUITS | 137 |
| VIII. APPENDIX B: FORMULAS FOR QST PRODUCT REVIEW CALCULATIONS | 163 |

# PURPOSE AND SCOPE

## 1.1 PURPOSE

The purpose of this manual is to ensure that all ARRL testing of HF transceivers, transmitters, and receivers will be conducted in a consistent and well-defined manner. The procedures used for this testing have been broken down into clear, step-by-step instructions. All testing should be "done by the book" in order to guarantee that all equipment is evaluated to the same standards.

This manual contains three major sections: Transmitter tests (Chapter 4), Receiver tests (Chapter 5) and Data Sheets (Chapter 6). The data sheets are arranged to allow test results to be recorded in the same order that measurements are taken.

## 1.2 SCOPE

This manual is designed to cover a wide range of amateur HF equipment. It is not intended to replace common sense or the expertise of an experienced test engineer. It is important, therefore, for the test engineer to be familiar with the Lab test equipment and the Device Under Test (DUT). The manufacturer's manual for the DUT should be completely read and understood before any testing is performed. At no time should any equipment be operated in a manner that is inconsistent with the manufacturer's recommended procedures or published limits. Failure to understand the unit under test could result in test error or, even worse, damage to the equipment, or worst of all, damage to the test engineer.

## II. LIST OF FIGURES


| Figure | Description |
|---|---|
| 4-1 | Power Output Test Hook-Up |
| 4-2 | Transmit Frequency Range Test Hook-Up (no changes from 4-1) |
| 4-3 | CW Transmit Accuracy Test Hook-Up |
| 4-4 | Spectral Purity Test Hook-Up |
| 4-4A | Example Spectral Purity Plot |
| 4-5 | Two-Tone Transmit IMD Test Hook-Up |
| 4-6 | SSB Carrier and Unwanted Sideband Suppression Test Hook-Up (no changes from 4-5) |
| 4-7 | CW Keying Waveform Test Hook-Up |
| 4-7A | Example Scope Trace of RF Output CW Keying Waveform |
| 4-8 | PTT SSB/FM RF Output Test Hook-Up |
| 4-9 | Transmit/Receive Turn-Around Time Test Hook-Up |
| 4-10 | Keyer Speed and Side-tone Frequency Test Hook-Up |
| 4-11 | Composite Noise Test Hook-Up |
| 5-1 | CW Minimum Discernible Signal (MDS) Test Hook-Up |
| 5-2 | AM Receive Sensitivity Test Hook-Up |
| 5-3 | FM SINAD and Quieting Test Hook-Up |
| 5-4 | Receive Frequency Range Test Hook-Up |
| 5-5 | First IF and Image Rejection Test Hook-Up |
| 5-6 | Antenna Port Isolation Test Hook-Up |
| 5-7 | Blocking Gain Compression Test Hook-Up |
| 5-7A | Reciprocal Mixing Test Hook-Up |
| 5-8 | Two-Tone, 3rd Order Dynamic Range Test Hook-Up |
| 5-9 | FM Adjacent Channel Selectivity Test Hook-Up |
| 5-10 | FM Two-Tone, 3rd Order Dynamic Range Test Hook-Up |
| 5-11 | Audio Power Output Test Hook-Up |
| 5-12 | Audio and IF Frequency Response Test Hook-Up |
| 5-13 | Squelch Sensitivity Test Hook-Up |
| 5-14 | S-Meter Test Hook-Up |
| 5-15 | In-Band IMD Test Hook-Up |
| 5-16 | Notch Filter Test Hook-Up |
| 5-17 | Equivalent Rectangular Bandwidth Test Hook-Up |
| 6-1 | Low Voltage and Temperature Chamber Test Hook-Up |
| 6-2 | DSP Noise Reduction Test Hook-Up |
| 6-3 | Receiver Bit-Error-Rate (BER) Test Hook-Up |
| 6-4 | Transmitter Bit-Error-Rate (BER) Test Hook-Up |
| 6-5 | Receiver Phase-Noise Test Hook-Up |

## II. LIST OF REQUIRED TEST INSTRUMENTS

| Instrument (or equivalent) | Manufacturer | Model | Qty. |
|---|---|---|---|
| Spectrum Analyzer | HP | HP8563E | 1 |
| Distortion/Audio Meter | HP | 339A | 1 |
| RF Signal Generator | IFR | 2041 | 1 |
| RF Signal Generator | Marconi | 2041 | 1 |
| In-line RF Wattmeter | BIRD | 4381 | 1 |
| RF Signal Generator | Rhode&Schwarz | CMT 54 | 1 |
| Digital Phosphor Oscilloscope | Tektronix | 3052B | 1 |
| Two-Tone Audio Generator | ARRL | N/A | 1 |
| Power Attenuator | BIRD | 8329 | 1 |
| Keying-Test Generator | ARRL | N/A | 1 |
| Adjustable RF Attenuators | HP | 355C/D | 1 |
| Hybrid Combiner | Ten-Tec | 651 (2-60 MHz) | 1 |
| Voltmeter | Bench | N/A | 1 |
| Ammeter | Bench | N/A | 1 |
| Phase-Noise Mixer | ARRL | N/A | 1 |
| 14.025 MHz Low-Noise Osc. | ARRL | N/A | 1 |
| Power Supply | HP | 6268B | 1 |
| Telegraph Key | Any | N/A | 1 |
| Attenuators | BIRD | 8340-100 or similar | As R'qd |
| HI-Z Audio Monitor Amp | ARRL | N/A | 1 |
| Wideband Noise Generator | Elecraft | N-gen | 1 |

## IV. TRANSMITTER TESTS

4.0.1 As shown in the Table of Contents, there are 11 transmitter tests outlined in this chapter. They have been arranged to minimize the required number of hook-up changes and modifications. However, each hook-up is shown complete, with all changes from the previous test clearly indicated. A block diagram accompanies each hook-up and any changes from the previous test are shown within a dotted rectangle. This affords the flexibility to easily start anywhere within the test plan and to perform these tests in any desired order.

4.0.2 Before any transmitter testing is performed, it is essential for the test engineer to be completely familiar with the Device Under Test (DUT) and the test equipment that will be used. The transmitter output power must not exceed the limitations of any test equipment used or the manufacturer's specifications for the DUT – *including duty cycle or time*. The transmitter must be operated in a manner exactly as specified and be properly tuned. Any test in this manual that would cause a piece of equipment to be operated in a manner inconsistent with its manual must be accordingly modified as required by the test engineer.

4.0.2.1 Other considerations are as follows:

*Spectrum Analyzer/Frequency Counter*

The input to the spectrum analyzer or frequency counter should not exceed the range of –5 dBm to 0 dBm. Therefore, it is necessary to attenuate the output of transmitters or amplifiers down to this level. For example, if a particular transmitter is rated at 100 watts output, it will be necessary to use 50 dB of attenuation to obtain a level of 0 dBm.

*RF Power Meter*

The necessary element for the inline RF-power meter must be selected for the expected power and frequency. There are also multiplier switches on the wattmeter. Select the combination of switches that equals the power rating of the element.

*RF Power and Step Attenuators*

RF power attenuators have an INPUT and an OUTPUT. Make sure that the RF source is connected to the INPUT of the power attenuator. Connecting RF power to the OUTPUT of the attenuator will result in damage to the attenuator or to the transceiver. The attenuators have a power rating, which must be observed. It is always best to be conservative.

**4.1 OUTPUT POWER TEST**

4.1.1 The purpose of the Transmitter Output Power Test is to measure the RF output power of the DUT across each band in each of its available modes. A two-tone audio input, at a level within the manufacturer's microphone-input specifications, will be used for the SSB mode. No modulation will be used in the AM and FM modes. DC current consumption at the manufacturer's specified supply voltage is also measured, if applicable.

4.1.2 Test hook-up (See Fig. 4-1)

4.1.2.1 With all test equipment and DUT power switches in the OFF position and the transceiver in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUTPUT to Wattmeter INPUT | As Required to Type N | 50-Ohm Coax |
| Wattmeter OUTPUT to RF Power Attenuator | Type N to Type N | 50-Ohm Coax |
| Two-Tone Audio Generator to DUT Microphone INPUT | Banana or BNC to As Required | Coax |
| Telegraph Key to DUT KEY INPUT | As Required | As Required |
| AC power Only/AC Source to DUT Power Input | As Supplied with DUT | As Supplied with DUT |
| DC Power Only | As Required | As Required |

1) Power supply to AC source.
2) Power supply OUT to voltmeter and ammeter.
3) Set power supply to specified voltage and connect to DUT.

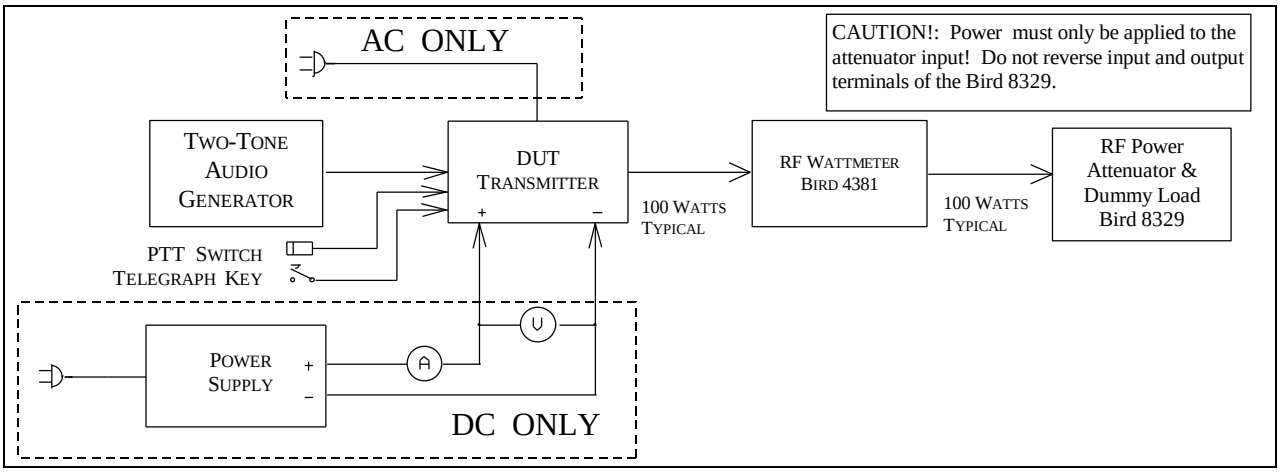


Fig. 4-1 Power Output Test Hook-Up

4.1.3 Test Procedure

4.1.3.1 Turn the DUT and RF wattmeter power switches to ON and set the following controls:

| Instrument | Control | Position |
|---|---|---|
| Two-Tone Audio Generator | TONE A | OFF |
| | TONE B | OFF |
| | HI-Z/LO-Z | As Required |
| | BALANCE | Center |
| | LEVEL | Full CCW |
| | Attenuator | –30 dB |
| RF Wattmeter | Mode Select Button | FWD PEP |
| | Element | As Required |
| | Forward Element Range | As Required |
| DUT | Mode | CW |
| | BAND Selector | Lowest Available |
| | XMIT/RCV | Receive |
| | DRIVE or RF LEVEL | Minimum |

4.1.3.2 Allow all equipment at least 10 minutes warm-up time before proceeding to step 4.1.3.3

4.1.3.3 Tune the DUT per the operator's manual. Put the DUT in the CW transmit mode at minimum DRIVE/RF LEVEL.

> **NOTE**: The following three paragraphs, 4.1.3.4 , 4.1.3.5 and 4.1.3.6 apply only to transceivers with a DC power supply. Skip these paragraphs if testing an ac-powered DUT and make a note of the power type on Data Sheet.

4.1.3.4 Depress telegraph key. Observe that some minimum level of RF power is shown by the DUT Power Output Meter and Bird Wattmeter. Increase the RF DRIVE/LEVEL control if necessary. Record the power-supply ammeter and voltmeter values in 4.1.3.4 of the data sheet.

4.1.3.5 Increase RF DRIVE/LEVEL control to the maximum allowed by the manufacturer. Again, depress the telegraph key and record the power-supply meter readings in 4.1.3.5 of the data sheet. Decrease the RF DRIVE control to minimum. Release the telegraph key.

4.1.3.6 Increase the DUT AF gain control to maximum. (There should be no input signal.) Any light or display illumination should be in the default position. Measure and record the power supply readings as in the previous step.

4.1.3.7 Depress the key and observe the DUT power output meter and wattmeter indications. Slowly tune the DUT to the upper band edge while observing these meters. Release the key and record the maximum value observed on the wattmeter in 4.1.3.7 of the data sheet.

4.1.3.8 Increase the DRIVE/RF LEVEL control to the maximum allowed. Depress the telegraph key. Slowly tune the DUT down to the lower band edge while observing both the DUT power output meter and wattmeter as in the previous step. Release the telegraph key and record the minimum wattmeter values observed in 4.1.3.8 of the data sheet. Return the DRIVE/RF LEVEL control to minimum.

4.1.3.9 Set the DUT for the lower edge of the 80-meter band and retune as necessary. Repeat steps 4.1.3.7 and 4.1.3.8. Record all values in 4.1.3.9 of the data sheet. Repeat this procedure for all the remaining bands. Record both the DUT power output meter and wattmeter indications for the 20-meter band. Note any significant deviations in the DUT meter indication observed between bands.

4.1.3.10 Return the transmitter to the lower edge of the 20-meter band and retune as necessary. Set the two-tone audio generator power switch to ON. Adjust for proper balance between both tones by setting the BALANCE control to the position so indicated. Adjust the generator LEVEL control for maximum specified ALC. Set the RF wattmeter to FWD PEP. Place the DUT in the USB mode and tune as required with the DUT mic gain control set about half-way. Adjust the audio generator for maximum audio as specified by the manufacturer's manual with the DUT mic gain control set about half-way.

4.1.3.11 Repeat steps 4.1.3.7 and 4.1.3.8 and using the PTT switch to key the transmitter. Verify similar power output performance with the DUT in the LSB mode. Note any significant deviation from the USB mode. Record DUT power output meter and wattmeter indications in 4.1.3.11 of the Data Sheets.

4.1.3.12 Set the RF Wattmeter to FWD CW and the two-tone generator power switch to OFF. Return the transmitter to the lower edge of the 80-meter band. Place the DUT in the AM mode. Repeat steps 4.1.3.7 and 4.1.3.8 using the PTT switch to key the transmitter. *Be sure not to exceed the maximum power output specified by the manufacturer for this mode*. Record DUT power output meter and wattmeter indications in 4.1.3.12 of the Data Sheets.

4.1.3.13 Return the transmitter to the lower edge of the 10-meter band. Place the DUT in the FM mode. Repeat steps 4.1.3.7 and 4.1.3.8 for this mode. *Be sure not to exceed the maximum power output specified by the manufacturer for this mode.*

**NOTE:** Proceed to the following step *only* if the DUT has a transverter output.

4.1.3.14 Put the DUT into the CW mode and activate the transverter function.. Set to the 15 meter band. Connect the HP-437 microwatt meter and any needed attenuation to the transverter output. (Input to the microwatt meter must not exceed +20 dBm.) Repeat steps 4.1.3.7 and 4.1.3.8. Record on Data Sheet. Also observe and record any RF output that may appear at the normal antenna output.

## 4.2 TRANSMIT FREQUENCY RANGE TEST

4.2.1 The purpose of the Transmit Frequency Range Test is to determine the range of frequencies, including those outside amateur bands, for which the transmitter may be used.

4.2.2 Test Hook-up (See Fig. 4-2)

> **NOTE**: No further hook-up changes are required if proceeding directly from the previous Power Output Test. This hook-up procedure, therefore, does not apply in this case and you may now proceed to step 4.2.3.1.

4.2.2.1 Connect the following with all power switches in the OFF position and the transceiver in the receive mode:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUTPUT to Wattmeter INPUT | As Required to Type N | 50-Ohm Coax |
| Wattmeter OUTPUT to RF Power Attenuator | Type N to Type N | 50-Ohm Coax |
| Telegraph Key to DUT KEY INPUT | As Required | As Required |
| AC power Only/AC Source to DUT Power Input | As Supplied with DUT | As Supplied with DUT |
| DC Power Only<br>1) Power supply to AC source.<br>2) Power supply OUT to voltmeter and ammeter. | As Required | As Required |

3) Set power supply to specified voltage and connect to DUT.

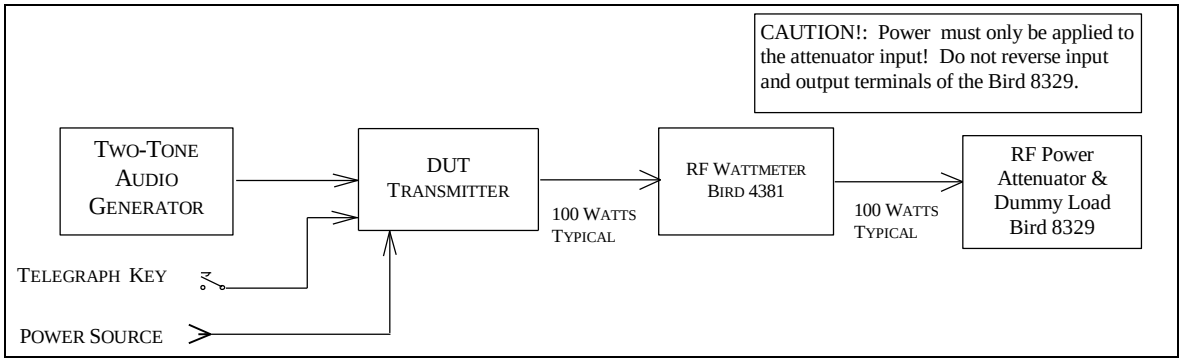



Fig. 4-2 — Transmit Frequency Range Test Hook-up

4.2.3 Test Procedure

4.2.3.1 Turn the DUT and RF wattmeter power switches to ON and set the following controls:

| Instrument | Control | Position |
|---|---|---|
| RF Wattmeter | Mode select button | FWD CW |
| | Element | As Required |
| | Power Range | As Required |
| DUT | Mode | CW |
| | Band Selector | Lowest available band |
| | XMIT/RCV | RECEIVE |
| | DRIVE/RF OUT | Minimum |

4.2.3.2 Allow all equipment at least 10 minutes warm up time.

4.2.3.3 Tune the DUT per the operator's manual for the lowest available band. Return the DRIVE/RF OUTPUT control to minimum. (The DUT should still be in the CW mode.)

4.2.3.4 Tune down near the low end of the band. Depress the telegraph key. Observe some minimum RF indication on the wattmeter. Slowly tune down until the RF drops out or the manufacturer's specified limit is achieved. Release the key and record this frequency, as indicated by the DUT display in 4.1.3.4 of the data sheet.

4.2.3.5 Tune to the upper limit of the band. Again, depress the telegraph key and slowly tune up until the RF drops out. Release the key and record this frequency in 4.1.3.5 of the data sheet.

4.2.3.6 Repeat steps 4.1.3.3 to 4.1.3.5 for all remaining available bands on the DUT.

## 4.3 CW TRANSMIT-FREQUENCY ACCURACY TEST

4.3.1 The purpose of the CW Transmit-Frequency Accuracy Test is to measure and compare the actual output frequency of the DUT with its display adjusted for 14.000 MHz down to the least significant available digit.

4.3.2 TEST HOOK-UP (See Fig. 4-3)

**NOTE**: If proceeding directly from the previous Transmit Frequency Range Test, you need only modify the existing hook-up. Proceed to the steps indicated with a dotted line.

4.3.2.1 With all power switches in the OFF position and the transceiver in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUTPUT to wattmeter INPUT | As Required to type N | 50-Ohm coax |
| Wattmeter OUTPUT to Power Attn INPUT | Type N to type N | 50-Ohm coax |
| Pwr Attn OUT to 10-dB step INPUT | Type N to BNC | 50-Ohm coax |
| 10-dB step Attn OUT to 1-dB step IN | BNC to BNC | 50-Ohm coax |
| 1-dB step ATTN OUT to Freq Counter IN | BNC to BNC | 50-Ohm coax |
| Telegraph key to DUT KEY INPUT | As Required | As Required |
| AC or DC power to DUT | As Required | As Required |

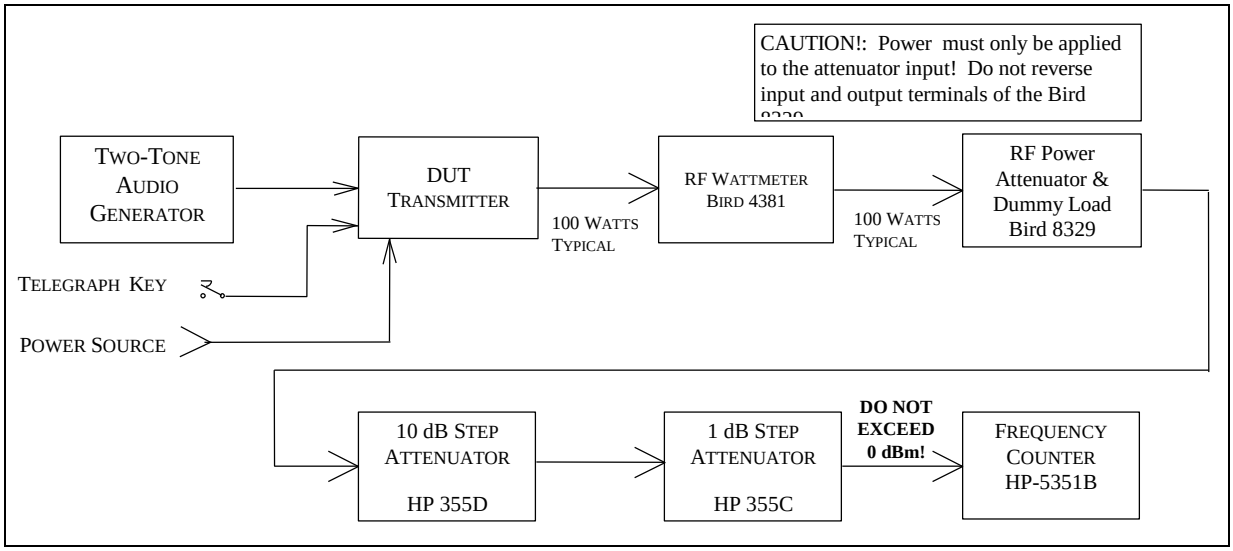



Fig. 4-3 — CW Transmit Accuracy Test Hook-up

4.3.3 Test Procedure

4.3.3.1 Turn the DUT, RF wattmeter and frequency counter power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | 20 Meters |
| | Frequency | 14.000 00 MHz |
| | XMIT/RCV | RECEIVE |
| | DRIVE/RF OUT | Minimum |
| RF Wattmeter | Mode Select Button | FWD CW |
| | Element | As Required |
| | Power Range | As Required |
| 10-dB Step Attn | Attenuator | 40 dB |
| 1-dB Step Attn | Attenuator | Any |
| Frequency Counter | 50 Ohm | 50 Ohm (Depress) |

4.3.3.2 Key the transmitter. Observe the minimum output power on the RF wattmeter and unkey the transmitter. Set the step attenuators to provide –5 to 0 dBm input to the frequency counter.
**Do not exceed the maximum input limitation to the Frequency Counter.**

4.3.3.3 Check and readjust, if necessary, the DUT for 14.000 00 MHz as indicated by its display. Record this value if the DUT has a digital type display, or, as best as possible with an analog display.

4.3.3.4 Key the transmitter and note the frequency as indicated by both the DUT and the frequency counter. Place a check mark in 4.3.3.4 of the data sheet if there is any significant deviation from 14.000 00 MHz

4.3.3.5 Increase the step attenuators to 40 dB. Set the transmitter power to the maximum allowed. Key the transmitter and observe the RF output power as shown by the wattmeter. Set the attenuators to provide –5 to 0 dBm input to the counter.

4.3.3.6 Key the transmitter. Record the DUT and Frequency Counter indications in 4.3.3.6 of the Data Sheet. Unkey the transmitter.

4.3.3.7 Reduce the voltage to 12 V dc (if using a DC supply). Again key the transmitter and note the frequency displayed on the DUT and frequency counter. Place a check mark in 4.3.3.7 of the data sheet if there is any significant deviation from initial data.

## 4.4 SPECTRAL PURITY TEST

4.4.1 The purpose of the Spectral Purity Test is to determine and measure the content of any spurious emissions in the output of the transmitter. Full-power carriers will be examined and minimum power checked on all available bands.

4.4.2 Test hook-up (See Fig. 4-4)

**NOTE**: If proceeding from the previous Transmit Frequency Accuracy Test, skip to the steps indicated with a dotted line.

4.4.2.1 With all power switches in the OFF position and the transceiver in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUTPUT to wattmeter INPUT | As Required to type N | 50-Ohm coax |
| Wattmeter OUTPUT to Power Attn INPUT | Type N to type N | 50-Ohm coax |
| Power Attn OUT to 10-dB step INPUT | Type N to BNC | 50-Ohm coax |
| 10-dB step Attn OUT to 1-dB step IN | BNC to BNC | 50-Ohm coax |
| 1-dB step Attn OUT to Spec Analyzer IN | BNC to BNC | 50-Ohm coax |
| Telegraph key to DUT KEY INPUT | As Required | As Required |
| AC or DC power to DUT | As Required | As Required |

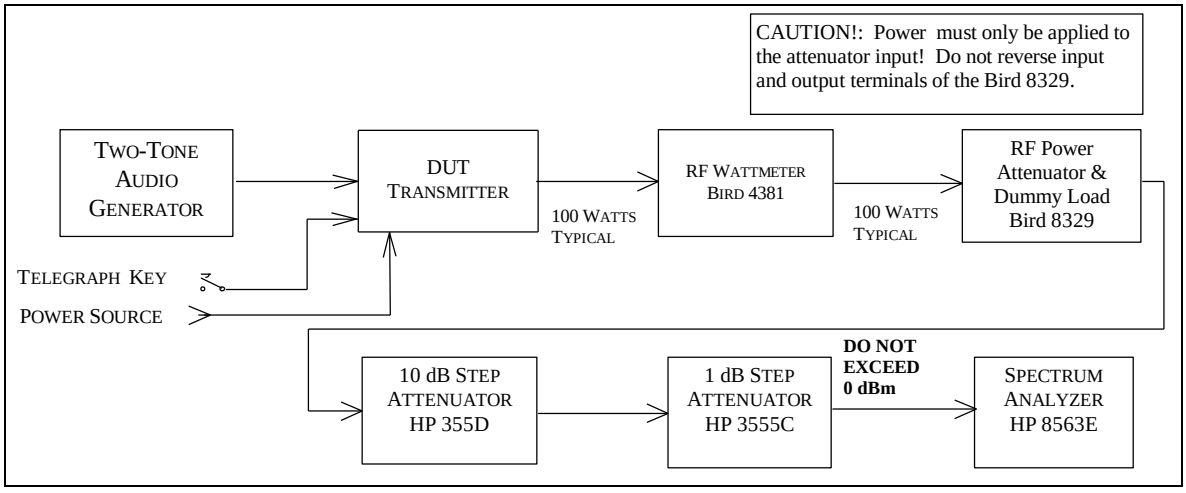



Fig. 4-4 — Spectral Purity Test Hook-up

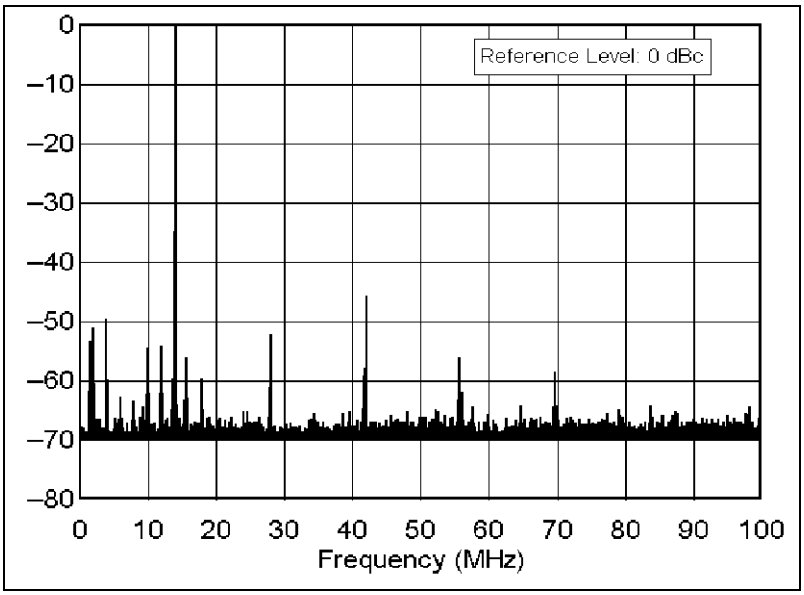


Fig. 4-4A — Example Spectral Purity Plot

4.4.3 Test Procedure

4.4.3.1 Turn the DUT, RF wattmeter and spectrum analyzer power switches to ON and set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | Lowest available |
| | XMIT/RCV | Receive |
| | DRIVE/RF LEVEL | Minimum |
| RF Wattmeter | Push Button Mode Select | FWD CW |
| | Element | As Required |
| | Forward Element Range | As Required |
| 10-dB Step Attn | Attenuator | 40 dB |
| 1-dB Step Attn | Attenuator | Any |
| Spectrum Analyzer | START FREQ - STOP FREQ | 0 – 50 MHz |
| | REF LEV (AMPLITUDE) | –10 dBm |
| | ATTEN (AMPLITUDE) | 20 dB |
| | RES BW (BW) | 10 kHz |
| | VIDEO BW (BW) | 30 kHz |
| | THRESHOLD (DISPLAY) | –80 dBm |
| | SWP TIME (SWEEP) | AUTO |

4.4.3.2 **NOTE**: If proceeding directly from the previous CW Transmit Frequency Accuracy Test, skip this paragraph.

Receiver hiss should be heard; adjust volume to desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 4.4.3.3.

4.4.3.3 Tune the DUT per the operator's manual at the low end of the band. Put the DUT in the CW transmit mode at maximum DRIVE/RF LEVEL.

4.4.3.4 Depress the telegraph key and observe the power output shown by the Bird wattmeter. Release the key and set the step attenuators for –10 dBm input to the spectrum analyzer. *CAUTION: The input to the spectrum analyzer at no time should exceed +10 dBm. Damage to this instrument will occur at an input level of +30 dBm or greater.*[1]

4.4.3.5 Key the transmitter. The largest pip to the right of 0 MHz should be the fundamental. (This may be verified by use of the PEAK SEARCH button. With the marker to be at the top of the pip, the indicated marker frequency should be very close to the transmitted frequency.)

4.4.3.6 Adjust the REF LEV control on the spectrum analyzer so that the peak of the fundamental pip is at the Log Ref (0 dB) line on top of the display graticule. (To optimize the display, the FREQ START-FREQ STOP may be set for 0-10 MHz and the RES BW may be set for 30 kHz on the 160-meter band.)

4.4.3.7 The spectrum analyzer is now calibrated. The smaller pips are harmonics or spurs. The level of each spur, in dB below the fundamental, can be read directly from the display graticule. Each horizontal division represents 5 MHz each in the case of a 0-50 MHz frequency range. The frequency range may be adjusted for the specific display being considered, or as deemed appropriate by the test engineer. Each vertical division represents 10 dB.

4.4.3.8 Turn on the crystal calibrator, if the DUT has one, and observe any spurious emissions created by the calibrator. Record in notes section of data sheet and turn off the calibrator.

4.4.3.9 Slowly tune up the band while observing the analyzer display. Retune as necessary if the transmitter is not a broadband type. Note the worst case observed. Record on data sheet.

4.4.3.10 Reduce the transmitter power output to minimum. Again, observe the analyzer display while tuning down to the lower band edge. Note the worst case observed and record on data sheet.

4.4.3.11 Repeat steps 4.4.3.9 and 4.4.3.11 for all remaining available bands on the DUT. The Frequency Range settings for each band are as follows:

| | |
|---|---|
| 1.8 MHz | 0-50 MHz |
| 14 MHz | 0-100 MHz |
| 28 MHz | 0-200 MHz |
| 50 MHz | 0-500 MHz |
| 144 MHz | 0-1000 MHz |
| 222 MHz | 0-1000 MHz |
| 432 MHz: | 0-2000 MHz |

4.4.3.12 Return the transmitter to the worst case on the worst band. With the all significant spurs visible on the screen, take a single sweep by depressing the SGL SWP button. (Typically, this will be the second or third harmonic, although it is possible for VHF parasitic to be present.) The START FREQ, STOP FREQ and RES BW may be varied as required. Record all information on data sheet and save the plot to an appropriately named file.

---

[1] "Woe be unto he who breaks this thing." — Edward F. Hare, W1RFI

## 4.5 TWO-TONE TRANSMIT IMD TEST

4.5.1 The purpose of the Two-Tone Transmit Test is to measure the intermodulation-distortion (IMD) products present in the RF output of the DUT transmitter. The transmitter will be operated in the SSB mode at 3.900 MHz and 14.250 MHz initially and then on all other available bands subsequently. A two-tone audio input at frequencies of 700 and 1900 Hz, within the manufacturer's amplitude specifications, will be used.

4.5.2 Test hook-up (See Fig. 4-5)

**NOTE**: If proceeding in the test series, only the two-tone generator, shown within the dotted line, must be added to the previous Spectral Purity Test hook-up for the IMD Test.

4.5.2.1 With all power switches in the OFF position and the transceiver in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUT To Wattmeter IN | As Required To Type N | 50-Ohm Coax |
| Wattmeter OUT To PWR Attn IN | Type N To Type N | 50-Ohm Coax |
| PWR Attn OUT To 10-dB Step IN | Type N To BNC | 50-Ohm Coax |
| 10-dB Step OUT To 1-dB Step IN | BNC To BNC | 50-Ohm Coax |

| Connection | Connectors | Cable Type |
|---|---|---|
| 1-dB Step OUT To Spectrum Analyzer IN | BNC To BNC | 50 Ohm Coax |
| Two-Tone Generator OUT To DUT MIC IN | BNC To As Required | Coax |
| Power to DUT | As Required | As Required |

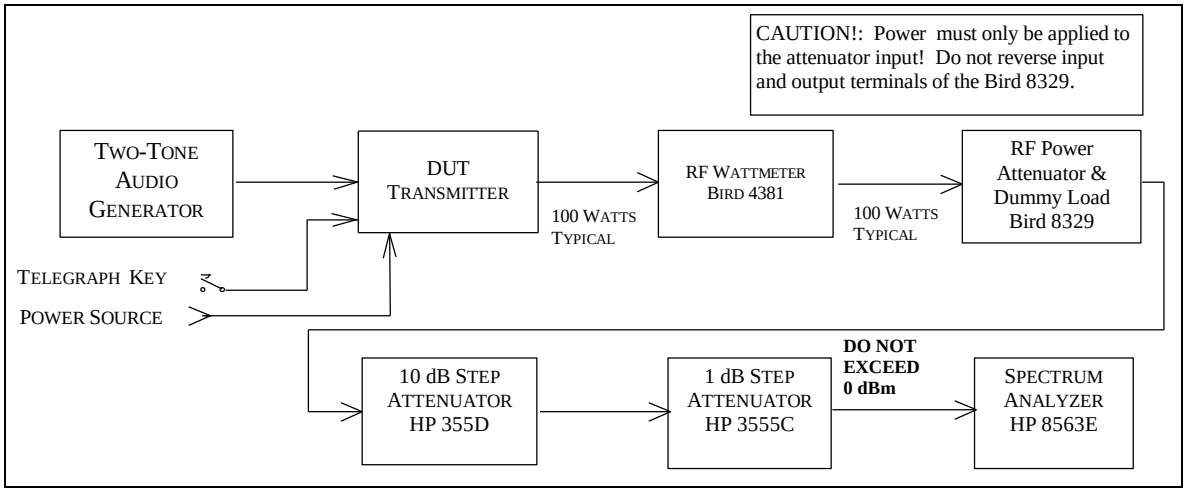



Fig. 4-5 — Two-Tone Transmit IMD Test Hook-up

4.5.3 Test Procedure

4.5.3.1 Turn on the DUT, RF wattmeter and spectrum analyzer and set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | LSB |
| | Band Selector | 80 Meters |

| | | |
|---|---|---|
| | Frequency | 3.900 00 MHz |
| | XMIT/RCV | RECEIVE |
| | DRIVE or RF LEVEL | Minimum |
| | Microphone | Minimum |
| Two-Tone Generator | TONE A (700 Hz) | OFF |
| | TONE B (1900 Hz) | OFF |
| | HI-Z/LO-Z | As Required |
| | BALANCE | Center |
| | LEVEL | Full CCW |
| | Attenuator | –30 dB |
| RF Wattmeter | Push Button Mode Select | FWD PEP |
| | Element | As Required |
| | Forward Element Range | As Required |
| 10-dB Step Attn | Attenuator | –40 dB |
| 1-dB Step Attn | Attenuator | Any |
| Spectrum Analyzer | CENTER FREQ (FREQUENCY) | 3.89870 MHz |
| (Menu in Parentheses) | SPAN (SPAN) | 20 kHz |
| | REF LEV (AMPLITUDE) | –40 dBm |
| | ATTEN (AMPLITUDE) | 20 dB |
| | RES BW (BW) | 100 Hz |
| | VIDEO BW (BW) | 10 kHz |
| | THRESHOLD (DISPLAY) | –110 dBm |
| | SWP TIME (SWEEP) | AUTO |

4.5.3.2 **NOTE**: If proceeding from the previous tests, this paragraph may be skipped.

Receiver hiss should be heard; adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 4.5.3.3.

4.5.3.3 Tune the DUT per the operator's manual for the test frequency of 3.900 MHz. Turn on the two-tone generator and set both tone switches to ON. With the DUT in the LSB mode, set the generator LEVEL and ATTENUATOR controls for the maximum audio input as specified by the manufacturer. If the manufacturer does not list a specification for this figure, adjust the 2-tone generator's amplitude for maximum rated RF output of the transmitter with the transmitter's microphone gain near maximum. Observe the transmitter power as shown by the wattmeter. Ensure that the output power of the DUT is not greater than the manufacture's maximum power output rating. Unkey the transmitter and set the step attenuators for approximately -46 dBm input to the spectrum analyzer.

**CAUTION**: The input to the spectrum analyzer at no time should exceed +10 dBm.

4.5.3.4 Place the DUT in the VOX mode and verify operation with the signal generator. Note on data sheet if VOX does not function correctly. Return the DUT to the PTT mode and key the transmitter. Set the BALANCE control on the generator for equal tone amplitude as shown on the display. Adjust the CENTER FREQ, if necessary, so that the display center is half-way between the two pips. The IMD-distortion products should now be visible.

4.5.3.5 Adjust the REF LEVEL control (and step attenuators, if necessary) for the peak of the two pips to be at –6 dB. The spectrum analyzer is now calibrated. The amplitude of each IMD distortion product may now be read in dB PEP (dB below the peak envelope power) directly from the display.

4.5.3.6 Manipulate, if necessary, the two-tone generator audio LEVEL and the transmitter audio gain and drive control to obtain the lowest possible IMD products. If this is done, the spectrum analyzer REF LEVEL control (and possibly the step attenuators) may need to be reset for tone pips of –6 dB.

4.5.3.7 Set the SWP TIME (in the SWEEP menu) for 6 seconds. Take a single sweep by depressing the SGL SWEEP button. Record all info on data sheet. Print and save to an appropriate file name.

4.5.3.8 Set and tune the transmitter for USB at a frequency of 14.250 MHz. Set the CENTER FREQ for 14.25130 MHz. and return the SWP TIME back to AUTO. Repeat paragraphs 4.5.3.3 to 4.5.3.7 for this frequency.

4.5.3.9 Repeat step 4.5.3.8 for the following frequencies (if applicable to the DUT): 1.850 MHz, 7.250 MHz, 10.120 MHz, 18.120 MHz, 21.250 MHz, 24.950 MHz, 28.350 MHz, 50.200 MHz, 144.200 MHz and 432.200 MHz.

## 4.6 SSB CARRIER AND UNWANTED SIDEBAND SUPPRESSION AT 14.250 MHz

4.6.1 The purpose of the SSB Carrier and Unwanted-Sideband Suppression Test is to determine the level of suppression of the unwanted sideband and carrier relative to Peak Envelope Power (PEP). The transmitter output is observed on the spectrum analyzer and the unwanted components are compared to the desired sideband.

4.6.2 Test hook-up (See Fig. 4-6)

**NOTE**: No changes in hook-up are required if proceeding directly from the previous Two-Tone Transmit IMD Test, proceed now, in this case, to paragraph 4.6.3.

4.6.2.1 With all power switches in the OFF position and the transceiver in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUT To Wattmeter IN | As Required To Type N | 50-Ohm Coax |
| Wattmeter OUT To PWR Attn IN | Type N To Type N | 50-Ohm Coax |
| PWR Attn OUT To 10-dB Step IN | Type N To BNC | 50-Ohm Coax |
| 10-dB Step OUT To 1-dB Step IN | BNC to BNC | 50-Ohm Coax |
| 1-dB Step OUT To Spec Analyzer IN | BNC To BNC | 50-Ohm Coax |
| Two-Tone Generator OUT to DUT MIC IN | BNC to As Required | Coax |

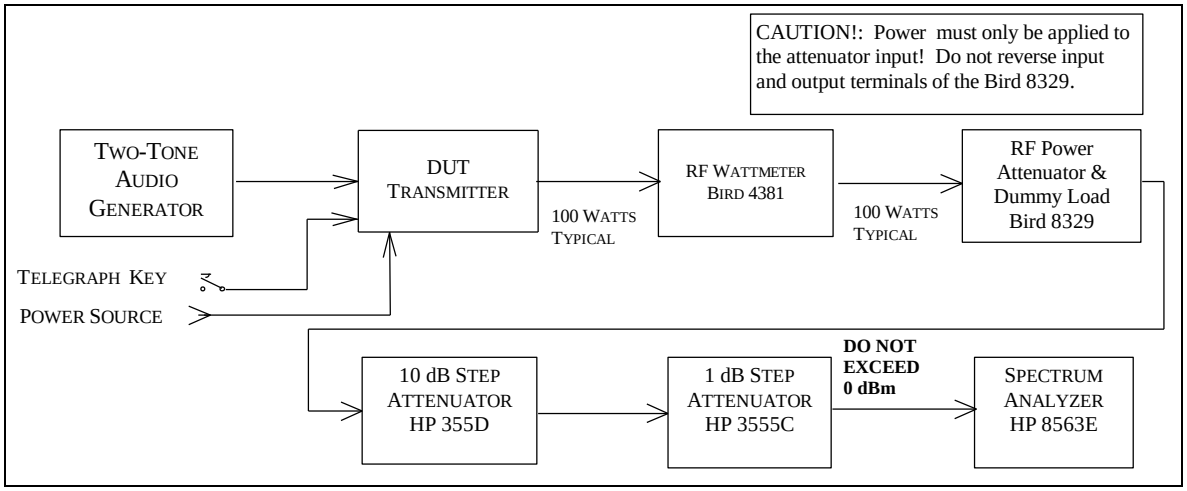



Fig. 4-6 — SSB Carrier and Unwanted Sideband Suppression Test Hook-up

4.6.3 Test Procedure

4.6.3.1 Turn the DUT, RF wattmeter and spectrum analyzer power switches to ON and set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | USB |
| | Band Selector | 20 Meters |
| | Frequency | 14.250 MHz |
| | XMIT/RCV | Receive |
| | DRIVE or RF LEVEL | Minimum |
| | Microphone | Minimum |

| Instrument | Control | Position |
|---|---|---|
| Two-Tone Generator | TONE A (700 Hz) | OFF |
| | TONE B (1900 Hz) | OFF |
| | HI-Z/LO-Z | As Required |
| | BALANCE | Center |
| | LEVEL | Full CCW |
| | Attenuator | –30 dB |
| RF Wattmeter | Push Button Mode Select | FWD PEP |
| | Element | As Required |
| | Forward Element Range | As Required |
| 10-dB Step Attn | Attenuator | –40 dB |
| 1-dB Step Attn | Attenuator | Any |
| Spectrum Analyzer | CENTER FREQ (FREQUENCY) | 14.25130 MHz |
| | SPAN (SPAN) | 20 kHz |
| | REF LEV (AMPLITUDE) | –40 dBm |
| | ATTEN (AMPLITUDE) | 20 dB |
| | RES BW (BW) | 100 Hz |
| | VIDEO BW (BW) | 10 kHz |
| | THRESHOLD (DISPLAY) | OFF |
| | SWP TIME (SWEEP) | AUTO |

4.6.3.2 **NOTE**: If proceeding from the previous tests, this paragraph may be skipped.

Receiver hiss should be heard; adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 4.6.3.3.

4.6.3.3 Tune the DUT per the operator's manual for the test frequency of 14.250 MHz. Turn on the two-tone generator and set the TONE B switch to ON. With the DUT in the USB mode, adjust the MIC gain and set the generator LEVEL and ATTENUATOR controls for the maximum audio input as specified by the manufacturer. Observe the transmitter output power as shown by the wattmeter. Unkey the transmitter and set the step attenuators for -40 dBm input to the spectrum analyzer.

> *CAUTION: The input to the spectrum analyzer at no time should exceed +10 dBm. Damage to the instrument will occur at an input level of +30 dBm or greater.*

4.6.3.4 Key the transmitter. Adjust the DUT USB signal frequency to 14.2532 MHz on the spectrum analyzer display.

4.6.3.5 A tiny low level pip may appear 1900 Hz to the left of the USB signal at the center of the spectrum analyzer display. This is the suppressed carrier. Another pip, if visible 3800 Hz to the left of the USB (at 14.2494 MHz) is the unwanted Lower Sideband (LSB). This may be verified by keying the transmitter off and on again. All pips from the DUT will disappear. If necessary, adjust the CENTER FREQ control so that the USB pip is centered (or nearly centered) on the display graticule.

4.6.3.6 Adjust the REF LEV control (and step attenuators if required) so that the peak of the USB pip is at 0 dB. The spectrum analyzer is now calibrated. Press the SGL SWEEP button for a single trace. The amplitude of the unwanted sideband and carrier may now be read in dB below Peak Envelope Power (PEP) directly from the display.

4.6.3.7 Record the value of both the suppressed carrier and the unwanted sideband in 4.6.3.7 of the Data Sheet.

4.6.3.8 Set the transmitter mode switch for LSB. Return the SWP TIME to AUTO. Repeat paragraphs 4.6.3.4 to 4.6.3.7 for the lower sideband at 14.2494 MHz (the suppressed carrier and unwanted upper sideband now appear to the right of the lower sideband at 14.2513 and 14.2532 MHz, respectively). If the pip is lost when the mode is changed, it may also be necessary reset the CENTER FREQ control. Record results in 4.6.3.8 of the data sheet.

4.6.3.9 If the DUT also has any VHF or UHF outputs, set the DUT to approximately 200 kHz above the bottom of each band and repeat paragraphs 4.6.3.3 to 4.6.3.8, adjusting the spectrum analyzer center frequency as appropriate (Examples: for 6M, use a center frequency of 50.2013; the USB signal will appear at 50.2032 and the LSB signal at 50.1994; for 2M, use a center frequency of 144.2013; the USB signal will appear at 144.2032 and the LSB signal at 144.1994).

## 4.7 CW KEYING WAVEFORM TEST

4.7.1 The purpose of the CW Keying Waveform Test is to photograph the first and second dit in a series of dits with the DUT in the VOX and QSK modes. The keying rate is 20 ms on and 20 ms off, a rate that corresponds to 60 WPM using the PARIS standard. A picture will also be taken of any other test conditions that result in any wave shape that is significantly different from the others (more than 10% difference, spikes, etc.).

4.7.2 Test hook-up (See Fig. 4-7)

> **NOTE**: If proceeding from the previous SSB Carrier and Unwanted Sideband Test, only the hook-up modifications indicated with a dotted line need to be implemented for this test.

4.7.2.1 With all power switches in the OFF position and the transceiver in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUTPUT to wattmeter INPUT | AS Required to type N | 50-Ohm coax |
| Wattmeter OUTPUT to Power Attn INPUT | Type N to type N | 50-Ohm coax |
| Power Attn OUT to 10-dB step INPUT | Type N to BNC | 50-Ohm coax |
| 10-dB step Attn OUT to 1-dB step IN | BNC to BNC | 50-Ohm coax |
| 1-dB step Attn OUT to CH2 Scope IN | BNC to BNC | 50-Ohm coax |
| Key Test Gen OUT to Scope EXT TRIG IN | BNC to BNC | 50-Ohm coax |
| Key Test Gen KEY OUT to DUT KEY IN | BNC to As Required | 50-Ohm coax |
| XMTR Key Line to Scope CH1 INPUT | Probe clip to BNC | 10X Probe |
| Scope GND to XMTR GND | As Required | As Required |
| Power to DUT | As Required | As Required |

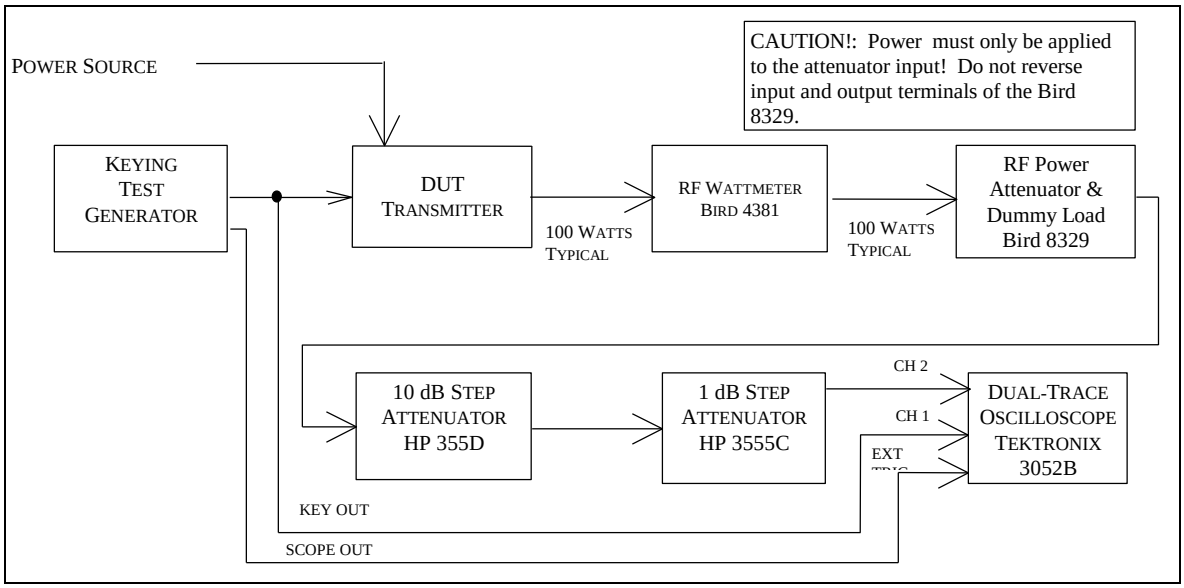



Fig. 4-7 — CW Keying Waveform Test Hook-up

4.7.3 Test Procedure

4.7.3.1 Turn the DUT, RF wattmeter and oscilloscope power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | 20 Meters |
| | Frequency | 14.020 MHz |
| | XMIT/RCV | Receive |
| | Drive or RF Level | Maximum |
| | Keying Mode | VOX |
| RF Wattmeter | Push-button Mode Select | FWD CW |
| | Element | As Required |
| | Forward Element Range | As Required |
| 10-dB Step Attn | Attenuator | –40 dB |
| 1-dB Step Attn | Attenuator | Any |
| Keying Test Generator | OUTPUT | OFF |
| | RANGE | 1-99 MS |
| | KEY DOWN | 20 |
| | KEY UP | 20 |
| Oscilloscope | CH1 VOLTS/DIV | 500 mV |
| | CH1 Coupling | DC |
| | CH1 Position | As Needed |
| | CH2 VOLTS/DIV | 1 V |
| | CH2 Coupling | DC |
| | CH2 Position | As Needed |
| | TIME/DIV | 10 MS |
| | Trigger Coupling | DC |
| | Trigger Source | External |
| | Trigger Level | As Required |

4.7.3.2 Set the keying test generator to ON. Allow all equipment at least 10 minutes warm-up time before proceeding.

4.7.3.3 Tune the DUT, if necessary, *exactly* per the operator's manual for the 14.250 MHz test frequency. Set for maximum power output.

4.7.3.4 Set the keying test generator OUTPUT to ON. Adjust the oscilloscope for a display trace similar to Fig. 4-7A. Note: If the keying generator pulse train shows capacitive shaping, this is due to the configuration of the DUT's keying input circuit and should be considered normal. Be sure that the RF-power level of the dit is approximately the same as the carrier level.

4.7.3.5 Capture the first dit and second dit in a string of dits by using the single-trigger mode of the oscilloscope. Turn the keying generator OFF then ON again to facilitate the string of dits. (NOTE: Some transmitters will lose or chop the first dit of a word).

4.7.3.7 Repeat steps 4.7.3.5 and 4.7.3.6 for QSK ON, and any other keying mode deemed appropriate by the Test Engineer. Also photograph the results of any mode of operation that results in a wave shape that is significantly different (>10% difference, spikes, etc.).

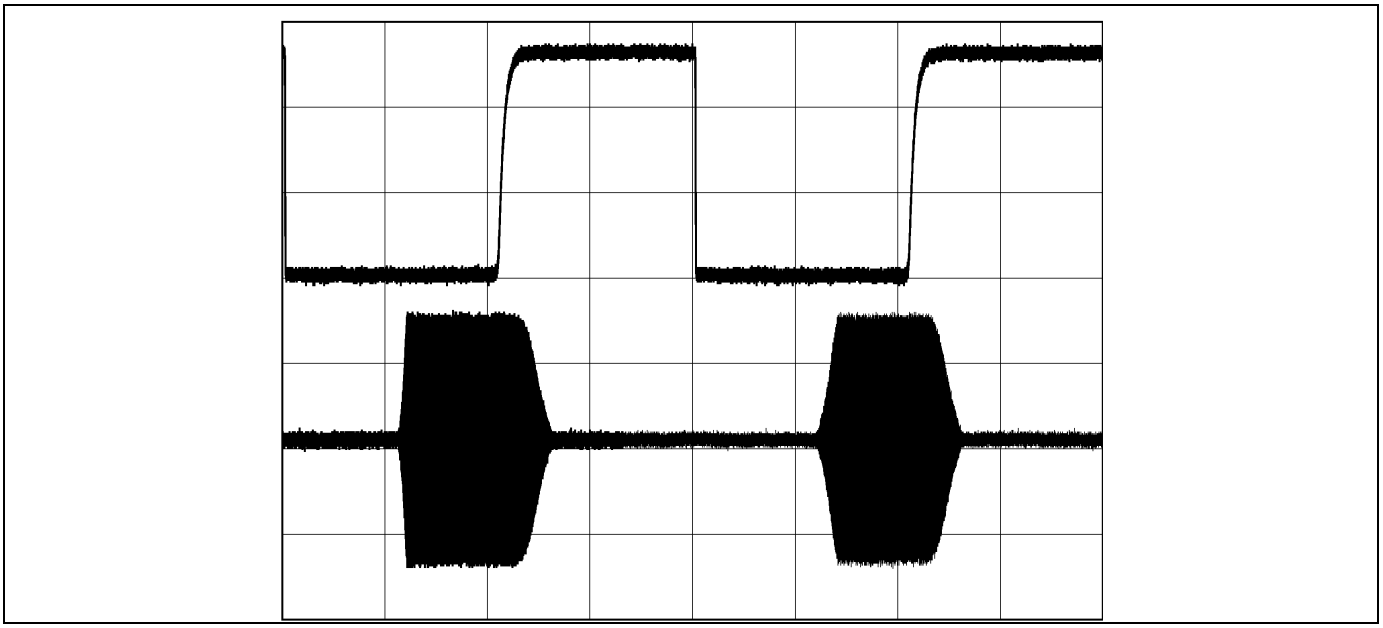


Fig. 4-7A — Oscilloscope Trace of RF Output Envelope

## 4.8 PTT TO SSB/FM RF OUTPUT TEST

4.8.1 The purpose of the PTT to SSB/FM RF Output Time Test is to determine the key time to RF output delay for the waveform 50% points in the SSB and FM modes. In the SSB mode, audio at a frequency of 700 Hz, and a level within the manufacturer's specified limits, will be applied to the DUT's mic input terminals. The FM mode is tested with an unmodulated carrier.

4.8.2 Test hook-up (See Fig. 4-8)

**NOTE**: If proceeding from the previous CW Keying Waveform Test, only the hook-up modifications indicated with a dotted line need to be implemented for this test.

4.8.2.1 With all power switches in the OFF position and the transmitter in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUTPUT to Wattmeter INPUT | As Required to type N | 50-Ohm coax |
| Wattmeter OUTPUT to Power Attn INPUT | Type N to type N | 50-Ohm coax |
| Power Attn OUT to 10-dB step INPUT | Type N to BNC | 50-Ohm coax |
| 10-dB step Attn OUT to 1-dB step IN | BNC to BNC | 50-Ohm coax |
| 1-dB step Attn OUT to CH2 Scope IN | BNC to BNC | 50-Ohm coax |
| Key Test Gen OUT to Scope EXT TRIG IN | BNC to BNC | 50-Ohm coax |
| Two-Tone Gen OUT to DUT MIC INPUT | Banana to As Required | Coax |
| Key Test Gen Key OUT to DUT PTT IN | BNC to As Required | 50-Ohm coax |
| XMTR PTT Key Line to Scope CH1 IN | BNC to BNC | 10X Probe |
| Power to DUT | As Required | As Required |
| Scope GND to XMTR GND | As Required | As Required |

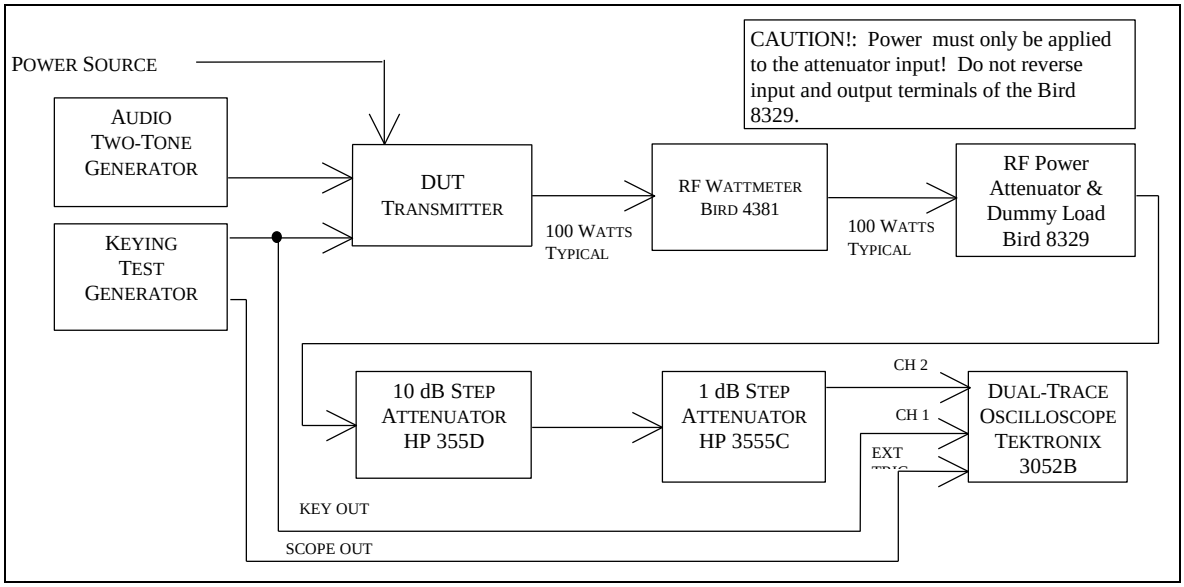



Fig. 4-8 — PTT SSB/FM RF Output Test Hook-up

4.8.3 Test procedure

4.8.3.1 Turn the DUT, RF wattmeter and oscilloscope power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | USB |
| | Band Selector | 20 Meters |
| | Frequency | 14.250 MHz |
| | XMIT/RCV | Receive |
| | DRIVE or RF LEVEL | Minimum |
| | VOX | Off |
| RF Wattmeter | Mode Select Button | FWD PEP |
| | Element | As Required |
| | Forward Element Range | As Required |
| 10-dB Step Attn | Attenuator | –40 dB |
| 1-dB Step Attn | Attenuator | Any |
| Keying Test Generator | OUTPUT | OFF |
| | RANGE | 1-99 MS |
| | KEY DOWN | 20 |
| | KEY UP | 20 |
| Two-Tone Generator | TONE A | ON |
| | TONE B | OFF |
| | HI-Z/LO-Z | As Required |
| | BALANCE | N/A |
| | LEVEL | Full CCW |
| | ATTENUATOR | –30 dB |
| Oscilloscope | CH1 VOLTS/DIV | 500 mV |
| | CH1 Coupling | DC |
| | CH1 Position | As Needed |
| | CH2 VOLTS/DIV | 1 V |
| | CH2 Coupling | DC |
| | CH2 Position | As Needed |
| | TIME/DIV | 5 MS |
| | Trigger Mode | AUTO |
| | Trigger Coupling | DC |
| | Trigger Source | EXT |
| | Trigger Level | As Required |

4.8.3.2 Set the Two-Tone and Keying Test Generator power switches to ON. Receiver hiss should be heard; adjust volume to desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 4.8.3.3

4.8.3.3 Tune the DUT transmitter, if necessary, *exactly* per the operator's manual for the 14.250 MHz test frequency. Adjust the two-tone generator OUTPUT level to the manufacturer's specified limit. Set the DUT for maximum rated power output.

4.8.3.4 Turn the keying test generator OUTPUT to ON. Adjust the oscilloscope for a display trace similar to Fig. 4-8A. It may be necessary to reduce the generator keying speed. Be sure that the RF power level of the Dit is approximately the same as the carrier level.

4.8.3.5 Measure the ON and OFF delay times between the 50% points of the keying pulse and the RF output pulse. Take a single sweep. Adjust the oscilloscope's sweep rate and horizontal position for optimum display results. It is often best to determine the on and off delay times from two different traces. The Trigger Level adjustment can be used to facilitate an easy transition between ON and off delay traces. Record both values in 4.8.3.5 of the data sheet. Turn off the two-tone and keying test generator outputs.

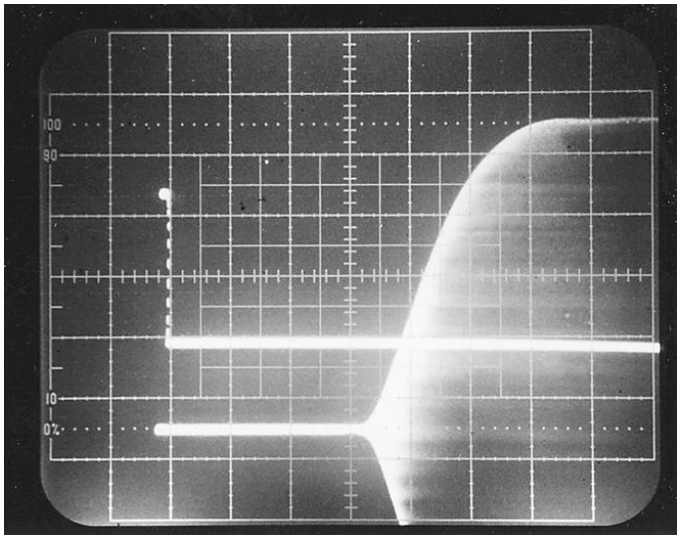


Fig. 4-8A — Oscilloscope Trace of RF Envelope

4.8.3.6 Place the DUT in the FM mode. Repeat paragraphs 4.8.3.3, 4.8.3.4 and 4.8.3.5 for this mode at a test frequency of 29.000 MHz . Record the results in 4.8.3.6 of the data sheet.

4.8.3.7 Repeat step 4.8.3.6 for test frequencies of 52.000 MHz, 146.000 MHz and 440.000 MHz, as appropriate.

**4.9 TRANSMIT/RECEIVE TURN-AROUND TIME TEST**
**(Transceivers Only)**

4.9.1 The purpose of the Transmit/Receive Turn-Around Time Test is to measure the time delay required for a transceiver to switch from transmit to the receive mode.

4.9.2 Test hook-up (See Fig. 4-9)

4.9.2.1 With all power switches in the OFF position and the transceiver in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUTPUT to wattmeter INPUT | As Required to type N | 50-Ohm coax |
| Wattmeter OUTPUT to Power Attn INPUT | Type N to type N | 50-Ohm coax |
| Power Attn OUT to 10-dB step INPUT | Type N to BNC | 50-Ohm coax |
| 10-dB step Attn OUT to 1-dB step IN | BNC to BNC | 50-Ohm coax |
| 1-dB step Attn IN to Signal Gen RF OUT | BNC to Type N | 50-Ohm coax |
| DUT Audio OUT to Scope CH2 INPUT | As Required | 10X or 1X Probe |
| Key Gen Scope OUT to Scope EXT TRIG IN | BNC to BNC | 50-Ohm coax |
| Key Test Gen Key OUT to PTT Key IN | BNC to As Req | 50-Ohm coax |
| XMTR PTT Line to Scope CH1 INPUT | Probe clip to BNC | 10X Probe |

| Connection | Connectors | Cable Type |
|---|---|---|
| Two-Tone Gen OUT to DUT MIC INPUT | Banana to As Required | Coax |
| Power to DUT | As Required | As Required |
| Scope GND to XMTR GND | As Required | As Required |

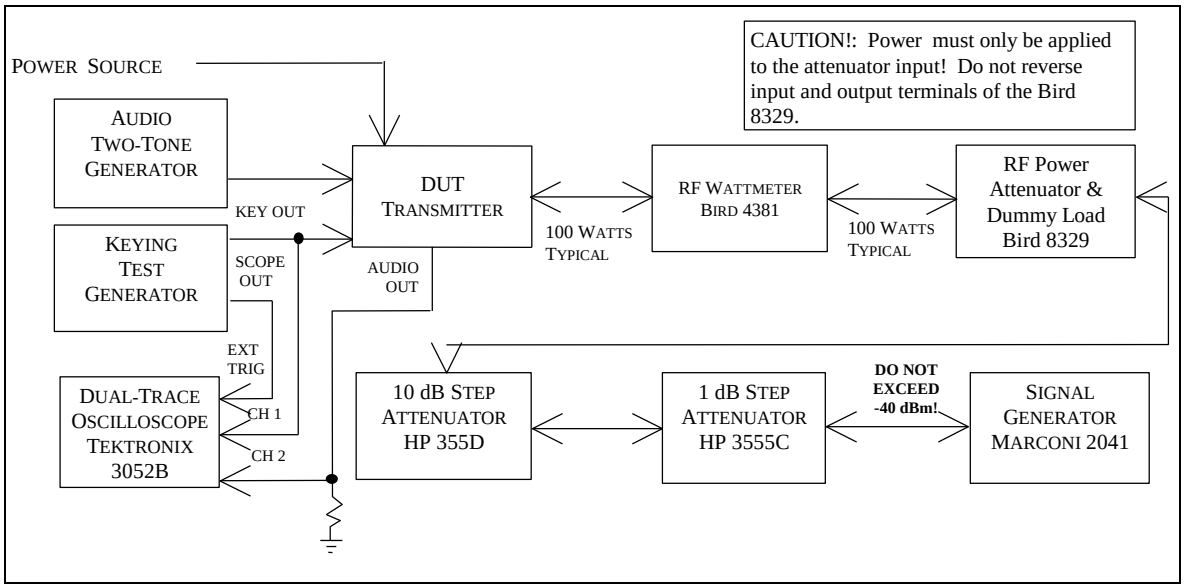



Fig. 4-9 — Transmit/Receive Turnaround Time Test Hook-up

4.9.3 Test procedure

4.9.3.1 Turn the DUT, RF wattmeter and oscilloscope power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | USB |
| | Frequency | 14.250 MHz |
| | XMIT/RCV | Receive |
| | AGC | ON (Fastest available) |
| | Squelch | OFF |
| RF Wattmeter | Push-button Mode Select | FWD PEP |
| | Element | As Required |
| | Forward Element Range | As Required |
| 10-dB Step Attn | Attenuator | 120 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| Keying Test Generator | OUTPUT EN | OFF |
| | RANGE | 1-99 ms |
| | KEY DOWN | 50 ms |
| | KEY UP | 50 ms |
| Oscilloscope | CH1 VOLTS/DIV | 0.5 V |
| | CH1 Coupling | DC |
| | CH2 VOLTS/DIV | 1 V |
| | CH2 Coupling | DC |
| | TIME/DIV | 5 ms |
| | Trigger Coupling | DC |
| | Trigger Source | EXT |
| Two-Tone Generator | TONE A | ON |
| | TONE B | OFF |
| | HI-Z/LO-Z | As Required |
| | BALANCE | N/A |
| | LEVEL | FULL CCW |
| | ATTENUATOR | –30 dB |
| RF Generator | FREQUENCY TUNE | 14.250 MHz |
| | OUT LEVEL | +10 dBm |
| | RF | ON |
| | AM | OFF |
| | FM | OFF |

4.9.3.2 (If proceeding from the previous PTT to SSB RF Output Time Test, this paragraph may be skipped). Set the Two-Tone and Keying Test Generator power switches to ON. Receiver hiss should be heard; adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 4.9.3.3.

*CAUTION: Reverse power to the RF signal generator at no time should exceed -40 dBm (approx. 2 $Vr_{ms}$). Shut-down of the instrument may result.*

4.9.3.3 Reduce the step attenuators until the signal is heard and a deflection can be observed on the S meter. Tune the receiver for maximum S meter indication. Reduce the RF step attenuators until the receiver S meter indicates an S9 signal. If the DUT does not have an S Meter, then set for a 50 μV input to the DUT.

4.9.3.4 Tune the transmitter, if necessary, exactly per the operator's manual for the 14.250 MHz test frequency. Turn on the two-tone generator and set the A tone switch to ON. With the DUT in the USB mode, set the generator LEVEL and ATTENUATOR controls for the maximum audio input level as specified by the DUT manufacturer. specified by the manufacturer. Unkey the transmitter and turn the keying test generator 4.9.3.5 Key the transmitter and set for the maximum power as ON. Adjust the oscilloscope settings (including sweep rate) as required for an optimum display trace as shown in Fig. 4-9A. If the audio output is not visible at all on the oscilloscope it will be necessary to increase the keying test generator KEY-UP and KEY-DOWN time. Measure and record the time it takes to go from PTT key-up to 50% audio output. Set the keying test generator to OFF.

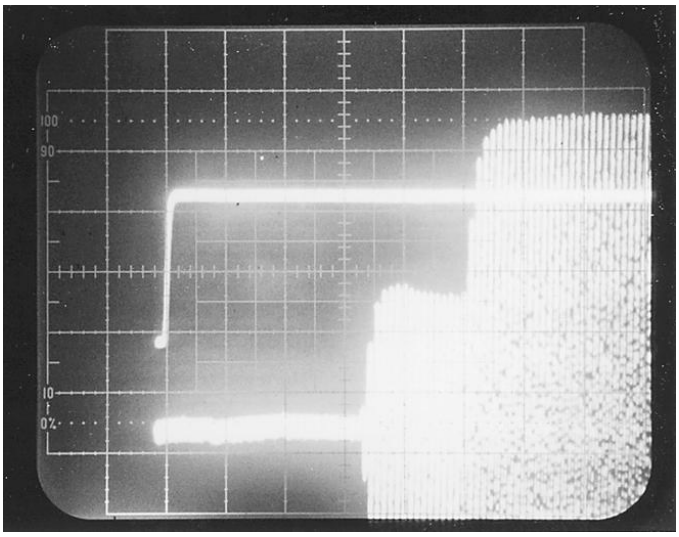


Fig. 4-9A — Oscilloscope Trace of RF Envelope

4.9.3.6 Repeat step 4.9.3.5 for other AGC options, if available, and a 20 dB decrease in signal strength. Check the appropriate box in 4.9.3.6 of the data sheet if no significant deviation is observed. Record the space provided any significant deviation.

4.9.3.7 AMTOR requires the transceiver audio to reach 50% of its full value in 35 ms or less. Determine the AMTOR suitability and record on the data sheet.

**4.10 KEYER SPEED AND SIDETONE FREQUENCY TEST**
**(For units having an internal keyer.)**

4.10.1 The purpose of this test is to measure the transmitter's internal keyer speed range and sidetone frequency range.

4.10.2 Test hook-up (See Fig. 4-10)

4.10.2.1 With all power switches in the OFF position and the transceiver in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUTPUT to wattmeter INPUT | As Required to type N | 50-Ohm coax |
| Wattmeter OUTPUT to Power Attn INPUT | Type N to type N | 50-Ohm coax |
| Power Attn OUT to Scope CH1 INPUT | BNC to Type N | 50-Ohm coax |
| DUT Audio OUT to Scope CH2 INPUT | As Required | 50-Ohm coax |
| DUT Audio OUT to 10 dB step Attn IN (10W or greater power rating) | As Required to BNC | 50-Ohm coax |
| 10 dB step Attn OUT to Freq Counter IN | BNC to BNC | 50-Ohm coax |
| DUT Audio OUT to HI-Z Amp / 8 Ohm load | As Required to BNC | 50-Ohm coax |
| XMTR Key Input to Telegraph Key | As Required to Test Clips | Any |
| Power to DUT | As Required | As Required |

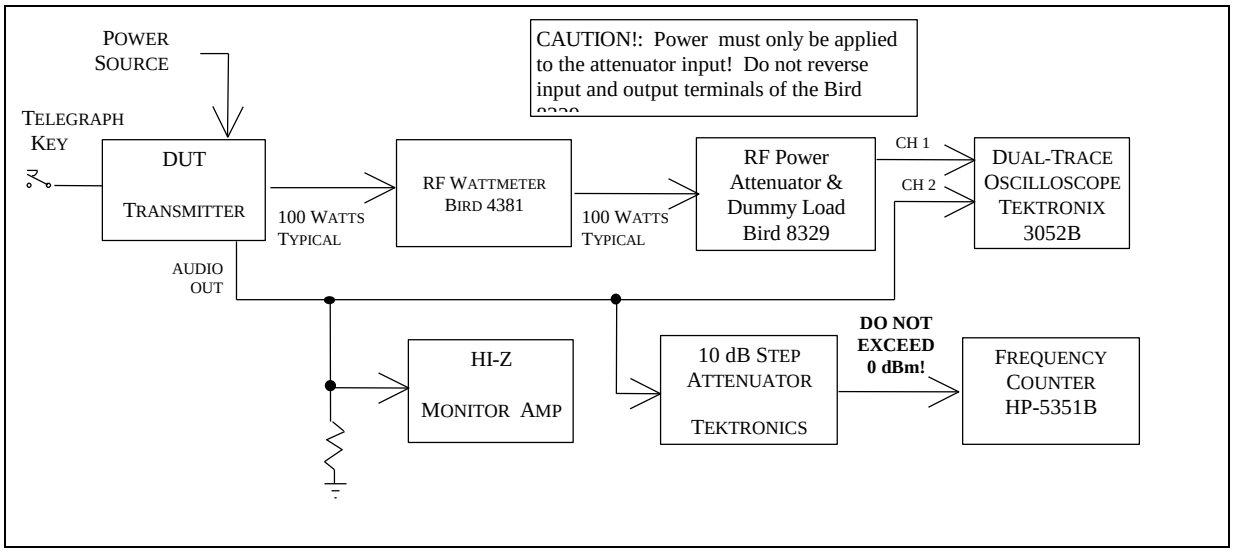



Fig. 4-10 — Keyer Speed and Sidetone Frequency Test Hook-up

4.10.3 Test procedure

4.10.3.1 Turn the DUT, RF wattmeter and oscilloscope power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|

| | | |
|---|---|---|
| DUT | Mode | USB |
| | Frequency | 14.250 MHz |
| | XMIT/RCV | Receive |
| | AGC | N/A |
| | Squelch | OFF |
| RF Wattmeter | Push-button Mode Select | FWD PEP |
| | Element | As Required |
| | Forward Element Range | As Required |
| 10-dB Step Attn | Attenuator | –40 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| Oscilloscope | CH1 VOLTS/DIV | 0.5 V |
| | CH1 Coupling | DC |
| | CH1 Position | As Needed |
| | CH2 VOLTS/DIV | 1 V |
| | CH2 Coupling | DC |
| | CH2 Position | As Needed |
| | TIME/DIV | 5 ms |
| | Trigger Coupling | DC |
| | Trigger Source | CH1 |
| | Trigger Level | As Required |

4.10.3.2 (If proceeding from the previous Turnaround Time Test, this paragraph may be skipped). Receiver hiss should be heard; adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 4.10.3.3.

4.10.3.3 Key the transmitter. Set the sidetone level for comfortable listening level. Adjust the step attenuators and oscilloscope controls for optimum display. Send a series of dits and then a series of dashes. Verify the dashes are three times longer in duration than the dits. Measure and record any significant deviation from the standard 3 to 1 ratio.

4.10.3.4 With keyer speed still set to minimum, send a series of dits and take a trace in the storage mode. Measure the time duration from the leading edge of a dit to the leading edge of the next. This duration is equivalent to two dits (one dit plus a dit space). Adjust the oscilloscope controls as required to obtain an optimum trace. Divide the measured time by (in ms) into 2400 to obtain the equivalent code speed (in WPM):

Code Speed in WPM = 1200 / Duration of Single Dit in ms  Record on Data Sheet.

4.10.3.5 Set keyer speed to maximum and repeat step 4.10.3.4.

4.10.3.6 Set keyer speed to center or the default position and repeat step 4.10.3.4.

4.10.3.7 Adjust the Frequency Counter Step Attenuators for minimum positive indication. Measure the initial default sidetone frequency . Reduce the sidetone for minimum frequency and measure again. Increase to maximum sidetone frequency and repeat. Record both all three frequencies on Data Sheet.

## 4.11 COMPOSITE NOISE TEST

4.11.1 The purpose of the Composite-Noise Test is to observe and measure the phase and amplitude noise, as well as any spurious signals generated by the DUT transmitter. Since phase noise is the primary noise component in any well-designed transmitter, it can be assumed, therefore, that almost all the noise observed during this test is phase noise.

4.11.1.1 This measurement is accomplished with the use of an integrated HP model 11848A phase noise test set that consists of a control box (phase discriminator, phase shift network, DC tuning voltage control, and analyzer outputs), HP 3561A FFT-based dynamic signal analyzer, HP 3563E (or 3562A) microwave spectrum analyzer, and as needed, an oscilloscope (for assistance in establishing zero beat). Measurements are accomplished by computer software run on the 486 computer. Additionally, an appropriate low-noise reference oscillator that can be tuned via DC is also necessary.

4.11.2 Test Hook-up (See Fig. 4-11).

4.11.2.1 With all power switches in the OFF position and the transceiver in the receive mode, connect the following:

| Connection | Connectors | Cable Type* |
|---|---|---|
| DUT RF OUTPUT to wattmeter INPUT | As Required to type N | 50-Ohm coax |
| Wattmeter OUTPUT to Power Attn INPUT | Type N to type N | 50-Ohm coax |
| Power Attn OUT to 10-dB step INPUT | Type N to BNC | 50-Ohm coax |
| 10-dB step Attn OUT to 1-dB step IN | BNC to BNC | 50-Ohm coax |
| 1-dB step Attn OUT to HP-11848A to "Right" Phase Detector IN | BNC to BNC | 50-Ohm coax |
| Ref. Oscillator OUT to HP-11848A "Left" Phase Detector IN | BNC to BNC | 50-Ohm coax |
| HP-11848A Tuning voltage to Test | BNC to BNC or clip leads | As Required |
| HP-11848A Phase Detector Output to HP-3561A FFT Signal Analyzer | BNC to BNC | 50-Ohm coax |
| HP-11848 Spectrum Analyzer OUT to HP-8563 E Spectrum Analyzer | BNC to BNC | 50-Ohm coax |
| Power to DUT | As Required | As Required |
| HP-11848A Aux Monitor OUT to TDS-3052B Oscilloscope CH 1 IN | BNC to BNC | 50-Ohm coax |

*NOTE: Composite noise setup is very sensitive to environment noise. Use of common-mode ferrite chokes may be required if test fixture problems are encountered as described in the procedure for this test.

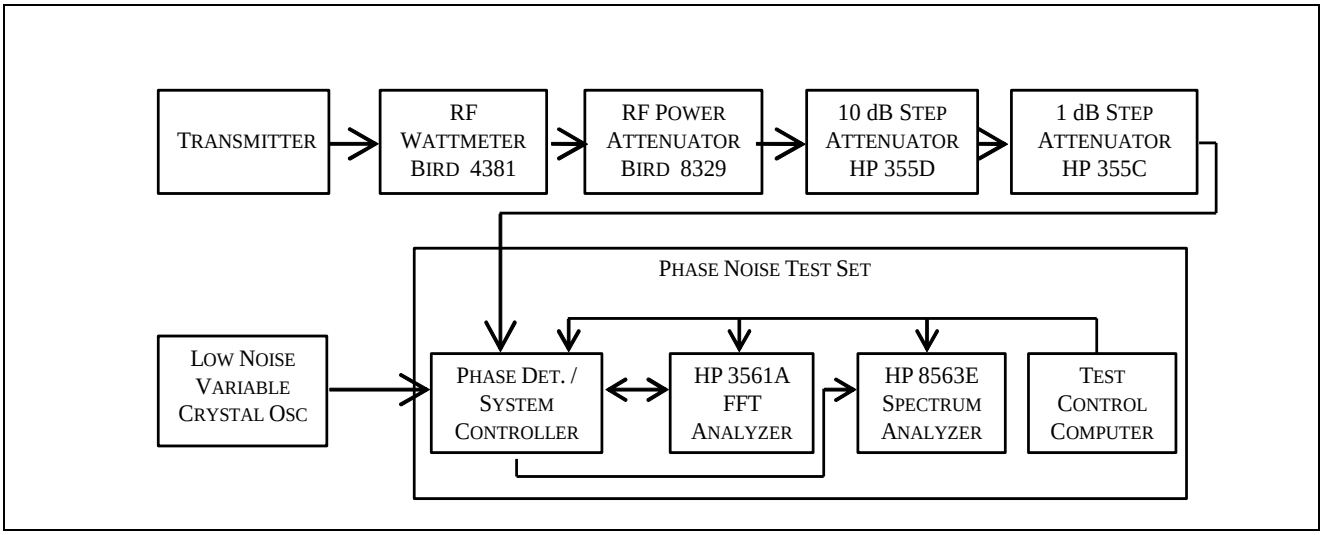



Fig. 4-11 — Composite Noise Test Hook-up

4.11.3 Test procedure

4.11.3.1 Turn the DUT, RF wattmeter, test system, spectrum analyzer power, and reference oscillator switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | 20 Meters |
| | Frequency | 14.025 MHz |
| | Output Power | 30-50 % |
| RF Wattmeter | Push-button Mode Select | FWD CW |
| | Element | As Required |
| | Forward Element Range | As Required |
| 10-dB Step Attn | Attenuator | As required for +15 dBm |
| 1-dB Step Attn | Attenuator | Any |
| Reference Oscillator | FREQUENCY | 14.025 MHz |

4.11.3.2 Allow all equipment, including the test oscillator, at least 15 minutes warm-up time before proceeding. TURN OFF all non essential test equipment and overhead fluorescent lights. Use the incandescent lighting.

4.11.3.3 Check the manufacturer's specification of the transmitting duty cycle. If not stated, set the power output to 30 watts for a 100 watt maximum transmitter, or 50 watts for a 200 watt maximum transmitter. The test will take about 15 minutes; do not exceed manufacturer's duty cycle

4.11.3.4 Turn on the control computer, allow it to complete the boot and program start processes, then step through the menu system to set the initial test frequency and VFO tune ranges. On the menu, the test frequency has a scientific format; example: 14.025E+6 (14.025 x 10 to the $6^{th}$ power). For the Wenzel crystal oscillators, use 14.025 MHz, a VCO tune center voltage of approximately 5 and a tune voltage range of +/-5 V. For the Marconi/IFR 1041 signal generators, set the tune center voltage to 0 and a tuning range +/-5V. All other parameters should be ok at their initial values.

4.11.3.5 Key the transmitter and adjust the frequency of the DUT to be approximately 1 Hz or less from the frequency of the reference oscillator. Use an oscilloscope , if necessary, to remove any undulating shape on the screen of the noise output. The two frequencies must not be equal or a phenomenon known as phase roll may result, and, if much greater than 2 Hz apart, the software will not run and will show an error message. The tuning voltage of the test oscillator may be adjusted slightly in the software if the transmitter tuning cannot be brought less than 1 Hz. The idea is to get as close as possible, without being identical. The signal should be unreadable and only the noise clearly visible on the analyzer display before proceeding to the next step. Un-key the transmitter.

4.11.3.6 To measure the Phase Noise, key the transmitter and run the program. After about three minutes, you should see a graph of the composite noise forming on the PC screen. After 15 minutes or so, the process will be complete and the file is saved as a type .PCX file. Before saving the file, press "M" and omit the spurs (red lines on the graph). Copy the file to disc.

4.11.3.7 Repeat steps 4.11.3.4 through 4.11.3.6 with a test frequency of 50.020 MHz. With this test, you will replace the Test Oscillator with the IFR 2041 Signal Generator in the "Low Noise" mode. Set the following controls on the IFR2041:

External 1 to "DC Coupling"
Deviation to 3 kHz, MODULATION ENABLED
RF Level to +13 dBm
Frequency to 50.020
RF Output to "Left" Phase Detector Input
External 1 to Tune Voltage Output

Be sure to change the following on the software menu:

Frequency to 50.020E+6
Detector Frequency is + 6 volts
Center Tune Volts is 0.0 volts
Tune Range is 6 volts.

Near zero beat using the scope and run the program. Save file to disc.

## V. RECEIVER TESTS

5.0.1 As shown in the Table of Contents, there are 17 Receiver tests outlined in this chapter. They have been arranged, just as in the previous transmitter tests, to minimize the required level and frequency of hook-up changes and modifications. Each hook-up, however, is shown complete with all changes from the previous test clearly indicated. A block diagram accompanies each hook-up and any changes from the previous test are shown within a dotted rectangle. This affords the flexibility to easily start anywhere and perform these tests in any desired order.

5.0.2 Before performing any receiver testing, it is essential for the test engineer to be completely familiar with the Device Under Test (DUT) and the test equipment that will be used. The RF output of a transceiver will damage the test equipment and therefore, *it is essential never to accidentally key the transmitter while performing these tests!* Completely disable the transmitter if at all possible and set all RF output controls to minimum, reduce drive, disconnect the telegraph key, etc. The receiver must be operated in a manner exactly as specified by the manufacturer. Any test that would cause equipment to be operated in a manner inconsistent with its operating manual must be modified accordingly.

5.0.2.1 Other considerations are as follows:

1) The RF Generator output must not exceed –14 dBm when using the Hybrid Combiner. This restriction is an effort to minimize intermodulation in the combiner and signal generators.

2) The level of input to the receiver should not exceed +10 dBm or the manufacturer's specification, whichever is less.

3) Read and understand all pertinent manuals before operating any laboratory instrument or the DUT receiver.

4) The testing should be conducted in a relatively RF free environment. If an RF source is in close proximity to the testing site, an RF screen room should be used.

5) If a preamp is available, each test must be performed with each level of the preamp and without the preamp.

6) If the audio output impedance is not 8 Ohms, select the appropriate load resistance on the HI-Z Audio Amplifier.

## 5.1 NOISE FLOOR TEST

5.1.1 The purpose of the Noise Floor Test (also known as "Minimum Discernible Signal" or MDS) is to determine the level of signal input to the receiver that will produce an audio output where the power in the signal is equal to the power in the noise (S + N = N + 3 dB). The test is conducted with the receiver in the CW mode using the 500 Hz, or closest available IF filter (or audio filters where IF filters are not available. For receivers that have appropriate IF filters, all audio filtering is disabled.) Set the AGC to the OFF position if possible. The test is performed frequencies of 1.020 MHz, 3.520 MHz, 14.020 MHz, 50.020 MHz, 144.020 MHz and 432.020 MHz. For the expanded set of tests, this test is performed on all available amateur bands, 20 kHz above the lower band edge.

5.1.2 Test hook-up (See Fig. 5-1)

5.1.2.1 With all power switches in the OFF position, the transmitter function disabled to the fullest extent possible and the Generator RF switch OFF, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen OUTPUT to 10-dB Step Attn INPUT | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50-Ohm Coax |
| DUT AUDIO OUT to Dist/Audio Meter IN | As Required to BNC | 50-Ohm Coax |
| 8-Ohm Load/HI-Z Amp Across Dist/Audio Meter Input | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

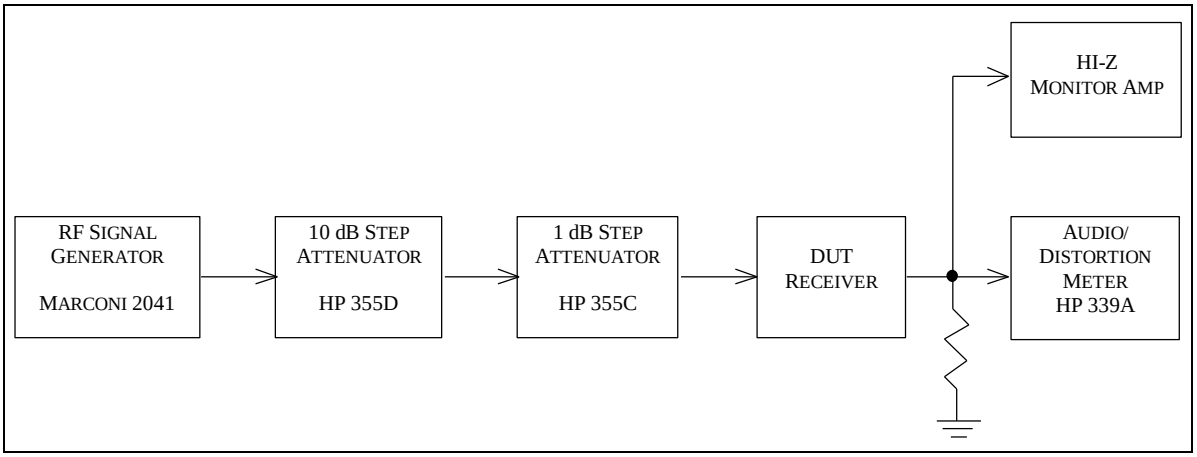



Figure 5-1 -- CW MDS Test Hook-up

5.1.3 Test Procedure

5.1.3.1 Turn the DUT and all test equipment power switches to ON. If the DUT does not cover 1.020 MHz, proceed directly to the second test frequency of 3.520 MHz. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | As Required for 1.020 MHz |
| | Frequency | 1.020 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | 500 Hz or Closest Available |
| | AGC | OFF |
| | Preamplifier | OFF |
| 10 dB Step Attn | Attenuator | 0 dB |
| RF Generator | CARRIER FREQ | 1.020 MHz |
| | RF LEVEL | –110 dBm |
| | NOISE MODE (UTIL) | LOW NOISE |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All Off (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.1.3.2 Receiver hiss should be heard. Adjust the volume of the DUT and HI-Z monitor amp to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.1.3.3.

5.1.3.3 Verify the RF Generator Output frequency has remained at 1.020 MHz. Reset if necessary.

5.1.3.4 Tune the receiver for 1.020 MHz. Adjust the INPUT RANGE and RELATIVE ADJUST controls as required to maintain approximately a mid-scale meter indication while carefully tuning the receiver for peak signal response. (Rotate the Generator OUTPUT LEVEL control as required until the signal is just heard in the receiver.)

5.1.3.5 Set the RF Generator RF switch to OFF. Decrease the Audio Meter INPUT RANGE until the meter indication is near mid-scale and the two lights above this control are out. Adjust the audio meter RELATIVE ADJUST control until the audio meter reads –6 dB on the upper scale. (Adjust the INPUT RANGE control one step in either direction if necessary.)

5.1.3.6 Set the RF Generator RF switch to ON. Rotate the generator OUTPUT LEVEL control to produce an audio meter reading of –3 dB. Ensure that the DUT is tuned for peak response.

5.1.3.7 Check to ensure that the Generator frequency is still at 1.020 MHz. Set the RF Generator RF to OFF. The audio meter should return to –6 dB. Turn the RF back on and the meter should again indicate –3 dB.

5.1.3.8 Determine the noise floor (MDS) of the receiver by computing the sum of the RF generator output in dBm and the 10 dB step attenuator. (NOTE: Be sure to include any additional attenuation you may have included in the line between the generator and the DUT.) Record on the Data Sheet.

Example: a) The RF generator is set for –128.6 dBm output.
b) The step attenuators are set for –10 dB.
c) The receiver MDS, therefore, is:

–128.6 – 10 = –138.6 dBm

5.1.3.9 Repeat steps 5.1.3.4 to 5.1.3.8 with DUT preamplifier set to ON.

5.1.3.10 Re-set the generator output level to –110 dBm. Repeat paragraphs 5.1.3.3 to 5.1.3.9 for a test frequency of 3.520 MHz.

5.1.3.11 Repeat step 5.1.3.10 for a test frequency of 14.020 MHz.

5.1.3.12 Repeat step 5.1.3.10 for test frequencies of 50.020 MHz, 144.020 MHz and 432.020 MHz, as applicable to the DUT.

5.1.3.13 For an expanded set of tests, repeat step 5.1.3.10 for all remaining amateur bands.

## 5.2 AM RECEIVE SENSITIVITY TEST

5.2.1 The purpose of the AM receive Sensitivity Test is to determine the level of an AM signal, 30% modulated at 1 kHz, that will produce a tone 10 dB above the noise level (MDS) of the DUT. Two frequencies, 1.020 MHz and 3.800 MHz are used for this test.

5.2.2 Test hook-up (See Fig. 5-2)

**NOTE**: If proceeding from the MDS Test, no hook-up changes are required. Proceed to step 5.2.3.

5.2.2.1 With all power switches in the OFF position and the transmitter function disabled to the fullest extent possible, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen OUTPUT to 10-dB Step Attn INPUT | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50-Ohm Coax |
| DUT Audio OUTPUT to Dist/Audio Meter INPUT | As Required to BNC | 50-Ohm Coax |
| 8 Ohm Load/HI-Z Amp Across Dist/Audio Meter INPUT | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

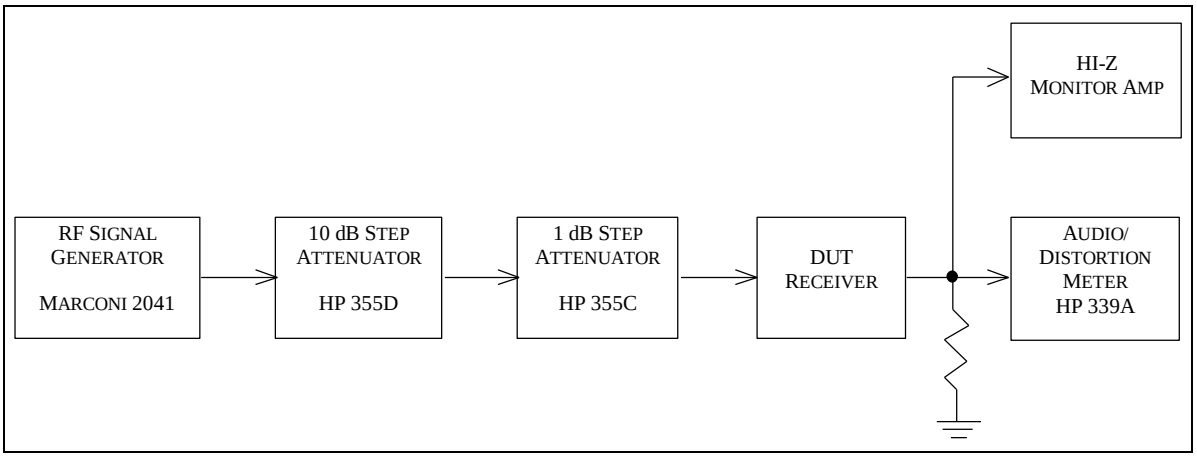



Fig. 5-2. AM Receive Sensitivity Test Hook-Up

5.2.3 Test Procedure

5.2.3.1 Turn the DUT and all test equipment power switches to ON. If the DUT does not cover 1.020 MHz, proceed directly to the second test frequency of 3.800 MHz. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | AM |
| | Band Selector | As Required for 1.020 MHz |
| | Frequency | 1.020 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | AM (Approx. 6 kHz) |
| | AGC | OFF |

| | | |
|---|---|---|
| | Preamplifier | OFF |
| | DUT Audio Filter(s) | Disabled |
| 10-dB Step Attn | Attenuator | 0 dB |
| RF Generator | CARRIER FREQ | 1.020 MHz |
| | RF LEVEL | –110 dBm |
| | CARR ON-OFF | ON |
| | AM | ON |
| | AM DEPTH | 30% |
| | Modulation Frequency | 1000 Hz |
| | FM | OFF |
| | NOISE MODE | LOW NOISE MODE |
| Audio/Distortion Meter | FUNCTION | REL. LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All OFF (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.2.3.2 Receiver hiss should be heard. Adjust the DUT and HI-Z monitor amp volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.2.3.3.

5.2.3.3 Tune the receiver for 1.020 MHz. Adjust the Audio Meter INPUT RANGE control as required to maintain approximately a mid-scale meter indication while carefully tuning the receiver for peak signal response. (If necessary, adjust the Generator OUTPUT LEVEL as required until the signal is just heard in the receiver.)

5.2.3.4 Set the RF Generator modulation control switch to OFF. Decrease the Audio Meter INPUT RANGE until the meter indication is near mid-scale and the two lights above this control are out. Adjust the audio meter RELATIVE ADJUST control until the audio meter reads –11 dB on the upper scale. (Adjust the INPUT RANGE control one step in either direction if necessary.)

5.2.3.5 Set the RF Generator RF switch to ON. Adjust the setting of the generator RF OUTPUT LEVEL to produce an audio meter reading of –1 dB +/-.5 dB. Recheck for peak response using the audio meter.

5.2.3.6 Check to ensure that the Generator frequency is still at 1.020 MHz. Set the RF Generator modulation to OFF. The audio meter should return to –11 dB. Turn the modulation back on and the meter should again indicate –1 dB.

**NOTE: The following measurement method is often better**, especially in cases where the DUT AGC cannot be disabled. The Audio Meter settings are the same as for the FM SINAD Test except the DISTORTION RANGE is set for 0-100% distortion. Adjust the signal generator output level for the following equivalent distortion level as required:

10 dB (S+N)/N = 31.6%

5.2.3.7 Determine the AM 10 dB sensitivity of the receiver by computing the sum of the output level of the RF generator, as read from the OUTPUT LEVEL switch and meter, and any additional attenuation in the line between the generator and the DUT. Record on the Data Sheet.

Example: a) The RF generator is set for –113 dBm output.
b) The step attenuators are set for 10 dB.
c) The receiver AM sensitivity is:
–113 - 10 = –123 dBm

5.2.3.8 Set the DUT preamplifier to ON and repeat steps 5.2.3.3 to 5.2.3.7.

5.2.3.9 Return the RF generator OUTPUT LEVEL control to –110 dBm and set for 3.800 MHz. Repeat paragraphs 5.2.3.3 to 5.2.3.8 for this test frequency.

## 5.3 FM SINAD AND QUIETING TEST

5.3.1 The purpose of the FM SINAD and Quieting Test is to determine the 12 dB SINAD value.
SINAD is an acronym for "Signal plus Noise and Distortion" and is a measure of signal quality.
The exact expression for SINAD is:

$$\text{SINAD} = \frac{\text{Signal + Noise + Distortion (expressed in dB)}}{\text{Noise + Distortion}}$$

If we consider distortion to be merely another form of noise, (distortion, like noise, is something unwanted added to the signal), we can further reduce the equation for SINAD to:

$$\text{SINAD} = \frac{\text{Signal + Noise (expressed in dB)}}{\text{Noise}}$$

If we now consider a practical circuit in which the signal is much greater than the noise, the value of the SIGNAL + NOISE can be approximated by the level of the SIGNAL alone. The SINAD equation then becomes the signal to noise ratio and can be approximated by:

$$\text{SINAD} = \frac{\text{Signal (expressed in dB)}}{\text{Noise}}$$

For 25% level of distortion, SINAD can be calculated as: SINAD = 20 log (1/25%) = 20 log 4 = 12 dB

5.3.2 Test hook-up (See Fig. 5-3)

**NOTE**: If proceeding from the previous AM Receive Sensitivity Test, no hook-up changes are required. Proceed directly to step 5.3.3.

5.3.2.1 With all power switches in the OFF position and the transmitter function disabled to the fullest extent possible, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen OUTPUT to 10-dB Step Atten INPUT | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50-Ohm Coax |
| DUT AUDIO OUTPUT to Dist/Audio Meter IN | As Required to BNC | 50-Ohm Coax |
| 8 Ohm Load/HI-Z Amp Across Dist/Audio Meter INPUT | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

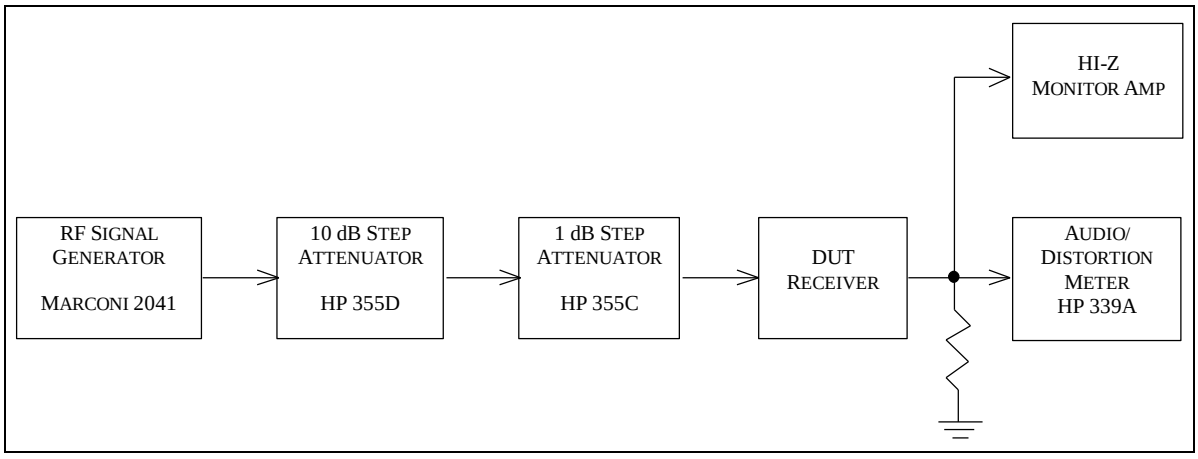



Fig. 5-3 FM SINAD and Quieting Test Hook-Up

5.3.3 Test Procedure

5.3.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | FM |
| | Frequency | 29.000 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | RF Gain | Maximum |
| | IF Filters | FM (Aprx.15 kHz) |
| | AGC | OFF |
| | Preamplifier | OFF |
| 10-dB Step Attn | Attenuator | 0 dB |
| RF Generator | CARRIER FREQUENCY | 29.000 MHz |
| | RF LEVEL | –110 dBm |
| | CARR ON-OFF | ON |
| | NOISE MODE | NORMAL[1] |
| | FM | On |
| | AM | Off |
| | FM DEVN | 3 kHz |
| | Modulation Frequency | 1000 HZ |

| | | |
|---|---|---|
| Audio/Distortion Meter | FUNCTION | DISTORTION |
| | RELATIVE ADJUST | Any |
| | METER | –10/30 |
| | FREQUENCY CONTROLS | 1.0 X 1 kHz |
| | FILTERS | Low Pass On |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.3.3.2 Receiver hiss should be heard. Adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.3.3.3.

5.3.3.3 Turn the DUT's preamp on and tune the receiver for 29.000 MHz. Rotate the Generator OUTPUT LEVEL control as required until the signal is just heard in the receiver. Set the DUT volume control for a normal listening level, and carefully tune the receiver for minimum distortion. (This control should be set roughly between 1/4 and 3/4 of the maximum limit.)

5.3.3.4 Gradually increase the Signal Generator output until the Distortion Meter indicates 25% distortion. (The step Attenuator may also be used, if necessary, to help achieve the reading.)

5.3.3.5 Measure the level of the input signal to the DUT receiver and record in 5.3.3.5 of the Data sheet. Be sure to account for any attenuation in the line from the step attenuators or hybrid combiner.

Example: a) The RF generator is set for –112.4 dBm output
b) The step attenuator is set for 10 dB
c) The FM SINAD is:

<u>–112.4 - 10 = –122.4 dBm</u>

5.3.3.6 Repeat steps 5.3.3.3 to 5.3.3.5 with the DUT preamp off.

5.3.3.7 Reduce the Generator Output to –110 dBm. Repeat paragraphs 5.3.3.4 and 5.3.3.5 for any other applicable FM filters.

**NOTE:** The following tests are only performed if quieting is specified by the manufacturer. Repeat these tests for all applicable bandwidths and preamp settings.

*10 dB Quieting*

5.3.3.8 Reduce the Generator Output to -110 dBm and reset the step attenuators if necessary to 0 dB. Set the Audio/Distortion meter to REL LEVEL and the RF Generator FM modulation button to OFF.

5.3.3.9 Set the Generator RF button to OFF. Adjust the INPUT RANGE and the RELATIVE ADJUST controls for a reading of -3 dB on the audio meter. Rotate the INPUT RANGE control 1 detent CCW. Gradually increase the output of the RF generator until the audio meter again indicates -3 dB (or a 10 dB decrease). The step attenuators may be used as necessary to help achieve this reading.

5.3.3.10 Set the RF button to OFF and the INPUT RANGE control 1 detent CW. The meter should return to -3 dB. Return the RF switch to ON and rotate the INPUT RANGE control 1 detent CCW. The meter should again indicate -3 dB. Record the input level to the DUT in 5.3.3.10 of the Data Sheet.

*20 dB Quieting*

5.3.3.11 Set the RF button to OFF. Adjust the INPUT RANGE and the RELATIVE ADJUST controls for a reading of -3 dB on the audio meter. Rotate the INPUT RANGE control 2 detents CCW. Gradually increase the output of the RF generator until the audio meter again indicates -3 dB (or a 20 dB decrease). The step attenuators may be used as necessary to help achieve this reading.

5.3.3.12 Set the RF button to OFF and the INPUT RANGE control 2 detent positions CW. The meter should return to -3 dB. Return the RF button to ON and rotate the INPUT RANGE control 2 detent positions CCW. The meter should again indicate -3 dB. Record the input level to the DUT in 5.3.3.12 of the Data Sheet.

5.3.3.13 Repeat paragraphs 5.3.3.3 to 5.3.3.7 for the following frequencies, as applicable to the DUT:

## 5.4 RECEIVE FREQUENCY RANGE TEST

5.4.1 The purpose of the Receive Frequency Range Test determine both the practical receive frequency range as well as the absolute maximum frequency range limits.

5.4.2 Test hook-up (See Fig. 5-4)

5.4.2.1 With all power switches in the OFF position, the transmitter function disabled to the fullest extent possible and the Generator RF switch OFF, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen OUTPUT to 10-dB Step Attn INPUT | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50-Ohm Coax |
| DUT Audio Output to Dist/Audio Meter Input | As Required to BNC | 50-Ohm Coax |

| Connection | Connectors | Cable Type |
|---|---|---|
| 8-Ohm Load/HI-Z Amp to Dist/Audio Meter Input | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

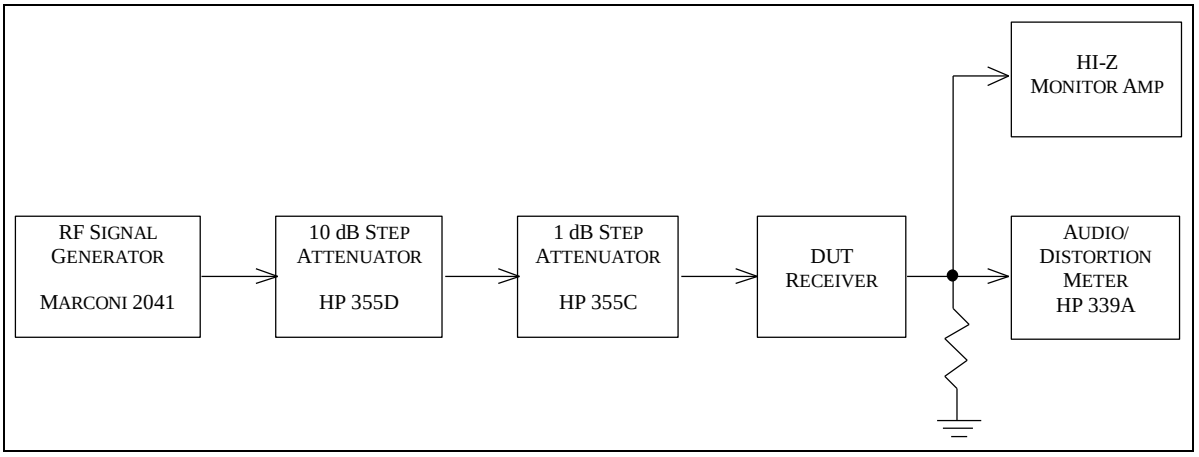



Fig. 5-4 — Receive Frequency Range Test Hook-up

5.4.3 Test Procedure

5.4.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | Lowest Available |
| | Frequency | Lowest Available |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | 500 Hz or Closest Available |
| | AGC | OFF |
| | Preamplifier | OFF |
| 10 dB Step Attn | Attenuator | 10 dB |
| RF Generator | CARRIER FREQ | Same as DUT |
| | RF LEVEL | –110 dBm |
| | CARR ON-OFF | OFF |
| | AM | Off |
| | FM | Off |
| | NOISE MODE | LOW NOISE |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All Off (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.4.3.2 Receiver hiss should be heard. Adjust the volume of the DUT and HI-Z monitor amp to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.4.3.3.

5.4.3.3 Tune the receiver for the minimum frequency possible. Verify useful sensitivity by performing an MDS test and record results in 5.4.3.2 of the Data Sheet. If useful sensitivity is not achieved, increase frequency and repeat until useful sensitivity is obtained. Perform an MDS test the DUT's specified limit if other than the lowest possible frequency.

NOTE: Useful sensitivity is somewhat ambiguous. In general terms, a receiver with less than about –100 dBm MDS sensitivity is probably marginal. Anything less than that is usually not practical for most applications.

5.4.3.4 Tune the receiver to the minimum frequency possible. Note and record any frequencies not being received by or not included in the tuning range of the DUT.

5.4.3.5 Repeat step 5.4.3.3 for the highest possible frequency. Reduce the as required for useful sensitivity and perform an MDS test at the maximum specified frequency.

**NOTE: For amateur equipment, always check for reception in the cellular bands at 824-849 and 869-894 MHz.**

## 5.5 FIRST IF AND IMAGE REJECTION TEST

5.5.1 The purpose of the first IF and Image Rejection Test is to determine the level of signal input to the receiver at the first IF and Image frequencies that will produce an audio output equal to the MDS level. The test is conducted with the receiver in the CW mode using the 500 Hz, or closest available, IF filters. Any audio filtering is disabled and the AGC is set to the OFF position if possible. The test is performed with the DUT tuned to 14.020 MHz.

5.5.2 Test hook-up (See Fig. 5-5)

5.5.2.1 With all power switches in the OFF position, the transmitter function disabled to the fullest extent possible and the Generator RF switch OFF, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen OUTPUT to 10-dB Step Attn INPUT | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50-Ohm Coax |
| DUT AUDIO OUTPUT to Dist/Audio Meter IN | As Required to BNC | 50-Ohm Coax |
| 8-Ohm Load/HI-Z Amp Across Dist/Audio Meter Input | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

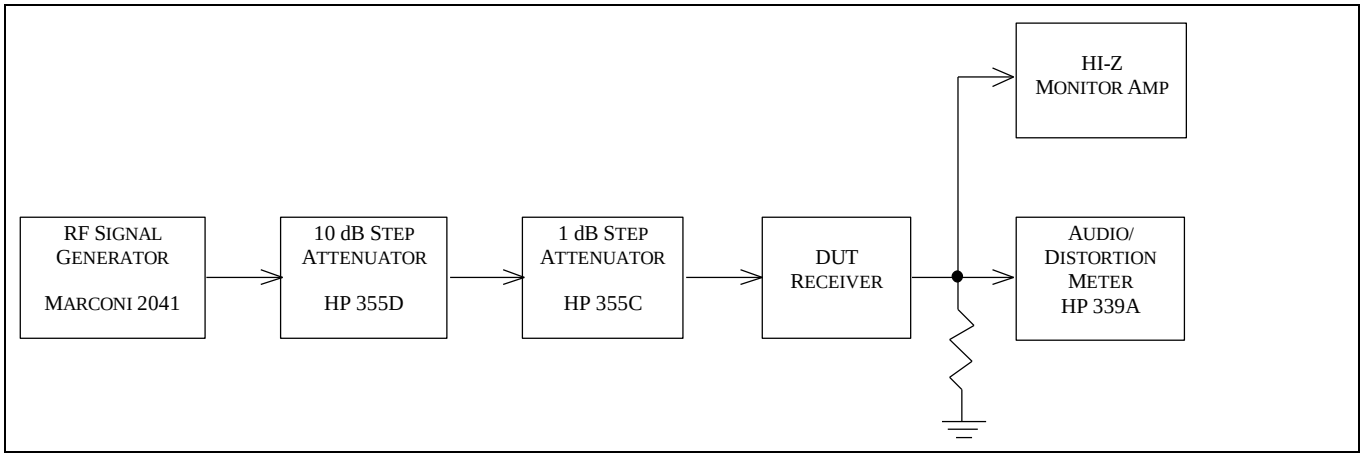



Fig. 5-5 — IFAnd Image Test Hook-up

5.5.3 Test Procedure

5.5.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | 20 Meters |
| | Frequency | 14.020 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | 500 Hz or Closest Available |
| | AGC | OFF |
| | Preamplifier | OFF |
| 10 dB Step Attn | Attenuator | 10 dB |
| RF Generator | CARRIER FREQ | DUT 1st IF |
| | RF LEVEL | –80 dBm |
| | CARR ON-OFF | ON |
| | AM | Off |
| | FM | Off |
| | NOISE MODE | LOW NOISE |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All Off (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.5.3.2 Receiver hiss should be heard. Adjust the volume of the DUT and HI-Z monitor amp to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.5.3.3.

5.5.3.3 Verify the RF Generator Output frequency has remained at the DUT 1st IF. Reset if necessary.

5.5.3.4 Tune the receiver for 14.020 MHz. Rotate the Generator OUTPUT LEVEL control as required until the signal is just heard in the receiver. Adjust the INPUT RANGE and RELATIVE ADJUST controls as required to maintain approximately a midscale meter indication while carefully tuning the Generator CARRIER FREQ control for peak signal response.

5.5.3.5 Set the RF Generator RF switch to OFF. Decrease the Audio Meter INPUT RANGE until the meter indication is near mid-scale and the two lights above this control are out. Adjust the audio meter RELATIVE ADJUST control until the audio meter reads –6 dB on the upper scale. (Adjust the INPUT RANGE control one step in either direction if necessary.)

5.5.3.6 Set the RF Generator RF switch to ON. Rotate the generator OUTPUT LEVEL control to produce an audio meter reading of –3 dB. Ensure that the Generator CARRIER FREQ control is tuned for peak response.

5.5.3.7 Set the RF Generator RF to OFF. The audio meter should return to –6 dB. Turn the RF back on and the meter should again indicate -3 dB.

5.5.3.8 Determine the IF level of the receiver by computing the sum of the RF generator output in dBm and the 10 dB step attenuator. (NOTE: Be sure to include any additional attenuation you may have included in the line between the

generator and the DUT.) Find the difference in dB from the 20 Meter MDS (with preamp off) previously measured in the test battery. Record on the Data Sheet.

Example: a) The RF generator is set for –36.1 dBm output.
b) The step attenuators are set for –10 dB.
c) The receiver IF rejection level, therefore, is:
–36.1 - 10 = –46.1 dBm
d) If we measured the noise floor to be –138.6 dBm
in the Noise Floor Test, then the IF rejection is:
–46.1 dBm - (–138.6 dBm) = 92.5 dB

5.5.3.9 Repeat steps 5.5.3.4 to 5.5.3.8 with DUT preamplifier set to ON.

5.5.3.10 Calculate the first IF image frequency as follows:

14.020 MHz +/- (2 X IF in MHz) = Image Frequency in MHz

Note there are two potential image frequencies if the 1st IF is less than 7.010 MHz Check *BOTH* frequencies for a response and record the appropriate one in 5.5.10 of the Data Sheet.

5.5.3.11 Tune the receiver for 14.020 MHz. Rotate the Generator OUTPUT LEVEL control as required until the signal is just heard in the receiver. Tune the Signal Generator to the other potential image frequency calculated in step 5.5.3.10. Compare the DUT response between the two input frequencies and set the Generator to the frequency that result in the greatest output response. Reduce the generator to provide an approximately the MDS level at the image frequency.

5.5.3.12 Adjust the INPUT RANGE and RELATIVE ADJUST controls as required maintaining approximately a mid-scale meter indication while carefully tuning the Generator CARRIER FREQ control for peak signal response. Increase the generator output level if necessary.

5.5.3.13 Set the RF Generator RF switch to OFF. Decrease the Audio Meter INPUT RANGE until the meter indication is near mid-scale and the two lights above this control are out. Adjust the audio meter RELATIVE ADJUST control until the audio meter reads –6 dB on the upper scale. (Adjust the INPUT RANGE control one step in either direction if necessary.)

5.5.3.14 Set the RF Generator RF switch to ON. Rotate the generator OUTPUT LEVEL control to produce an audio meter reading of –3 dB. Ensure that the Generator CARRIER FREQ control is tuned for peak response.

5.5.3.15 Set the RF Generator RF to OFF. The audio meter should return to -6 dB. Turn the RF back on and the meter should again indicate –3 dB. Record the Signal Generator output in 5.5.3.7 of the data Sheet and calculate the Image Rejection as follows:

Example: a) The RF generator is set for –59.6 dBm output.
b) The step attenuators are set for –10 dB.
c) The receiver image rejection level, therefore, is: –59.6 - 10 = –69.6 dBm
d) If we measured the noise floor to be -138.6 dBm in the previous
Noise Floor Test, the image rejection is: –69.6 dBm - (–138.6 dBm) = 69.0 dB

5.5.3.16 Reset the generator output level to –80 dBm. Repeat steps 5.5.3.13 to 5.5.3.15 with the DUT preamplifier set to ON.

## 5.6 ANTENNA PORT ISOLATION TEST
**(For Receivers with multiple antenna ports only)**

5.6.1 The purpose of this test is to determine the level of isolation between antenna ports with receivers having multiple antenna ports. A comparison in MDS sensitivity is made between an unused and used antenna port. The test is conducted with the receiver in the CW mode using the 500 Hz, or closest available, IF filters. Any audio filtering is disabled and the AGC is set to the OFF position if possible. The test is performed with the DUT tuned to 14.020 MHz.

5.6.2 Test hook-up (See Fig. 5-6)

5.6.2.1 With all power switches in the OFF position, the transmitter function disabled to the fullest extent possible and the Generator RF switch OFF, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen OUTPUT to 10-dB Step Attn INPUT | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn OUTPUT to DUT RF INPUT #1 | BNC to As Required | 50-Ohm Coax |
| DUT AUDIO OUT to Dist/Audio Meter IN | As Required to BNC | 50-Ohm Coax |
| 8-Ohm Load/HI-Z Amp Across Dist/Audio Meter Input | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

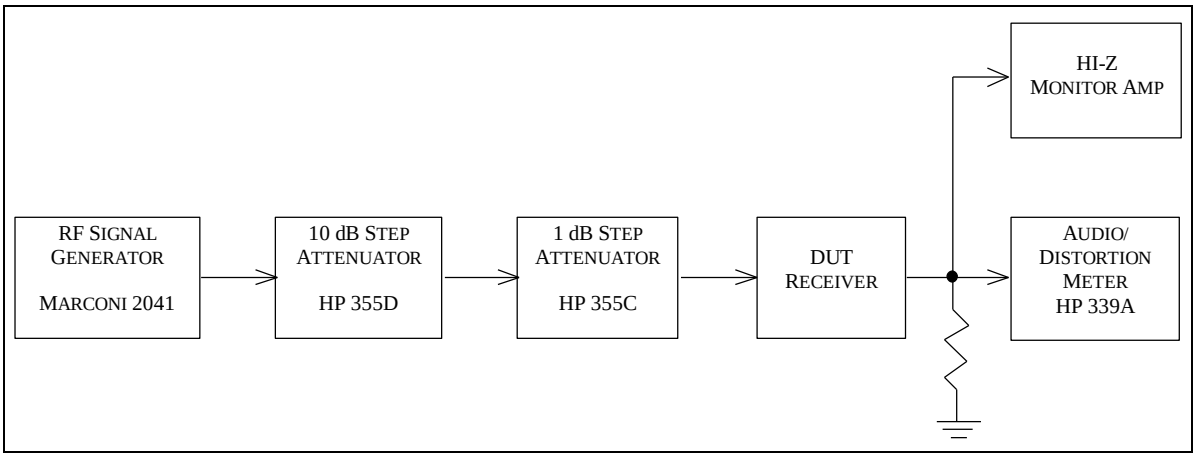



Fig. 5-6 Antenna Port Isolation Test Hook-Up

5.6.3 Test Procedure

5.6.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | 20 Meters |
| | Frequency | 14.020 MHz |
| | IF Filters | 500 Hz or Closest Available |
| | AGC | OFF |
| | Preamplifier | OFF |
| 10 dB Step Attn | Attenuator | 10 dB |

| | | |
|---|---|---|
| RF Generator | CARRIER FREQ | 14.020 MHz |
| | RF LEVEL | –80 dBm |
| | NOISE MODE | LOW NOISE |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All Off (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.6.3.2 Receiver hiss should be heard. Adjust the volume of the DUT and HI-Z monitor amp to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.6.3.3.

5.6.3.4 Tune the receiver for 14.020 MHz. Rotate the Generator OUTPUT LEVEL control as required until the signal is just heard in the receiver. Adjust the INPUT RANGE and RELATIVE ADJUST controls as required to maintain approximately a mid-scale meter indication while carefully tuning the Generator CARRIER FREQ control for peak signal response.

5.6.3.5 Set the RF Generator RF switch to OFF. Decrease the Audio Meter INPUT RANGE until the meter indication is near mid-scale and the two lights above this control are out. Adjust the audio meter RELATIVE ADJUST control until the audio meter reads –6 dB on the upper scale. (Adjust the INPUT RANGE control one step in either direction if necessary.)

5.6.3.6 Set the RF Generator RF switch to ON. Rotate the generator OUTPUT LEVEL control to produce an audio meter reading of –3 dB. Ensure that the Generator CARRIER FREQ control is tuned for peak response. Results should be similar to the previously measured 20 meter MDS measured in Test 5.1. Now remove the RF Input cable from RF Input #1 and connect it to RF Input #2. Increase the Generator RF LEV Control until the meter once again shows a 3 dB increase.

5.6.3.7 Set the RF Generator RF to OFF. The audio meter should return to –6 dB. Turn the RF back on and the meter should again indicate –3 dB.

5.6.3.8 Determine the Antenna Port Isolation by computing the difference between the two MDS measurements (in dBm). Record on the Data Sheet if deemed unsatisfactory by the test engineer.

5.6.3.9 Repeat steps 5.6.3.5 and 5.6.3.6 for preamp on and any other antenna port combinations deemed appropriate by the test engineer.

## 5.7 BLOCKING GAIN COMPRESSION

5.7.1 The purpose of the Blocking Gain Compression Test is to determine the level of gain compression, or desensitization that occurs as a result of another signal on a nearby frequency. The blocking gain compression is the difference between the level of the noise floor and the level of undesired signal that produces a 1 dB decrease of audio in a weak desired signal. Frequencies of 3.520 MHz, 14.020 MHz, 50.020 MHz, 144.020 MHz and 432.020 MHz are used for this test as appropriate for the DUT. The calculation is as follows:

Blocking Gain Compression = Blocking Level - Noise Floor (MDS level, expressed in dBm)

5.7.2 Test hook-up (See Fig. 5-7)

5.7.2.1 With all power switches in the OFF position, the transmitter function disabled to the fullest extent possible and the Generator RF switch OFF, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen 1 output to Hybrid Combiner input 1 | BNC to BNC | 50-Ohm Coax |
| Signal Gen 2 output to Hybrid Combiner input 2 | BNC to BNC | 50-Ohm Coax |
| Hybrid Combiner output to 10-dB Step Atten. input | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn output to 1-dB Step Attn input | BNC to BNC | 50-Ohm Coax |
| 1-dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50-Ohm Coax |
| DUT Audio OUTPUT to Dist/Audio Meter input | As Required to BNC | 50-Ohm Coax |
| DUT Audio Output to Signal Analyzer | BNC to BNC | 50-Ohm Coax |
| 8 Ohm Load/HI-Z Amp Across Dist/Audio Meter INPUT | As Required | As Required |

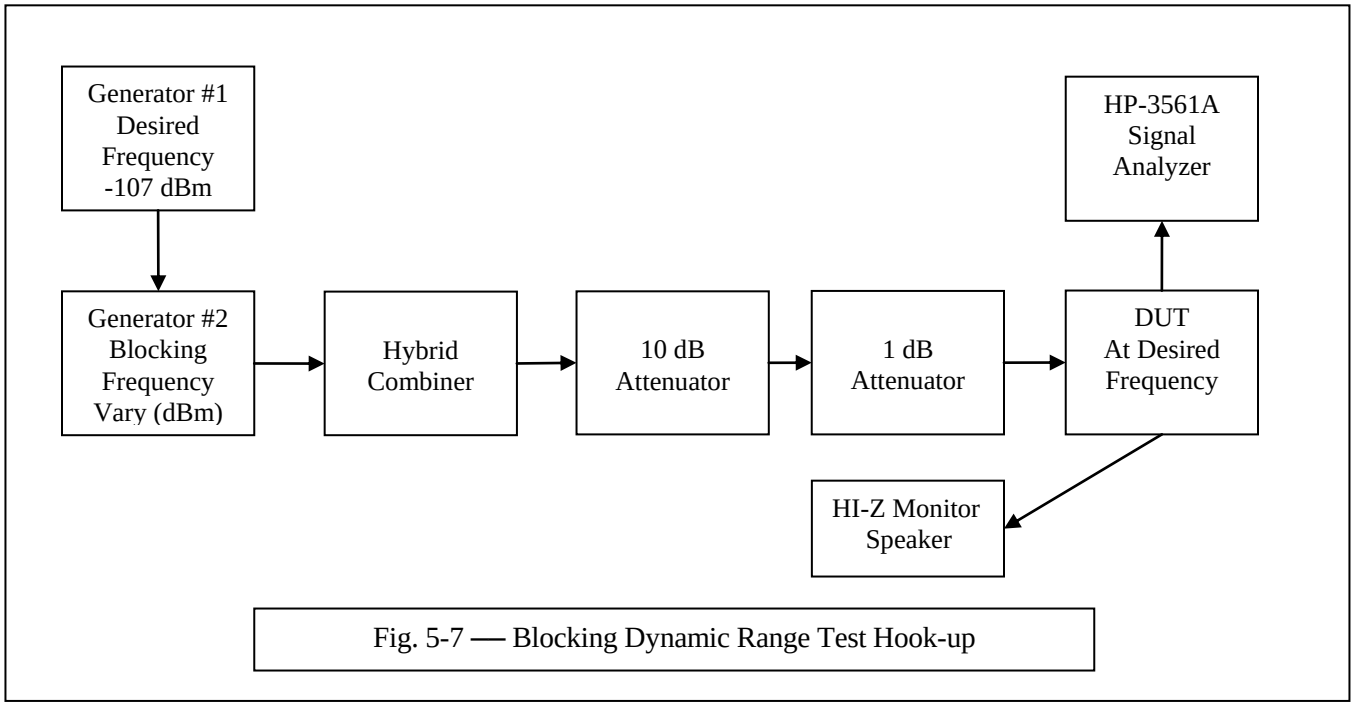



Fig. 5-7 — Blocking Dynamic Range Test Hook-up

5.7.3 Test Procedure

5.7.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Position | Control |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | 3.520 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | Closest Available to 500 Hz |
| | Any Audio Filtering | Disabled |
| | **AGC** | **OFF** |
| | VOLUME | Lowest Possible Setting |
| 10-dB Step Attn | Attenuator | 0 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| RF Generator #1 | CARRIER FREQ | 3.520 MHz |
| | RF LEVEL | –107 dBm |
| | CARR ON-OFF | ON |
| | NOISE MODE (UTIL) | LOW NOISE |
| RF Generator #2 | CARRIER FREQ | 3.500 MHz |
| | RF LEVEL | –100 dBm |
| | CARR ON-OFF | ON |
| | NOISE MODE (UTIL) | LOW NOISE |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All OFF (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |
| Signal Analyzer | CENTER FREQUENCY | 600 Hz |
| | RANGE | 10 dBv |
| | VERTICAL SCALE | -10 dBv |
| | SPAN | 100 Hz |
| | PEAK TRACK | ON |

5.7.3.2 Receiver hiss should be heard. Turn the volume level all the way down. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.7.3.3.

5.7.3.3 Verify generator #1 and generator #2 are still set for 3.520 MHz and 3.500 MHz, respectively. Set Generator #2 RF switch to OFF.

5.7.3.4 Tune the receiver for 3.520 MHz. Ensure Generator #1 RF Level is set to -107 dBm. Adjust receiver volume control, turning slowly until the desired signal is barely heard in the monitor speaker (high speaker levels will damage the Signal Analyzer!). Adjust the INPUT RANGE and REL ADJUST as required to maintain approximately a mid-scale meter indication while carefully tuning the receiver for peak signal response on the meter.

5.7.3.5 Check the Signal Analyzer for a pip and adjust the DUT volume level for a relative reading of -50 dBv on the scale. Make a note of the exact reading as displayed; ex: Y= -50.23 dB. Re-adjust the INPUT RANGE for a mid scale reading on the distortion meter. From this point on, the mid-scale reading on the distortion meter will be used for quick reference for adjustment of the volume control only. **NOTE:** Do not exceed 0 dBm into the HP-3561A Signal Analyzer or damage may result to its input circuits. Start each adjustment with the DUT volume down, turning it up slowly.

5.7.3.6 Determine the 1 dB reduction point (blocking level) by Switching Generator #2 to ON and increase the RF LEVEL to a point where a 1 dB decrease in audio is observed on the Signal Analyzer. Ex: Y= -50.23 dB with GEN #2 OFF and Y= -51.23 dB with GEN #2 ON and at the blocking level. An increase of noise level is likely, especially at 5 and 2 kHz spacing. Reduce the SPAN on the signal analyzer to 10, 5 or as low as 2 Hz, as needed, to separate the desired signal from the back round noise. (It may take several minutes for the Signal Analyzer to read the relative audio level, with the SPAN set to a narrow bandwidth, each time the blocking signal level is changed)

Record the value of the blocking level (Generator #2 level + 3 dB combiner *loss* = blocking level in dBm) in 5.7.3.5 of the Data Sheet. Be sure to account for the hybrid combiner *and* any step attenuator losses.

EX: (+5.6 dBm Blocking Signal) + (-3 dBm combiner loss)) = +2.6 dBm = Blocking Level

**NOTE:** Certain SDR receivers use direct conversion and an analog to digital converter (ADC). The blocking level may be a few decibels past the point where ADC clipping occurs and may be well above 0 dBm. Do not exceed +10 dBm into DUT.

5.7.3.7 Repeat the above steps for Generator #2 set to 3.540 MHz, then at 5 and 2 kHz above and below the desired frequency, then repeat steps with Generator #1 set to: 14.020, 50.020 and 432.020 MHz. Record on data sheet.

5.7.3.8 Repeat paragraphs 5.7.3.2 to 5.7.3.7 for 20 kHz spacing above and below the desired frequency with the DUT preamp ON. If the DUT has more than one preamp setting, test all settings and record on the data sheet.

## 5.7A RECIPROCAL MIXING TEST

5.7A.1 The purpose of the Phase Noise Test is to determine the level of noise generated by the reciprocal mixing of the DUT due to nearby strong signals. This test, along with Blocking Gain Compression, shows clearly how well a receiver performs with strong RF signals on nearby frequencies. A 14.025 MHz crystal controlled oscillator at +15 dBm, with low phase distortion and noise, is connected to the DUT RF input. The receiver is tuned to a series of nearby frequencies, 2, 5 & 20 kHz above and 2, 5 & 20 kHz below the oscillator's frequency. The oscillator attenuation is then reduced to a point where the level of noise is 3 dB above the noise floor of the DUT. Reciprocal Mixing is calculated for each point by use of the following equation:

Reciprocal Mixing (in dBc) = (MDS) – (Crystal Oscillator Level at the DUT input)

5.7A.2 Test hook-up, See Figure 5.7A

5.7A.2.1 With all power switches in the OFF position and the transmitter function disabled to the fullest extent possible, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Low Noise Osc OUTPUT to 10-dB Step Attn INPUT | BNC to BNC | 50 Ohm Coax |
| 10-dB Step Attn OUTPUT to 1-dB Step Attn INPUT | BNC to BNC | 50 Ohm Coax |
| 1 dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50 Ohm Coax |
| DUT AUDIO OUTPUT to Dist/Audio Meter INPUT | As Required to BNC | 50 Ohm Coax |
| 8 Ohm Load/HI-Z Amp Across Dist/Audio Meter INPUT | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |
| Power Source to Crystal Oscillator | As Required | As Required |

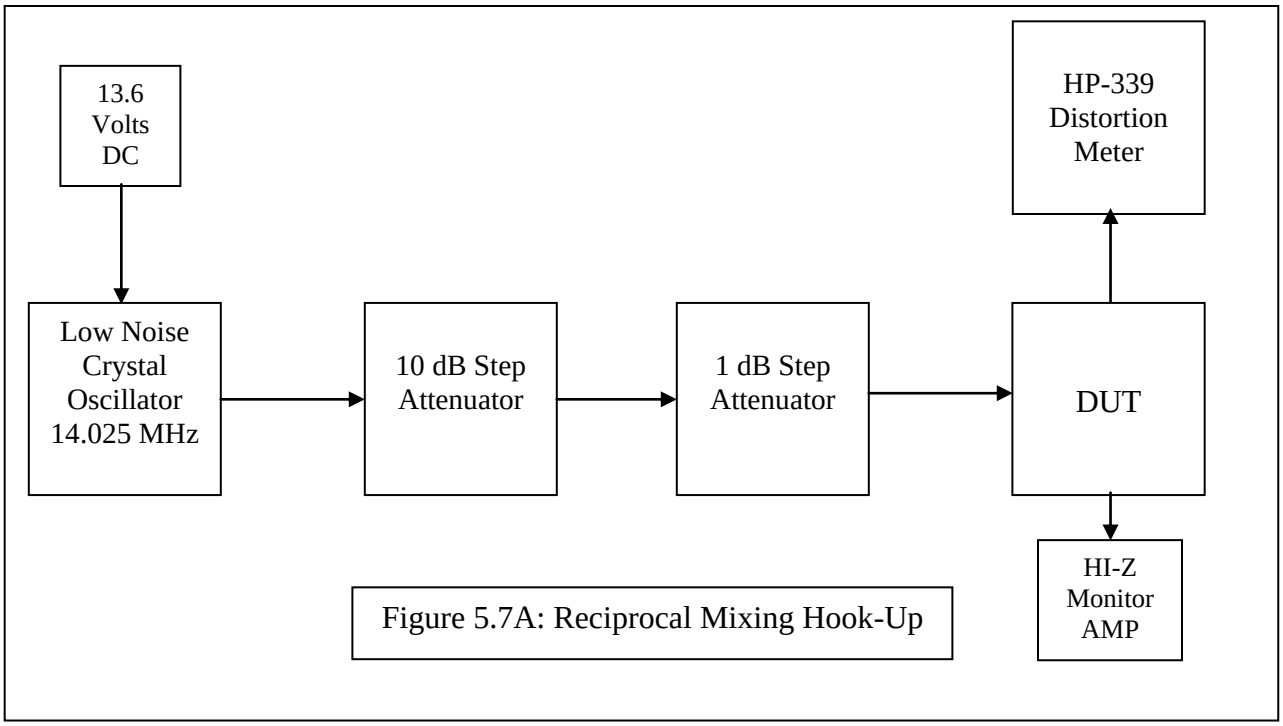



Figure 5.7A: Reciprocal Mixing Hook-Up

5.7A.3 Test Procedure

5.7A.3.1 Turn the DUT and Audio Distortion Meter power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | 20 Meters |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | Same as MDS Test (5.1) |
| 10-dB Step Attn | Attenuator | –100 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| Low Noise Oscillator | ON/OFF Switch | OFF |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All OFF (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.7A.3.2 Receiver hiss should be heard. Adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 6.5.3.3.

5.7A.3.3 Tune the receiver for 14.025 MHz. Decrease the 10 dB Step Attenuator, one step at a time, until the signal is just heard in the receiver. Adjust the INPUT RANGE control as required to maintain approximately a midscale meter indication while carefully tuning the receiver for peak signal response. Tune the DUT receiver down 20 kHz from the point of peak response.

5.7A.3.4 Turn the Low Noise Oscillator switch to OFF. Set the meter REL ADJUST and INPUT RANGE controls for –6 dB. Turn the Low Noise Oscillator Switch to ON. Carefully decrease the Step Attenuators for a meter indication of –3 dB. Calculate the input to the DUT, in dBm, by subtracting the step attenuator settings from the +15 dBm output of the oscillator. Record on the Data Sheet.

5.7A.3.5 Continue to tune the DUT up to the following frequencies and repeat step 6.5.3.4. (NOTE: It is unnecessary to continue this procedure beyond the point that 3 dB of noise above MDS can no longer be achieved.)

14.020 MHz---Peak Response - 5 kHz (Completed in step 6.5.3.4)
14.023 MHz---Peak Response - 2 kHz
14.027 MHz---Peak Response + 2 kHz
14.030 MHz---Peak Response + 5 kHz
14.045 MHz---Peak Response + 20 kHz

5.7A.3.6 Calculate Reciprocal Mixing for each test frequency by use of the following equation:

Reciprocal Mixing (in dBc) = (MDS) – (Crystal Oscillator Level at the DUT input)

Record all results on the Data Sheet.

5.7A.3.7 The above steps can be repeated, substituting the IFR-2041 Signal Generator for the crystal oscillator on 3.502 MHz, provided that the Generator is set for low noise mode.

**NOTE:** Direct Conversion Receivers contain Analog to Digital Converters which may have little or no reciprocal mixing up to the point of ADC Clipping. Reciprocal Mixing should be reported as:
Reciprocal Mixing = > (MDS) – (threshold level of ADC Clipping)

EX: MDS = -126 dBm
ADC Clipping = +0.2 dBm
RM = -126 – (+0.2) = 126.2 dBc

We report “better than” –126 dBc

## 5.8 TWO-TONE 2ND AND 3RD ORDER DYNAMIC RANGE TEST

5.8.1 The purpose of the Two-Tone Dynamic Range Test is to determine the range of signals that can be tolerated by the DUT while producing essentially no undesired spurious responses. To perform the 3rd Order test, two signals of equal amplitude and spaced 20 kHz apart, are injected into the input of the receiver. If we call these frequencies $f_1$ and $f_2$, the third-order products will appear at frequencies of $(2f_1-f_2)$ and $(2f_2-f_1)$. Similarly, the 2nd order test also requires two input signals of equal amplitude. The product, however, appears at a frequency of $(f_1+f_2)$. The intercept points are calculated for each test point. This test is performed on 3.5 MHz, 14 MHz, 50 MHz, 144 MHz and 432 MHz as appropriate for the DUT. Additional tests are performed at various frequencies and spacing for the expanded set of tests (see data sheet).

5.8.2 Test hook-up (See Fig. 5-8)

**NOTE**: If proceeding from the previous Blocking Dynamic Range Test, no hook-up changes are required.

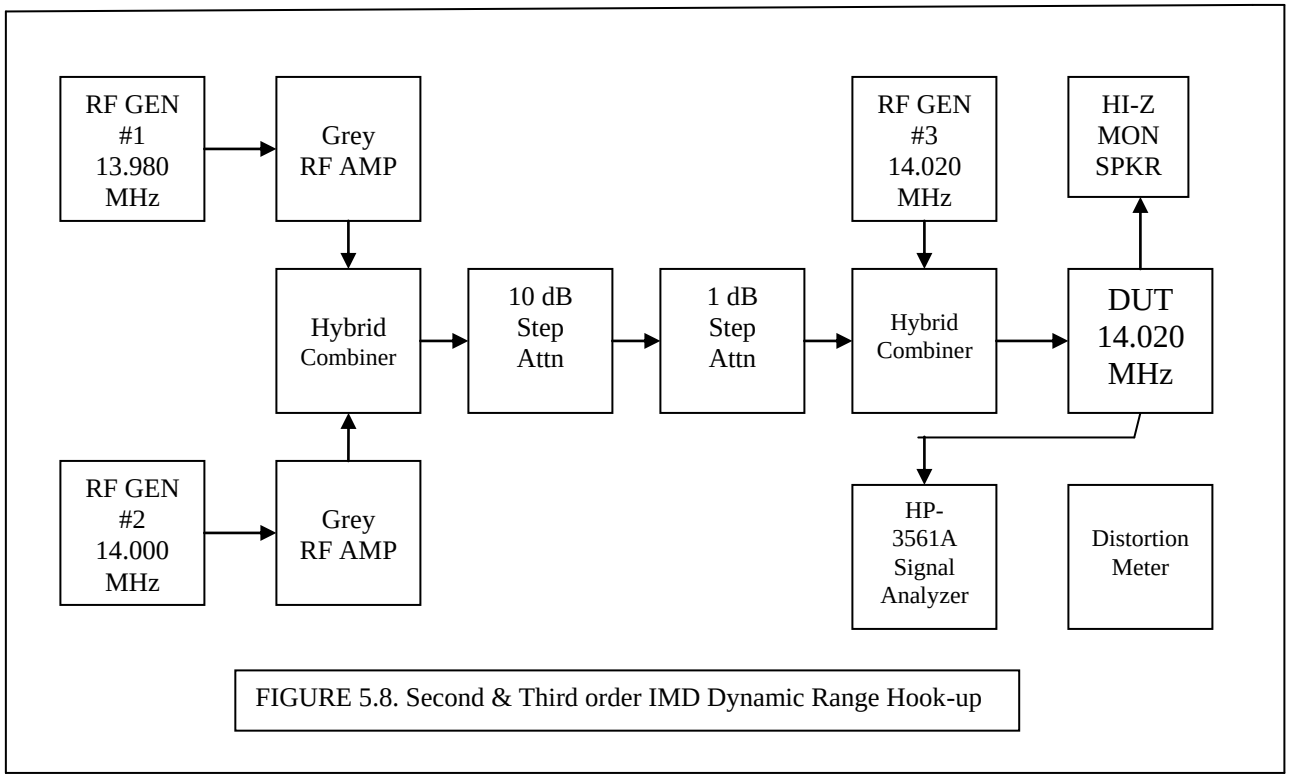



FIGURE 5.8. Second & Third order IMD Dynamic Range Hook-up

RF Generator 1: Marconi 2041
RF Generator 2: IFR 2041
RF Generator 3: Rohde & Schwarz CMT 54
"Grey Amp": Synergy Microwave low noise, wideband amplifier

5.8.2.1 With all power switches in the OFF position and the transmitter function disabled to the fullest extent possible, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen 1 output to input of Grey Amp #1 | BNC to BNC | 50-Ohm Coax |
| Signal Gen 2 output to input of Grey Amp #2 | BNC to BNC | 50-Ohm Coax |
| Grey Amp #1 output to Hybrid Combiner input | BNC to BNC | 50-Ohm Coax |
| Grey Amp #2 output to Hybrid Combiner input | BNC to BNC | 50-Ohm Coax |

| | | |
|---|---|---|
| Hybrid Combiner output to 10-dB Step Attn input | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn output to 1-dB Step Attn input | BNC to BNC | 50-Ohm Coax |
| 1-dB Step Attn output to Hybrid Combiner input | BNC to BNC | 50-Ohm Coax |
| Signal Generator #3 to Hybrid Combiner input | BNC to BNC | 50-Ohm Coax |
| Hybrid Combiner output to DUT RF input | BNC to as required | 50-Ohm-Coax |
| DUT audio output to Dist/Audio Meter input | As Required to BNC | 50-Ohm Coax |
| DUT to HP-3156A Signal Analyzer | BNC to BNC | 50-Ohm Coax |
| 8 Ohm Load Across Dist/Audio Meter input | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

5.8.3 Test Procedure

5.8.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | 20 Meters |
| | Frequency | 14.020 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | Closest Available to 500 Hz |
| | DUT Audio Filter(s) | Disabled |
| | AGC | **ON** |
| 10-dB Step Attn | Attenuator | 110 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| RF Generator #1 | CARRIER FREQ | 13.980MHz |
| | RF LEVEL | –18 dBm |
| | CARR ON-OFF | OFF |
| | NOISE MODE (UTIL) | LOW NOISE |
| RF Generator #2 | CARRIER FREQ | 14.000 MHz |
| | RF LEVEL | –18 dBm |
| | CARR ON-OFF | OFF |
| | NOISE MODE (UTIL) | LOW NOISE |
| RF Generator #3 | CARRIER FREQ | 14.020 MHz |
| | RF LEVEL | -97 dBm |
| | CARR ON-OFF | OFF |

| | | |
|---|---|---|
| Signal Analyzer | CENTER FREQUENCY | 600 Hz |
| | RANGE | 10 dBv |
| | VERTICAL SCALE | -10 dBv |
| | SPAN | 100 Hz |
| | PEAK TRACK | ON |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All OFF (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

HI-Z Monitor Amp Speaker

5.8.3.2 Receiver hiss should be heard. Adjust the volume to the lowest possible setting. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.8.3.3.

5.8.3.3 Verify RF generators #1, #2 and #3 are still set for 13.980, 14.000 and 14.020 MHz, respectively. ALL CARR ON-OFF controls *must* be in the OFF position before proceeding to the next step.

**IMPORTANT NOTE**: Before proceeding, it is imperative to insure that both Grey RF amps have the same output of 0 dBm at the output of the Hybrid Combiner. Check the outputs buy hooking up a BCN to BNC cable from the combiner output to the HP-8563E spectrum analyzer, making sure the two signals are equal and at 0 dBm. Adjust the output control on each Grey amp if needed.

5.8.3.4 Set Generator #3 to CARR OFF-ON to ON. Turn up the DUT volume so it is just heard in the HI-Z monitor amp. Tune the receiver for peak response at 14.020 MHz. Decrease the audio meter INPUT RANGE control until the meter indication is near mid scale. (The two lights just above the RANGE Control should go out.) Check to make sure the Generator level is adjusted to -94 dBm, which is a -97 dBm (S5) level at the combiner output.

5.8.3.5 Look for a PIP on the Signal Analyzer. This is the **desired signal**. Center pip if needed. Adjust the DUT volume so it is close to -50 dB on the scope. Set the audio meter RELATIVE ADJUST control until meter reads –6 dB (upper scale). Adjust the INPUT RANGE switch as required to achieve this indication. From this point on, The audio meter is used as a handy reference for adjusting the DUT volume. Do NOT exceed 0 dBm at the input to the Signal Analyzer, or damage to its input circuits will result. Always start each measurement with the volume down.

5.8.3.6 Turn both the generators #1 & #2 CARR ON-OFF switches to ON. Slowly turn up the 10 dB step attenuator until a second PIP appears on the Signal Analyzer. This is the **IMD signal**. With peak tracking on, make both pips on the screen equal by adjusting the step attenuators. When both are equal, the IMD level is what the attenuators are set to, *plus* 3 dB of the combiner loss. Example: Attenuators are set to 26 dB. The IMD level would be -26 dB + 3 dB. Record the IMD level on the data sheet.

5.8.3.7 Repeat steps 5.8.3.4 to 5.8.3.7, but with Generator #1 & 2 set for 14.010 & 14.015 MHz respectively (5 kHz spacing). Repeat steps having Generator 1 & 2 at 14.016 & 14-018 MHz respectively (2 kHz spacing). Repeat steps for 2, 5 & 20 kHz spacing above the desired frequency. Record on data sheet.

**NOTE:** Measurements at 5 and 2 kHz spacing may require reducing the bandwidth on the Signal Analyzer, especially when Generator #3 is set to the MDS level (plus 3 dB) . When doing so, it will be necessary to put both pips on the scope as close together as possible before reducing the bandwidth. This is accomplished by adjust either Generator #1 or Generator #2 by small fractions of a Hertz. Slowly adjust one of the generators while watching the IMD signal, placing the IMD pip close to the desired signal pip. The SPAN may now be adjusted to a smaller bandwidth. (25 or 10 Hz works well).

5.8.3.8 Repeat the above IMD measurements, but with Generator #3 set to the MDS plus 3 dB. This results with an MDS level at the combiner output. If the MDS is -126 dBm, set the generator to -123 dBm.

5.8.3.9 Repeat steps 5.8.3.4 and 5.8.3.7 for preamp for 20 kHz spacing above and below the desired signal. Repeat if preamp has more than one setting. Return preamp to off.

5.8.3.10 Repeat IMD tests for the required frequencies and spacing shown on the data sheet.

5.8.3.11 Calculate the Two- Tone, Third Order IMD Dynamic Range for each data point.

3rd Order DR = MDS – IMD Level.
Example:
MDS = -126 dBm
IMD level = -26 dBm

3rd Order IMD DR = ( -126 dBm) – (-26 dBm) = 100 dB.

Calculate for each data point

5.8.3.12 Calculate the 3rd Oder Intercept Point (IP3) for each data point using the following formulas:

**IP3 at MDS** : (1.5)(IMD DR in dB)+(MDS in dB)

Example: IMD DR = 100 dB, MDS = -126 dBm

IP3 (at MDS) = (1.5)(100 dB) + (-126 dBm) = 24 dB

**IP3 at S5**: $\frac{3 \text{ X } (\text{S5 IMD level}) - (\text{S5 Reference})}{2}$

Example: IMD level = -15 dBm, S5 reference = -97 dBm

IP3 at S5 = $\frac{3 \text{ X } (-15 \text{ dBm}) - (-97 \text{ dBm})}{2}$ = 26 dB

Record all IP3 data points on data sheet.

**Second Order Two Tone IMD Test:**

Gen 1 = 6.000 MHz, Gen 2 =8.020 MHz, DUT = 14.020 MHz, same procedure as above.

Second Order Intercept = 2 X (S 5 IMD level) – (S 5 level)
Example IMD level = -10 dBm

2Nd Order Intercept Point, S 5 level = 2(-10 dBm) – (-97 dBm) = +77 dBm

Record IP2 data point on data sheet.

## 5.9 FM ADJACENT-CHANNEL SELECTIVITY TEST

5.9.1 The purpose of the FM Adjacent Channel Selectivity Test is to measure the ability of the DUT receiver to reject interference from individual undesired signals while receiving various levels of desired signal. The desired carrier signal will be at 29.000 MHz, modulated at 1000 Hz, and the offending signal will be located at adjacent nearby frequencies with 400 Hz modulation. (NOTE: The SINAD Test in 5.3 must be performed before this test can be completed.)

5.9.2 Test hook-up (See Fig. 5-9)

**NOTE**: If proceeding from the previous Two-Tone 3rd-Order Dynamic Range Test, no hook-up changes are required. Proceed directly to step 5.9.3.

5.9.2.1 With all power switches in the OFF position and the transmitter disabled, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen 1 Output to Hybrid Combiner Input 1 | BNC to BNC | 50-Ohm Coax |
| Signal Gen 2 Output to Hybrid Combiner Input 2 | BNC to BNC | 50-Ohm Coax |
| Hybrid Combiner output to 10-dB Step Attn input | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn output to 1-dB Step Attn input | BNC to BNC | 50-Ohm Coax |
| 1-dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50-Ohm Coax |
| DUT Audio Output to Dist/Audio Meter input | As Required to BNC | 50-Ohm Coax |
| **Connection** | **Connectors** | **Cable Type** |
| 8 Ohm Load/HI-Z Amp Across Dist/Audio Meter IN | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

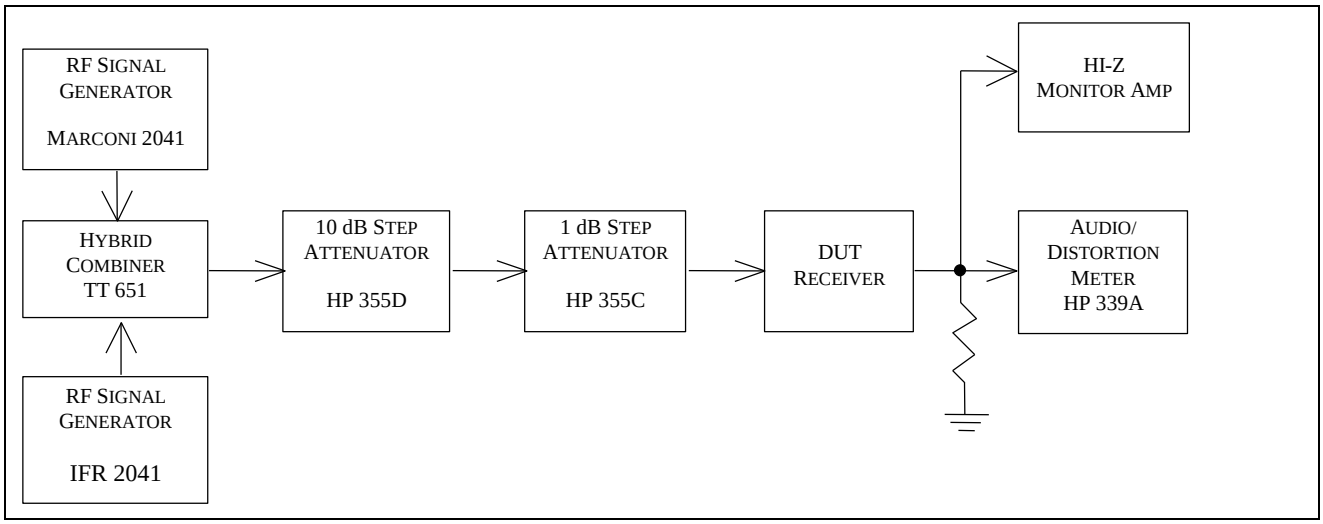



Fig. 5-9 — FM Channel Selectivity Test Hook-up

5.9.3 Test Procedure

5.9.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | FM |
| | Band Selector | 10 Meters |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | 15 kHz for FM |
| | Any Audio Filtering | Disabled |
| | Preamp | On |
| 10-dB Step Attn | Attenuator | 0 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| RF Generator #1 | CARRIER FREQ | 29.000 MHz |
| | RF LEVEL | –110 dBm |
| | NOISE MODE (UTIL) | NORMAL |
| | CARR ON-OFF | ON |
| | AM | OFF |
| | FM | ON |
| | FM DEVN | 3 kHz |
| | Modulation Frequency | 1000 Hz |
| RF Generator #2 | CARRIER FREQ | 28.980 MHz |
| | RF LEVEL | –110 dBm |
| | NOISE MODE (UTIL) | NORMAL |
| | CARR ON-OFF | OFF |
| | AM | OFF |
| | FM | ON |
| | FM DEVN | 3 kHz |
| | Modulation Frequency | 400 Hz |
| Instrument | Control | Position |
| Audio/Distortion Meter | FUNCTION | DIST |
| | DIST RANGE | 30 % |
| | FREQUENCY | 1.0 X 1K |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All OFF (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.9.3.2 Receiver hiss should be heard. Adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.9.3.3.

5.9.3.3 Tune the receiver for 29.000 MHz. Increase the #1 generator RF LEVEL, if necessary, until the signal is just heard in the receiver. Set the DUT volume control for a normal listening level, and carefully tune the receiver for minimum distortion. This control should be set for approximately 1/4 to 3/4 of the maximum.

5.9.3.4 Set the #1 signal generator level to produce a 12 dB SINAD response in the DUT. The generator output should be 16 dB greater than the level previously recorded in 5.3.3.5 of the Data Sheet during the initial SINAD test. (The 16 dB accounts for the attenuator and combiner losses.) The meter should now indicate 25% distortion. Change the DISTORTION RANGE to 100. Set the #2 signal generator RF switch to ON. Increase the output level of this generator until the distortion meter indicates 50% distortion or 6 dB SINAD. Note the generator and attenuator settings. Calculate the input signal level to the DUT and record in 5.9.3.4 of the Data Sheet.

5.9.3.5 Return generator and attenuator levels to their initial settings. Set the #2 generator to 29.020 MHz and set its RF switch to OFF. Repeat step 5.9.3.4.

5.9.3.6 Repeat paragraphs 5.9.3.4 and 5.9.3.5 for the following bands as applicable to the DUT: 52 MHz, 146 MHz and 440 MHz.

5.9.3.7 Calculate the adjacent channel rejection of the DUT for each test point by subtracting the 12 dB SINAD level in dBm from the 6 dB SINAD level in dBm measured in steps 5.9.3.4 and 5.9.3.5. Record in 5.9.3.9 of the Data Sheet.

Example: The 12 dB SINAD value measured in 5.3.3.6./5.3.3.7 = –113
The 6 dB SINAD level measured per this test = –68 dBm

Adjacent channel rejection = 6 dB SINAD - 12 dB SINAD = (–68 dBm) - (–113 dBm) = 45 dB

**NOTE:** The following step need only be performed if proceeding to the FM Two-Tone 3rd-Order Dynamic-Range Test (5.10). The results will be used to determine phase noise limiting during this test.

5.9.3.8 Return generator and attenuator levels to their initial settings and turn the #2 Generator RF LEVEL and FM switches to OFF. Repeat step 5.9.3.4 with the #2 generator still set for 29.020 MHz and with the FM switch still set to OFF. Calculate the equivalent phase noise limit using the same equations as in step 5.9.3.6. This limit figure represents the FM 2-tone, 3rd-order dynamic range of an unmodulated signal.

5.9.3.9 Repeat step 5.9.3.8 for the following bands as applicable to the DUT: 52 MHz, 146 MHz and 440 MHz.5.8.3.11 Calculate the Dynamic Range by subtracting the MDS noise floor from the IMD figure.

Example: Both RF generators are set at –14 dBm
The RF attenuators are set for –24 dB
The hybrid combiner loss is –6 dB
The receiver noise floor MDS is –135 dBm

IMD LEVEL = (–14 dBm) - 24 dB - 6 dB = –44 dBm
IMD Dynamic Range = IMD LEVEL - NOISE FLOOR = –44 dBm - (–135 dBm) = 91 dB

5.8.3.12 Calculate the Third Order Intercept Point for each applicable Dynamic Range as follows:

Third Order Intercept = (1.5)(IMD Dynamic Range in dB) + (MDS in dBm)

5.8.3.13 Calculate the Second Order Intercept Point using the applicable Dynamic Range as follows:

Second Order Intercept = (2)(2nd Order IMD Dynamic Range in dB) + (MDS @ 14.020 MHz in dBm)
Record on Data Sheet.

5.8.3.14 The preferred method for this test is to use S5 signal levels instead of MDS levels. First, use a signal generator tuned to the DUT frequency to determine the S5 reference signal level for each of the frequencies of interest. Note: The DUT's AGC may need to be on in order to get an S-meter indication.

5.8.3.15 Perform the 2-Tone, 3rd-Order as specified above, but set the attenuators to provide an S5 indication on the DUT for each of the IMD frequencies.

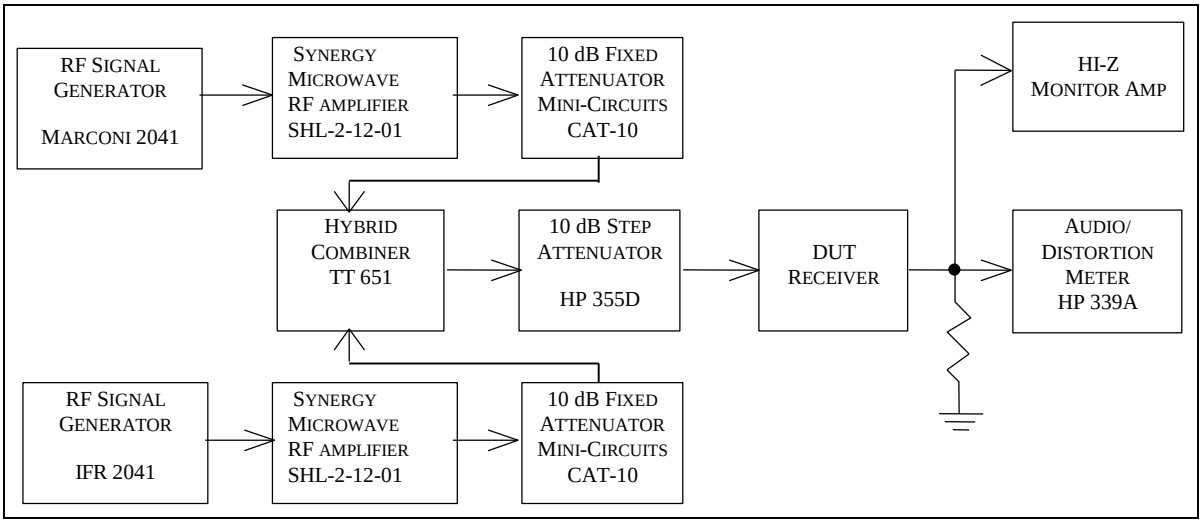
5.8.3.16 Calculate the 2nd and 3rd-Order intercept points as follows:

Third Order Intercept = (3 * (S5 IMD Level) - (S5 Reference)) / 2
Second Order Intercept = 2 * (S5 IMD Level) - (S5 Reference)

5.8.3.17 The results of this method should show close correlation with the intercept points determined by the MDS test. If not, the test engineer determines (from the MDS and dynamic range data) which method provides the more accurate result.

5.8.3.18 For the expanded set of tests, repeat this test on additional frequencies and spacing indicated on data sheet. The swept IMD dynamic range test is performed on 14 MHz only and the data is typically presented in graphical output form in the expanded test for a clear visual indication of the receiver's blocking dynamic range performance over a wide frequency span within the 14 MHz band.

5.8.3.19 Hook-up for higher-level measurements. With some high-performance transceivers, the necessary signal levels for producing a 3rd order or 2nd order IMD response exceed the spurious-free range of the standard test setup detailed in section 5.8.2. In these cases, the following alternate hook-up is used.




This hook-up should be used for generator levels above –14 dBm to prevent problems from test fixture IMD. The Synergy Microwave amplifiers specified are 24V "ultra-linear" units that provide approximately 25 dB of gain up to +29 dBm output for frequencies between 1 and 450 MHz.

## 5.10 FM TWO-TONE 3rd ORDER DYNAMIC RANGE TEST

5.10.1 The purpose of the FM Two-Tone 3rd Order Dynamic Range Test is to determine the range of signals that can be tolerated by the DUT in the FM mode while producing no spurious responses greater than the SINAD level. To perform this test, two signals, $f_1$ and $f_2$, of equal amplitude and spaced 20 kHz apart, are injected into the input of the receiver. The signal located 40 kHz from the distortion product being measured is modulated at 1,000 Hz with a deviation of 3 kHz. The DUT receiver is tuned to the Third Order IMD frequencies as determined by $(2f_1-f_2)$ and $(2f_2-f_1)$. The input signals are then raised simultaneously by equal amounts until 25% distortion, or the 12 dB SINAD point, is obtained.

5.10.2 Test hook-up (See Fig. 5-7)

**NOTE**: If proceeding from the previous FM Adjacent Channel Selectivity Test, no hook-up changes are required. Proceed directly to step 5.10.3.

5.10.2.1 With all power switches in the OFF position and the transmitter function disabled to the fullest extent possible, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen 1 output to Hybrid Combiner input 1 | BNC to BNC | 50-Ohm Coax |
| Signal Gen 2 output to Hybrid Combiner input 2 | BNC to BNC | 50-Ohm Coax |
| Hybrid Combiner output to 10-dB Step input | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn output to 1-dB Step Attn input | BNC to BNC | 50-Ohm Coax |
| 1-dB Step Attn output to DUT RF input | BNC to As Required | 50-Ohm Coax |
| DUT Audio Output to Dist/Audio Meter input | As Required to BNC | 50-Ohm Coax |
| 8 Ohm Load/HI-Z Amp Across Dist/Audio Dist/Audio Meter Input | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

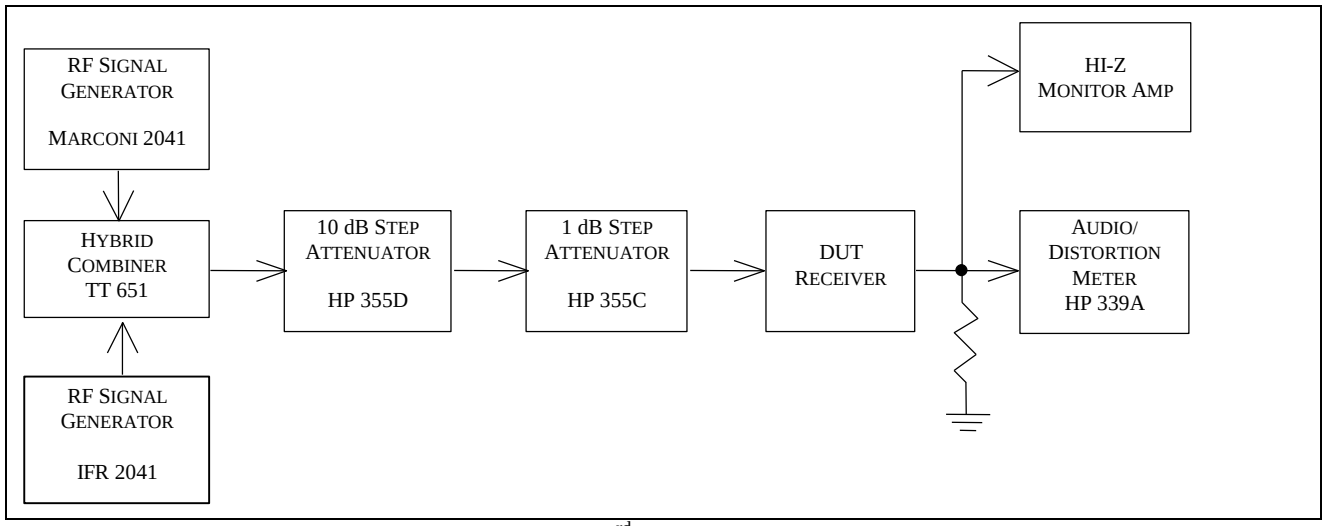



Fig. 5-10 — FM Two-Tone, 3rd Order Dynamic Range Test Hook-up

5.10.3 Test Procedure

5.10.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | FM |
| | Band Selector | 10 Meters |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | Narrowest Available |
| | DUT Audio Filter(s) | Disabled |
| | AGC | OFF |
| | Preamp | ON |
| 10-dB Step Attn | Attenuator | 120 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| RF Generator #1 | CARRIER FREQUENCY | 29.000 MHz |
| | RF LEVEL | –14 dBm |
| | CARR ON-OFF | ON |
| | NOISE MODE[1] | NORMAL[1] |
| | FM | On |
| | AM | Off |
| | FM DEVN | 3 kHz |
| | Modulation Frequency | 1000 Hz |
| RF Generator #2 | CARRIER FREQUENCY | 29.020 MHz |
| | RF LEVEL | –14 dBm |
| | CARR ON-OFF | ON |
| | NOISE MODE[2] | NORMAL[1] |
| | FM | On |
| | AM | Off |
| | FM DEVN | 3 kHz |
| | Modulation Frequency | 1000 Hz |
| Audio/Distortion Meter | FUNCTION | DIST |
| | RELATIVE ADJUST | Any |
| | FILTERS | All OFF (Out) |
| | DISTORTION RANGE | 30%/-10 dB |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |
| | FREQUENCY | 1.0 x 1 kHz |
| | OSCILLATOR | Any |

5.10.3.2 Receiver hiss should be heard. Adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.10.3.3.

5.10.3.3 Verify RF generators #1 and #2 are still set for 29.000 and 29.020 MHz, respectively. Both CARR ON-OFF controls must be in the ON position before proceeding to the next step. Set generator #1 FM switch to OFF.

5.10.3.4 Tune the receiver for the lower IMD frequency of 28.980 MHz. Decrease the attenuation of the step attenuators until the signal is just heard in the receiver. Decrease the audio meter INPUT RANGE control until the two lights just above the range switch go out. Tune the DUT for minimum distortion on the meter.

[2] NOTE: 3 kHz deviation is not possible at a frequency of 29 MHz unless the signal generator is in the Normal Noise Mode. Depress UTILITIES menu button for this setting.

5.10.3.5 Decrease the attenuator controls until the distortion meter indicates 25% distortion. Note all attenuator settings and record receiver signal input level in 5.10.3.5 of the Data Sheet.

5.10.3.6 Set #2 generator FM switch to OFF and the #1 generator FM switch to ON. Tune the DUT to the upper IMD frequency of 29.040 MHz and repeat paragraphs 5.10.3.3 to 5.10.3.5.

5.10.3.7 Calculate and record the 3rd order dynamic range in dBm for each IMD product as follows:

DYNAMIC RANGE = (IMD LEVEL) - (12 dB SINAD)

Example: Both RF generators are set at –14 dBm
The hybrid combiner loss is –3 dB
The attenuators are set for –24 dB
The 12 dB SINAD is –122.3 dBm

The IMD level = (–14 dBm) - 3 dB - 27 dB - (122.3 dBm) = 78.3 dB

Note: An FM 2-tone, 3rd-order dynamic range that is noise-limited will appear to be better than it actually is due to the additional signal necessary to overcome the increased noise. If the calculated FM 2-tone, 3rd-order dynamic range is greater than the phase noise limit determined by the adjacent channel selectivity test, then the actual dynamic range is the phase noise limit figure and the measurement is noise-limited.

5.10.3.8 Repeat paragraphs 5.10.3.3 to 5.10.3.7 for the following frequency bands as applicable to the DUT: 52 MHz, 146 MHz and 440 MHz. Also perform this test at a spacing of 10 MHz on these bands.

## 5.11 AUDIO POWER OUTPUT TEST

5.11.1 The purpose of the Audio Power Output Test is to measure the audio output voltage developed by the DUT at 1,000 Hz. The manufacturer's specified load and distortion is used for this test. If unspecified, the power is measured with an 8 Ohm load at 10% distortion. The audio power is then calculated by the equation:

$$P_{audio} = \frac{V^2}{8\Omega}$$

5.11.2 Test hook-up (See Fig. 5-8)

**NOTE**: 1) If proceeding from the previous FM Two-Tone 3rd Order Dynamic Range Test, only the hook-up changes shown with a dotted line are required.

2) Do not use headphone audio output for this test. It will not develop full power.

5.11.2.1 With all power switches in the OFF position and the transmitter function disabled to the fullest extent possible, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen output to 10-dB Step Attn input | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn output to 1-dB Step Attn input | BNC to BNC | 50-Ohm Coax |
| 1-dB Step Attn output to DUT RF input | BNC to As Required | 50-Ohm Coax |
| DUT Audio output to Dist/Audio Meter input | As Required to BNC | 50-Ohm Coax |
| Speaker Load/HI-Z Amp Across Dist/Audio Meter INPUT (See Note 1) | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

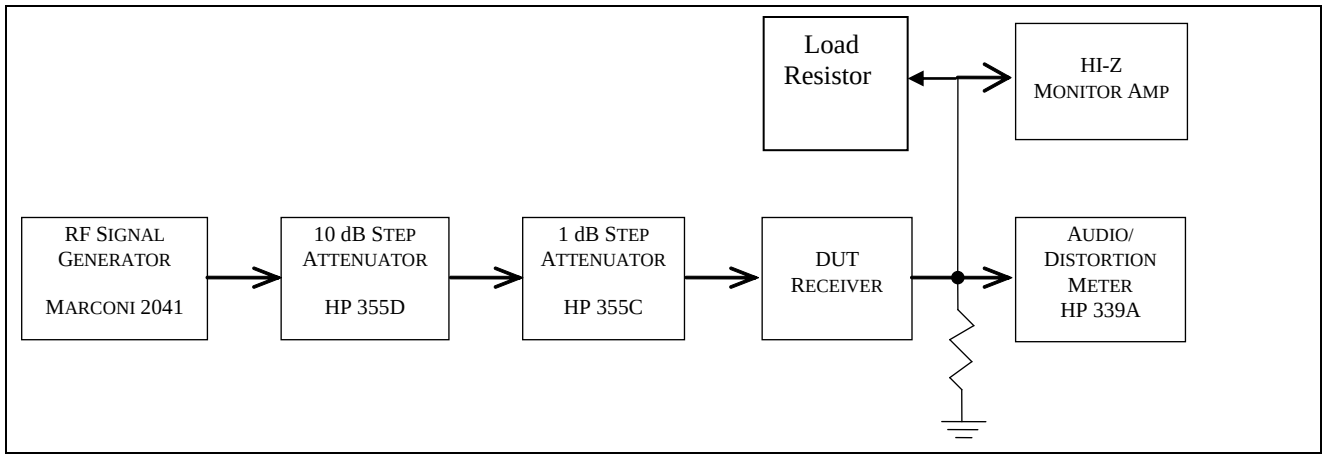



Fig. 5-11 — Audio Power Output Test Hook-up

5.11.3 Test Procedure

5.11.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | USB |
| | AGC | Enabled |
| | Band Selector | 20 Meters |
| | Frequency Tune | 14.200 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | Widest Available |
| | Any Audio Filter | Disabled |
| 10-dB Step Attn | Attenuator | 10 dB |
| 1-dB Step Attn | Attenuator | 0dB |
| RF Generator | FREQUENCY TUNE | 14.200 MHz |
| | OUT LEVEL | –80 dBm |
| | RF | ON |
| | AM | OFF |
| | FM | OFF |
| Audio/Distortion Meter | FUNCTION | DIST |
| | RELATIVE ADJUST | Any |
| | FILTERS | All OFF (Out) |
| | INPUT RANGE | 30 V |
| | DISTORTION RANGE | 30% |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |
| | FREQUENCY | 1.0 X 1 kHz |

5.11.3.2 Receiver hiss should be heard. Adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.8.3.3.

5.11.3.3 Select the appropriate speaker load resistor which matches the DUT audio output impedance.

511.3.4 Increase the generator OUTPUT LEVEL control, if necessary, until the signal is just heard in the receiver.

5.11.3.5 Tune the DUT for peak response on the receiver S meter. Increase the OUTPUT LEVEL control for an S9 signal.

5.11.3.6 Tune the DUT for approximately 1 kHz audio output. Carefully adjust the DUT tuning control so that both lights above the FREQUENCY control should are out. Disconnect or turn-off the HI-Z amplifier to eliminate the otherwise loud and unpleasant 1,000 kHz tone during this test.

5.11.3.7 Slowly increase the DUT audio gain until the meter indicates the manufacturer's specified distortion. If unspecified, increase the audio gain for a distortion of 10%. (If the distortion is under 10% at maximum volume, be sure to note the maximum distortion level on the Data Sheet.)[3]

[3] If the manufacturer specifies audio power output at other than 10% THD, or other than an 8Ω load, test to specified conditions.

5.11.3.8 Change the Meter FUNCTION control to INP LEVEL. Adjust the INPUT RANGE as necessary (over range lights should be out.) Read the audio output voltage on the correct scale of the Distortion/Audio meter and record on the Data Sheet.

5.11.3.9 Calculate the audio power using the following equation:

$P_{audio} = V^2/8$ Also, record THD% with 1 $V_{rms}$ Audio Level.

If the manufacturer's specified load used for this test was other than 8 ohms, be sure to substitute the correct load impedance when calculating power output. Record on the Data Sheet.

## 5.12 AUDIO AND IF FREQUENCY RESPONSE TEST

5.12.1 The purpose of the Audio and IF Frequency Response Test is to measure the audio frequencies at which the receiver audio output drops by 6 dB from the peak signal response. The frequency response is then calculated by taking the difference between the upper and lower frequency.

5.12.2 Test hook-up (See Fig. 5-12)

**NOTE**: If proceeding from the previous Audio Power Output Test, only the hook-up changes shown in the diagram are required. Proceed to step 5.12.3.

5.12.2.1 With all power switches in the OFF position and the transmitter function disabled to the fullest extent possible, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen OUTPUT to 10 dB Step Attn INPUT | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn OUTPUT to 1-dB Step Attn INPUT | BNC to BNC | 50-Ohm Coax |
| 1-dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50-Ohm Coax |
| DUT AUDIO OUTPUT to Dist/Audio Meter INPUT | As Required to BNC | 50-Ohm Coax |
| Speaker Load to Oscilloscope Input | As Required to BNC | 50-Ohm Coax |
| Speaker Load/HI-Z Amp Across Dist/Audio Meter INPUT | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

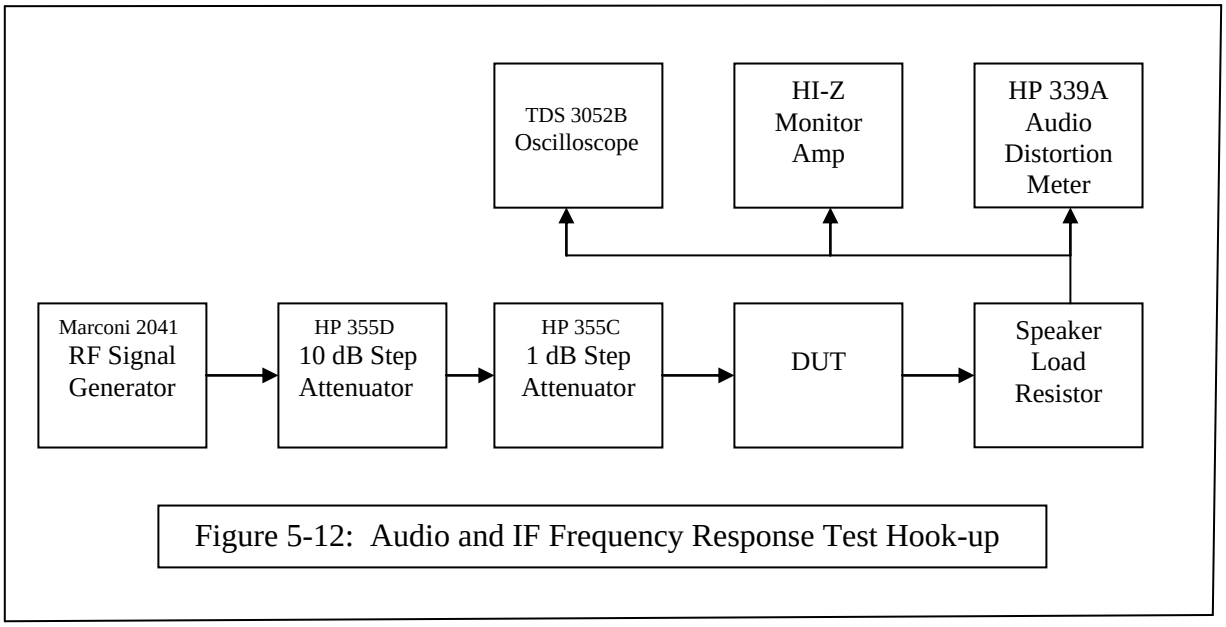



Figure 5-12: Audio and IF Frequency Response Test Hook-up

5.12.3 Test Procedure

5.12.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | USB |
| | Band Selector | 20 Meters |
| | Frequency | 14.200 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | Widest Normal SSB |
| | AGC | ON |
| 10-dB Step Attn | Attenuator | –70 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| RF Generator | FREQUENCY | 14.200 MHz |
| | OUTPUT LEVEL | –110 dBm |
| | RF | ON |
| | AM | OFF |
| | FM | OFF |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All OFF (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.12.3.2 Set the DUT, Signal Generator and Audio Distortion Meter and Frequency Counter power switches to ON. Receiver hiss should be heard. Adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.12.3.3.

5.12.3.3 Set the RF generator to 14.200 MHz. Tune the DUT receiver to the Generator frequency. Rotate the generator OUTPUT LEVEL control to a point where the S meter on the DUT just before it starts to move. Turn off the AGC.

5.12.3.4 Adjust the INPUT RANGE control as required to maintain approximately a mid-scale meter indication while carefully tuning the Generator for peak signal response on the meter.

**NOTE:** Set the Generator to 20 dB above the MDS level if the AGC cannot be disabled.

5.12.3.6 Record the IF filter bandwidth in 5.12.3.6 of the data sheet. Set the RELATIVE ADJUST for a –1 dB indication on the meter. Tune the Generator downward in frequency until the meter indicates –7 dB. Adjust the oscilloscope for amplitude and frequency to show several sine waves. Using the *measure* function on the oscilloscope, the frequency of the waveform is displayed. Record Frequency indication in 5.12.3.6 of the Data Sheet. Return the Generator output to its initial level and the DUT to the peak response frequency. The meter should again indicate –1 dB.

5.12.3.7 Tune the Generator upward in frequency until the meter indicates –7 dB. Adjust the oscilloscope for amplitude and frequency to show several sine waves. Using the *measure* function on the oscilloscope, the frequency of the waveform is displayed. Record the Frequency indication in 5.12.3.7 of the Data Sheet. Return the Generator output to its initial level and the DUT to the peak response frequency. The meter should again indicate –1 dB. Subtract the lower –6 dB Frequency indication from the upper and record this difference in the space provided.

5.12.3.8 Tune the generator for peak response and repeat paragraphs 5.12.3.6 to 5.12.3.7 for all remaining DSP, SSB and CW IF filters and IF filter combinations.

*AM Audio Frequency Response*

5.12.3.9 Connect the output of the HP-8116A Function Generator to the EXT MOD INPUT of the RF Generator. Adjust the generators to provide an S8 to S9 signal with 80% modulation at approximately 1000 Hz.

5.12.3.10 Tune the DUT for peak audio response as indicated by the Audio Meter. Increase the modulation frequency until approximately a 6 dB drop is observed. Again, tune the DUT for peak audio response. Record the IF filter bandwidth in 5.12.3.10 of the data sheet. Record the modulation frequency that produces peak meter response in 5.12.3.10 of the Data Sheet.

5.12.3.11 Adjust the modulation frequency for peak response. Set the RELATIVE ADJUST for a –1 dB indication on the meter. Reduce the modulation frequency until the meter indicates –7 dB. Record the modulation frequency in 5.12.3.10 of the Data Sheet. Return the modulation to its initial frequency and the meter should again indicate –1 dB.

5.12.3.12 Increase the modulation frequency until the meter indicates –7 dB. Record the modulation frequency in 5.12.3.10 of the Data Sheet. Return the modulation to its initial frequency. The meter should again indicate –1 dB. Subtract the lower –6 dB frequency indication from the upper and record this difference in the space provided.

5.12.3.13 Repeat paragraphs 5.12.3.10 to 5.12.3.12 for all remaining AM filters.

## 5.13 SQUELCH SENSITIVITY TEST

5.13.1 The purpose of the Squelch Sensitivity Test is to determine the level of the input signal required to break the squelch at the threshold point . This test is performed for both FM and SSB.

5.13.2 Test hook-up (See Fig. 5-13)

> **NOTE**: If proceeding from the previous Audio & IF Frequency Response Test, no hook-up changes except removal of the oscilloscope is required. Proceed to step 5.13.3.

5.13.2.1 With all power switches in the OFF position and the transmitter disabled, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen OUTPUT to 10-dB Step Attn INPUT | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn OUTPUT to 1-dB Step Attn IN | BNC to BNC | 50-Ohm Coax |
| 1-dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50-Ohm Coax |
| DUT Audio OUTPUT to Dist/Audio Meter IN | As Required to BNC | 50-Ohm Coax |
| 8-Ohm Load/HI-Z Amp Across Dist/Audio Meter INPUT | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

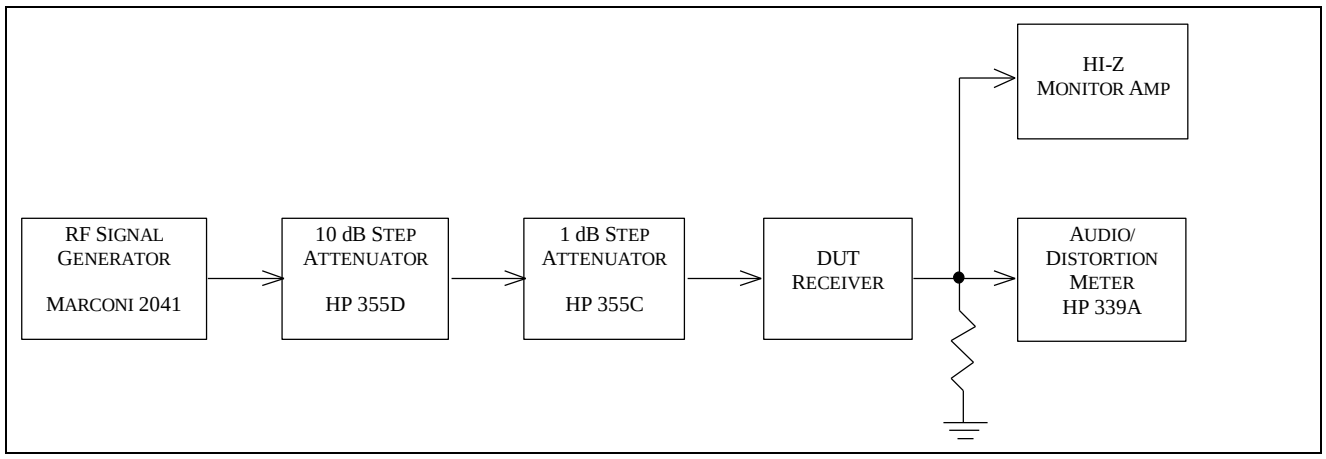



Fig. 5-13 — FM Squelch Sensitivity Test Hook-up

5.13.3 Test Procedure

5.13.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | FM |
| | Band Selector | 10 Meters |
| | Frequency | 29.000 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | Suitable for FM |
| | AGC | OFF |

| | | |
|---|---|---|
| | Preamp | ON |
| | Any Audio Filter | Disabled |
| 10-dB Step Attn | Attenuator | 10 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| RF Generator | FREQUENCY TUNE | 29.000 MHz |
| | OUT LEVEL | –110 dBm |
| | AM | OFF |
| | FM | INT |
| | DEVIATION | 3 kHz |
| | MODULATION FREQ | 1000 Hz |
| | RF | ON |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All OFF (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.13.3.2 Receiver hiss should be heard. Adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.13.3.3.

5.13.3.3 Increase the Generator OUTPUT LEVEL control, if necessary, until the signal is just heard in the receiver. Adjust the INPUT RANGE control as required to maintain approximately a mid-scale meter indication while carefully tuning the receiver for minimum distortion (FM) or peak signal response (SSB) as appropriate.

5.13.3.4 Set the RF Generator RF switch to OFF. Carefully adjust the squelch control on the DUT to just past the threshold point, closing the receiver. Set the RF switch back to ON. *Slowly* increase the Generator output to a level that holds the receiver open continuously without dropping out. (Use the step attenuator controls as necessary to obtain the threshold point.) Note the attenuator settings and generator output. Calculate the level of the input signal to the DUT and record on the Data Sheet. Return the step attenuators to their initial settings.

5.13.3.5 Repeat paragraphs 5.13.3.3 to 5.13.3.4 for the following frequency bands, as appropriate to the DUT:
52 MHz, 146 MHz and 440 MHz.

5.13.3.7 Return the attenuator controls to their initial settings. Set the generator for 14.200 and the FM switch to OFF. Set the DUT for USB at 14.200 MHz. Repeat paragraphs 5.13.3.4 through 5.13.3.4, if squelch is available for USB, EXCEPT this time, set the Generator OUTPUT LEVEL to a point where the squelch opens momentarily and closes again. Calculate the level of the input signal and record on the Data Sheet.
(**Note: AGC may have to be enabled**)

**5.14 S METER TEST**

5.14.1 The purpose of the S Meter Test is to determine the level of RF input signal required to produce an S9 indication on the receiver S Meter. This test is performed with the DUT in the CW mode at frequencies of 1.020, 14.020, 52.020, 146.020 and 440.020 MHz as appropriate for the DUT. An S9+20 dB signal is also used to check for significant S Meter deviation. The IF filter is set for the closest bandwidth to 500 Hz.

5.14.2 Test hook-up (See Fig. 5-14)

**NOTE**: If proceeding from the previous Squelch Sensitivity Test, no hook-up changes are required. Proceed to step 5.14.3.

5.14.2.1 With all power switches in the OFF position and the transmitter disabled, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen OUTPUT to 10-dB Step Attn INPUT | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn OUTPUT to 1-dB Step Attn IN | BNC to BNC | 50-Ohm Coax |
| 1-dB Step Attn OUTPUT to DUT RF INPUT | BNC to As Required | 50-Ohm Coax |
| DUT Audio Output to Dist/Audio Meter Input | As Required to BNC | 50-Ohm Coax |
| 8 Ohm Load/HI-Z Amp to Dist/Audio Meter Input | As Required | As Required |
| Power Source to DUT Power Input | As Required | As Required |

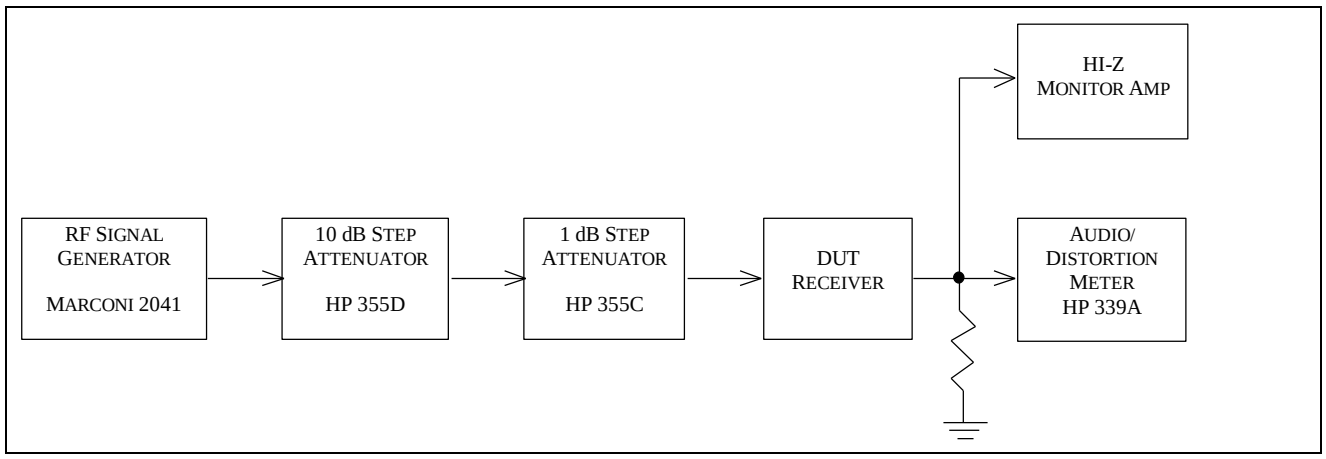



Fig. 5-14 — S-meter Test Hook-up

5.14.3 Test Procedure

5.14.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | SSB |
| | Band Selector | 20 Meters |
| | Frequency | 14.200 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | Closest to 500 Hz |

| | | |
|---|---|---|
| | AGC | ON |
| | Preamp | OFF |
| 10-dB Step Attn | Attenuator | –10 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| RF Generator | FREQUENCY TUNE | 14.200 MHz |
| | OUT LEVEL | –100 dBm |
| | RF | ON |
| | AM | OFF |
| | FM | OFF |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All OFF (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

5.14.3.2 Receiver hiss should be heard. Adjust the volume to the desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.14.3.3.

5.14.3.3 Rotate the Generator OUTPUT LEVEL control as required until the signal is just heard in the receiver. Adjust the INPUT RANGE control as required to maintain approximately a mid-scale meter indication while carefully tuning the receiver for peak signal response.

5. 14.3.4 Increase the Generator OUTPUT LEVEL control for approximately an S9 indication. Adjust the DUT tuning control for maximum S Meter indication, then set the Generator output for an exact indication of S9. Rock the DUT tuning control back-and-forth to ensure the S Meter is still peaked on the signal. Note the Generator output and attenuator settings and record the input signal level in microvolts to the DUT in the 5.14.3.4 of the Data Sheet.

5.14.3.5 Increase the Generator output for an S9+20 dB S Meter indication. Check the appropriate box in 5.14.3.5 if an approximately 20 dB increase in generator output was required. If not, record levels and settings as in step 5.14.3.4.

5.14.3.6 Repeat steps 5.14.3.4 through 5.14.3.5 with DUT preamp on.

5.14.3.7 Return DUT preamp to off and repeat steps 5.14.3.3 to 5.14.3.6 for frequencies of 1.020, 52.020, 146.020 and 432.020 MHz.

**5.15 IN-BAND IMD TEST**

5.15.1 The purpose of the In-Band IMD Test is to measure the intermodulation-distortion (IMD) products present in the audio output of the receiver. *This test is typically performed on units that will undergo the expanded set of tests.* The receiver will be operated in the SSB mode at 14.200 MHz. Two input signals, spaced 200 Hz apart, are applied to the DUT and adjusted for equal output amplitude. (The audio tones are approximately 900 and 1100 Hz.) The input signals are adjusted for approximately a single tone n S-9 level and the audio output is observed on the Spectrum Analyzer. The tones are set for a 0 dB level for a single tone reference. (This is unlike the transmit two-tone test, where the -6 dB is used for a PEP reference level.) The AGC fast option is used for this test.

5.15.2 Test Hook-up (See FIGURE 5-15)

Note: If proceeding from the previous S-Meter Test, only the hook-up changes shown with a dotted line are required.

5.15.2.1 With all power switches in the OFF position, the transmitter function disabled to the fullest extent possible and each Generator RF OFF, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen 1 Output to Hybrid Combiner Input 1 | BNC to BNC | 50-Ohm Coax |
| Signal Gen 2 Output to Hybrid Combiner Input 2 | BNC to BNC | 50-Ohm Coax |
| Hybrid Combiner Output to 10-dB Step Attn Input | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn Output to 1-dB Step Attn IN | BNC to BNC | 50-Ohm Coax |
| 1-dB Step Attn Output to DUT RF Input | BNC to As Req. | 50-Ohm Coax |
| 8 Ohm Load / HI-Z Amp across DUT Audio Output | As Req. | As Required |
| DUT Audio Output to Audio Attn Pad Input | As Req. to BNC | 50-Ohm Coax |
| Audio Attn Pad Output to Spectrum Analyzer Input | BNC to BNC | Coax |
| Power Source to DUT Power Input | As Req. | As Req. |

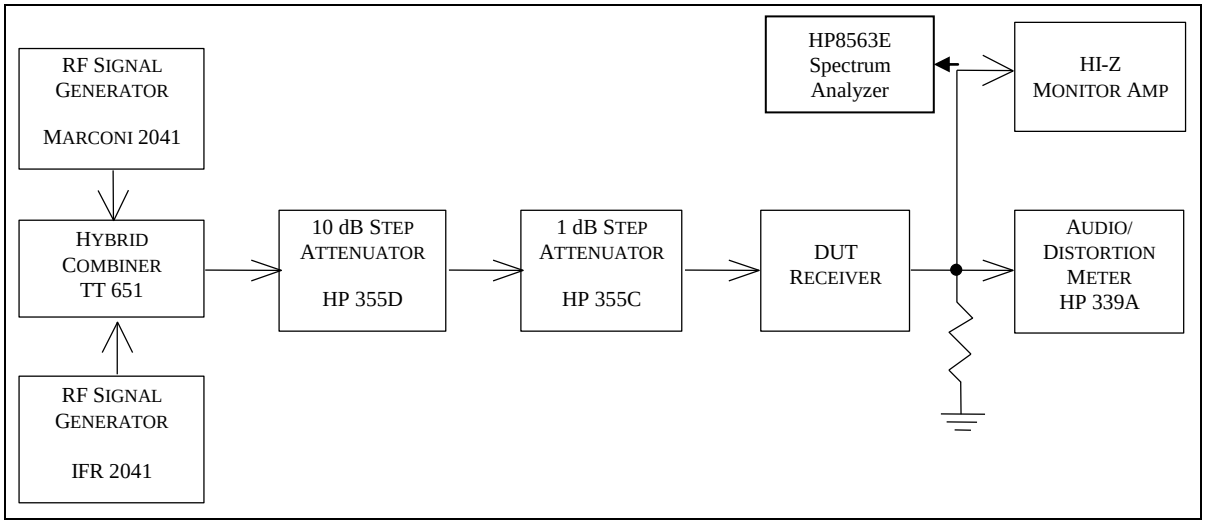


Figure 5-15 In-Band IMD Test Setup

5.15.3 Test Procedure

5.15.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | USB |
| | Band Selector | 20 Meters |
| | Frequency | 14.20000 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | SSB Widest Available |
| | Any Audio Filtering | Disabled |
| | AGC | Fastest Available |
| | AF Gain (Volume Control) | Minimum |
| 10-dB Step Attn | Attenuator | 10 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| RF Generator #1 | CARRIER FREQ | 14.2009 MHz |
| | RF LEVEL | –63 dBm |
| | CARRIER ON-OFF | ON |
| | AM | OFF |
| | FM | OFF |
| | NOISE MODE (UTIL) | LOW NOISE |
| RF Generator #2 | CARRIER FREQ | 14.2011 MHz |
| | RF LEVEL | –63 dBm |
| | CARRIER ON-OFF | ON |
| | AM | OFF |
| | FM | OFF |
| | NOISE MODE (UTIL) | LOW NOISE |

| Instrument | Control | Position |
|---|---|---|
| Spectrum Analyzer | START FREQ - STOP FREQ | 0 - 2 kHz |
| | REF LEV (AMPLITUDE) | –10 dBm |
| | ATTEN (AMPLITUDE) | 20 dB |
| | RES (BW) | 10 Hz |
| | VIDEO BW (BW) | 10 kHz |
| | THRESHOLD (DISPLAY) | -80 dBm |
| | SWP TIME (SWEEP) | AUTO (1.3 s) |

CAUTION: The input to the spectrum analyzer at no time should be greater than +10 dBm (0.707 Vrms). Damage to this instrument will occur at an input level of +30 dBm (7.07 Vrms) or greater. It is recommended to monitor the audio on the analyzer display at all times during this test - especially when increasing the AF Gain Control or decreasing the variable attenuators.[1]

5.15.3.2 Slowly increase the AF Gain control of the DUT. Receiver hiss should be heard. Adjust the volume to the desired level. Be careful not to exceed the -10 dBm reference limit on the analyzer display. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.15.3.3.

5.15.3.3 Tune the DUT for peak response as indicated by the receiver's S-Meter. Adjust #1 Generator RF LEVEL control, if necessary, for an S-9 indication. Set the #2 Generator RF switch to ON and set for the same output level as Generator #1. Both pips should now be clearly visible on the Spectrum analyzer display

5.15.3.4 Adjust the DUT Tuning control, if necessary, for the two tones to be centered on the spectrum analyzer display. The scan rate may be increased temporarily by adjusting the SWP TIME (SWEEP) control. (NOTE: Analyzer calibration is lost once the scan rate is increased beyond the point MEAS UNCAL appears on the screen.) Adjust the attenuators and DUT AF Gain control for an approximate indication of -10 dBm (top reference line) on the spectrum analyzer.

5.15.3.5 If necessary, slow the scan rate down to a point where the analyzer is again in calibration. Carefully adjust the #2 RF Generator RF LEVEL control for the second pip to be equal to the first pip. Adjust the DUT audio output so that both pips are now at the -10 dBm reference (top) line of the spectrum analyzer display. Re-adjust, if necessary, the #2 RF Generator RF LEVEL Control if the two tones are no longer equal in amplitude.

5.15.3.6 Take a single sweep by depressing the SGL SWEEP (SWEEP) button. Plot and save to an appropriately named file.

5.15.3.7 Repeat if desired for any other AGC options as deemed appropriate by the Test Engineer.

[1] "Woe be unto he who breaks this thing." ---- Edward F. Hare, W1RFI

## 5.16 NOTCH FILTER TEST

5.16.1 The purpose of the Notch Filter Test is to determine the notch depth of any and all applicable receiver filters. Two 20–meter input signals, spaced 1200 Hz apart, are applied to the DUT and adjusted for equal output amplitudes. The receiver notch filter is then set to null the undesired upper audio tone. The lower audio tone is adjusted to provide a 0–dB reference on the spectrum analyzer display. The depth of the notch filter is then determined by comparing the difference in levels between the two tones. Similar tests are performed with receivers having DSP auto-notch capabilities. In addition, the time to notch, or notch attack time will also be measured for DSP units.

5.16.2 Test Hook-up (See FIGURE 5-16) Note: If proceeding from the previous In-Band IMD Test, only the hook-up changes shown with a dotted line are required.

5.16.2.1 With all power switches in the OFF position, the transmitter function disabled to the fullest extent possible and each Generator RF OFF, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen 1 Out to Hybrid Combiner IN 1 | BNC to BNC | 50-Ohm Coax |
| Signal Gen 2 Out to Hybrid Combiner In 2 | BNC to BNC | 50-Ohm Coax |
| Hybrid Combiner Output to 10-dB Step Attn Input | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn Output to 1-dB Step Attn Input | BNC to BNC | 50-Ohm Coax |
| 1-dB Step Attn Output to DUT RF Input | BNC to As Req. | 50-Ohm Coax |
| DUT Audio Output to Dist/Audio Meter IN | As Req. to BNC | 50-Ohm Coax |
| 8 Ohm Load / HI-Z Amp across Dist/ Audio Meter Input | As Req. | As Req. |
| 20 dB Audio Pad (Attn) Output to Spectrum Analyzer Input | BNC to BNC | Coax |
| Power Source to DUT Power Input | As Req. | As Req. |

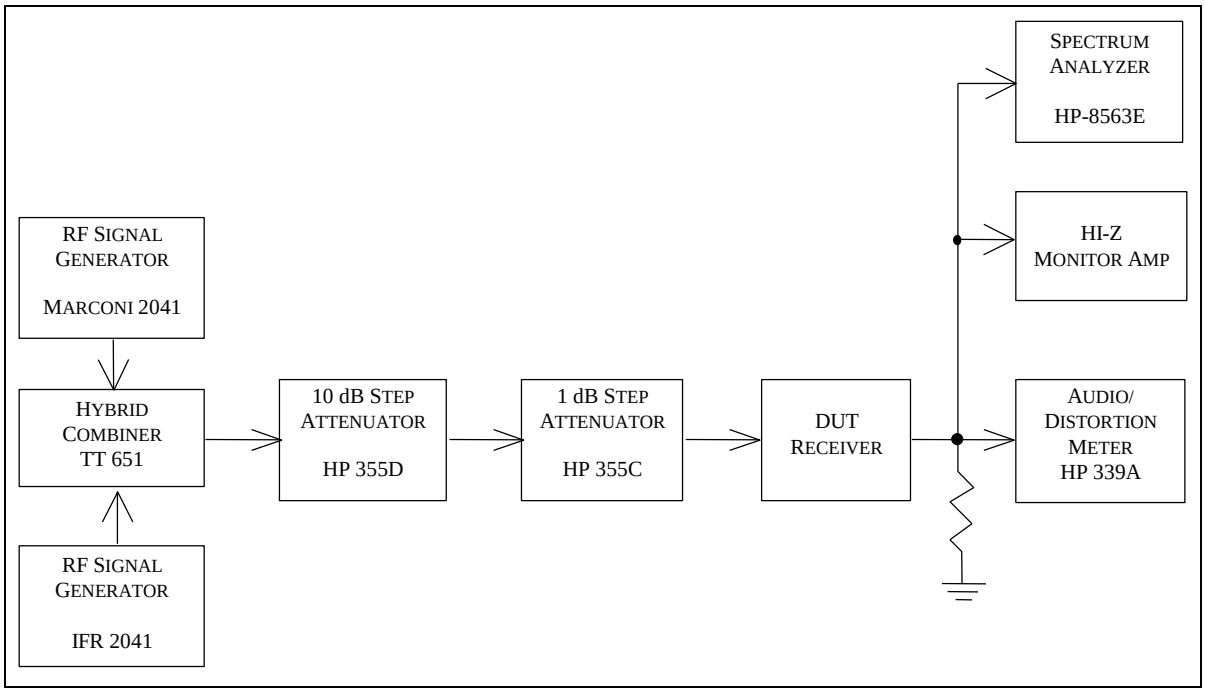



Figure 5-16 Notch Filter Test Hook-Up

5.16.3 Test Procedure

5.16.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | USB |
| | Band Selector | 20 Meters |
| | Frequency | 14.200 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | SSB Widest Available |
| | Any Audio Filtering | Disabled |
| | AGC | OFF (if available) or Fast |
| | Notch Filter | OFF |
| | AF Gain (Volume Control) | Minimum |
| 10-dB Step Attn | Attenuator | 10 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| RF Generator #1 | CARRIER FREQ | 14.2000 MHz |
| | RF LEVEL | –80 dBm |
| | CARRIER ON-OFF | ON |
| | AM | OFF |
| | FM | OFF |
| | NOISE MODE (UTIL) | LOW NOISE |
| RF Generator #2 | CARRIER FREQ | 14.2012 MHz |
| | RF LEVEL | –80 dBm |
| | CARRIER ON-OFF | ON |
| | AM | OFF |
| | FM | OFF |
| | NOISE MODE (UTIL) | LOW NOISE |

| | | |
|---|---|---|
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All Off (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |
| Spectrum Analyzer | START FREQ - STOP FREQ | 0 – 3 kHz |
| | REF LEV (AMPLITUDE) | –10 dBm |
| | ATTEN (AMPLITUDE) | 20 dB |
| | RES (BW) | 10 Hz |
| | VIDEO BW (BW) | 10 kHz |
| | THRESHOLD (DISPLAY) | OFF |
| | SWP TIME (SWEEP) | AUTO |

CAUTION: The input to the spectrum analyzer at no time should be greater than +10 dBm (0.707 Vrms). Damage to this instrument will occur at an input level of +30 dBm (7.07 Vrms) or greater. It is recommended to monitor the audio on the analyzer display at all times during this test - especially when increasing the AF Gain Control or decreasing the variable attenuators.

5.16.3.2 Slowly increase the AF Gain control of the DUT. Receiver hiss should be heard. Adjust the volume to the desired level. Be careful not to exceed the –10 dBm reference limit on the analyzer display. Allow all equipment at least 10 minutes warm-up time before proceeding to step 5.16.3.3.

5.16.3.3 Adjust the #1 Generator RF LEVEL control until the signal is just heard in the receiver. Adjust the INPUT RANGE and receiver controls as required to maintain approximately a mid-scale meter indication while carefully tuning the receiver for peak signal response on the audio level meter.

5.16.3.4 Set the #2 Generator RF switch to OFF. Peak the DUT tuning control for maximum response as indicated by the audio level meter.

5.16.3.5A If the DUT AGC is OFF, adjust the #1 RF generator for a level approximately 10 dB below the 1-dB compression point. Recheck the receiver tuning for peak response on the Distortion Meter. The audio output frequency should be approximately 500 to 1,000 Hz. If necessary, adjust the receiver tuning to bring the output audio into this range while maintaining a peak response on the meter as best as is possible.

5.16.3.5B If the DUT AGC is set to Fast, increase the #1 RF generator for approximately an S-9 indication on the DUT S-Meter. (The input signal to the DUT should be approximately 50 μV.) Recheck the receiver tuning for peak response on the Distortion Meter. The audio output frequency should be approximately 500 to 1,000 Hz. Adjust, if necessary, the receiver tuning to bring the output audio into this range while maintaining a peak response on the meter as best as is possible.

5.16.3.6 Adjust the Distortion Analyzer INPUT LEVEL and RELATIVE ADJUST Controls for a meter reading of –1 dB. Turn on the notch filter and observe any change in the meter indication. Adjust the notch filter for the minimum meter reading *change* between Notch Filter On and Off. Record the difference between the –1 dB reference and the meter reading in 5.16.3.6 of the Data Sheet. Return the notch filter adjustment to its upper or lower limit.

5.16.3.7 Observe the receiver's output audio on the spectrum analyzer. The scan rate may be increased temporarily by adjusting the SWP TIME (SWEEP) control. (NOTE: The analyzer will not be in calibration if the scan rate is increased beyond the point MEAS UNCAL appears on the display.) Adjust the attenuators and DUT AF Gain control for an indication of –10 dB on the spectrum analyzer.

5.16.3.8 Set the #2 RF Generator RF switch to ON. A second tone that is 1,200 Hz greater in pitch should be heard in the receiver output. Adjust the Spectrum Analyzer CENTER FREQ (FREQUENCY) Control for this second tone to appear right on the display vertical centerline. Adjust the #2 RF Generator RF LEVEL Control for the second pip to be equal to the first pip. (The first pip should now be visible just over two divisions to the left of the display centerline.)

5.16.3.9 If necessary, slow the scan rate down to a point where the analyzer is again in calibration. Adjust the spectrum analyzer input step attenuator for the two pips to be at the –10 dBm reference (top) line of the spectrum analyzer display. Re-adjust, if necessary, the #2 RF Generator RF LEVEL Control if the two tones are no longer equal in amplitude.

5.16.3.10 Set the #1 RF Generator RF switch to OFF. Adjust the notch filter for the best null of the remaining higher pitched tone. Both the ear and the spectrum analyzer display may be used to determine the point of best null. The scan rate may again be temporarily increased to facilitate this adjustment. Be sure to slow it down so that the display is once again calibrated before proceeding to the next step.

5.16.3.11 Set the #1 Signal Generator RF Switch to ON. Adjust, if necessary, the Step Attenuators and DUT AF GAIN Control for the #1 Generator tone for the -10 dBm reference on the display. The notched #2 Generator tone should now be on the vertical centerline. The notch depth may now be read directly on the display. A single sweep may now be taken to facilitate this measurement by depressing the SGL SWEEP Button. Record in 5.16.3.11 of the Data Sheet.

5.16.3.12 Repeat for all other available DSP, IF and AF notch filters and notch filter combinations. Also try the following combinations:

| | |
|---|---|
| 2 tones (S9 each), AGC on, manual notch | 2 tones (S9 + S1), AGC on, manual notch |
| 1 tone, AGC on, manual notch | 1 tone, AGC off, manual notch |
| 1 tone, AGC on, auto notch | 1 tone, AGC off, auto notch |

Note changes in tone level and noise level. Also note changes in S-meter for AGC on tests.

*Auto-Notch Attack Time: Proceed only if DUT has DSP Auto-Notch Feature*

5.16.3.13 With all power switches in the OFF position and the transceiver in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen Output to 10-dB step INPUT | Type N to BNC | 50-Ohm coax |
| 10-dB step Attn OUT to 1-dB step IN | BNC to BNC | 50-Ohm coax |
| 1-dB step Attn IN to DUT RF Input | BNC to Type N | 50-Ohm coax |
| Signal Gen Output to Scope CH1 INPUT | BNC clip to BNC | 50-Ohm coax |
| DUT Audio Output to Scope CH2 INPUT | As Required to BNC | 50-Ohm coax |
| DUT Audio Output to 8 Ohm Load | As Required to BNC | 50-Ohm coax |

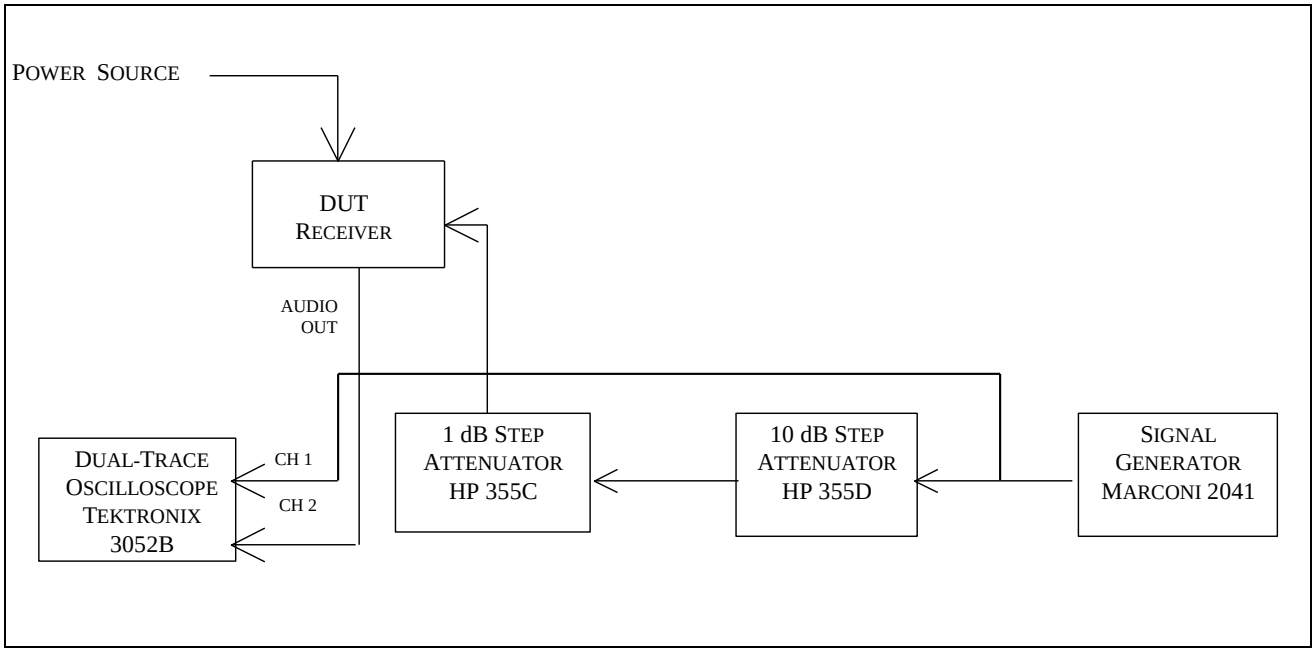



Fig. 5-16A — Notch Attack Time Test Hook-up

5.16.3.14 Turn the DUT and test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | USB |
| | Band Selector | 20 Meters |
| | Frequency | 14.200 MHz |
| | XMIT/RCV | Receive |
| | DRIVE or RF LEVEL | Minimum |
| | AGC | OFF |

| Instrument | Control | Position |
|---|---|---|
| RF Generator | FREQUENCY TUNE | 14.200 MHz |
| | OUT LEVEL | –63 dBm |
| | RF | ON |
| | AM | OFF |
| | FM | OFF |

| | | |
|---|---|---|
| 10-dB Step Attn | Attenuator | 10 dB |
| 1-dB Step Attn | Attenuator | 0 dB |

| | | |
|---|---|---|
| Oscilloscope | CH1 VOLTS/DIV | 0.5 V |
| | CH1 Coupling | DC |
| | CH1 Position | As Needed |
| | CH2 VOLTS/DIV | 1 V |
| | CH2 Coupling | DC |
| | CH2 Position | As Needed |
| | TIME/DIV | 5 ms |
| | TRIG MODE | AUTO |
| | Trigger Coupling | DC |
| | Trigger Source | CH1 |
| | Trigger Level | as required |

5.16.3.15 Set the Signal Generator RF to ON. Tune the DUT for peak response. Set the DUT AF Gain control for comfortable listening level and adjust CH1 and CH2 VOLTS/DIV for suitable display similar to Figure 5.16B. Set the Signal Generator RF to OFF.

5.16.3.16 Place the oscilloscope into the single sweep storage mode and clear the display. Key the Signal Generator by pressing the RF to button to ON. Repeat and adjust the trigger level until a single sweep occurs every time the generator is keyed.

5.16.3.17 Measure the time delay for from the point the signal appears until it is notched by at least 50%. The oscilloscope sweep rate may be adjusted as required to facilitate this measurement. Make several sweeps and compare results. Record on data sheet.

**5.17 DSP NOISE REDUCTION TEST**

5.17.1 The purpose of this test is to determine the level of noise reduction possible with receivers equipped with DSP. An uninterrupted un-modulated CW will be mixed with noise and fed to the DUT RF input. The audio output will be observed on the spectrum analyzer. Noise level comparisons will be made with the DSP Noise Reduction on and off..

5.17.2 Test Hook-up (See FIGURE 5.17)

Note: If proceeding from the previous test, only the hook-up changes shown with a dotted line are required.

5.17.2.1 With all power switches in the OFF position, the transmitter function disabled to the fullest extent possible and each Generator RF OFF, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|
| Signal Gen Output to Hybrid Combiner INPUT | BNC to BNC | 50-Ohm Coax |
| Elecraft Noise Gen Output to 10-dB Step Attn Input | BNC to BNC | 50-Ohm Coax |
| 10-dB Step Attn Output to 1-dB Step Attn Input | BNC to BNC | 50 Ohm Coax |
| 1-dB Step Attn Output to Hybrid Combiner Input | BNC to As Req. | 50 Ohm Coax |
| Hybrid Combiner Output to DUT RF Input | BNC to As Req. | 50 Ohm Coax |

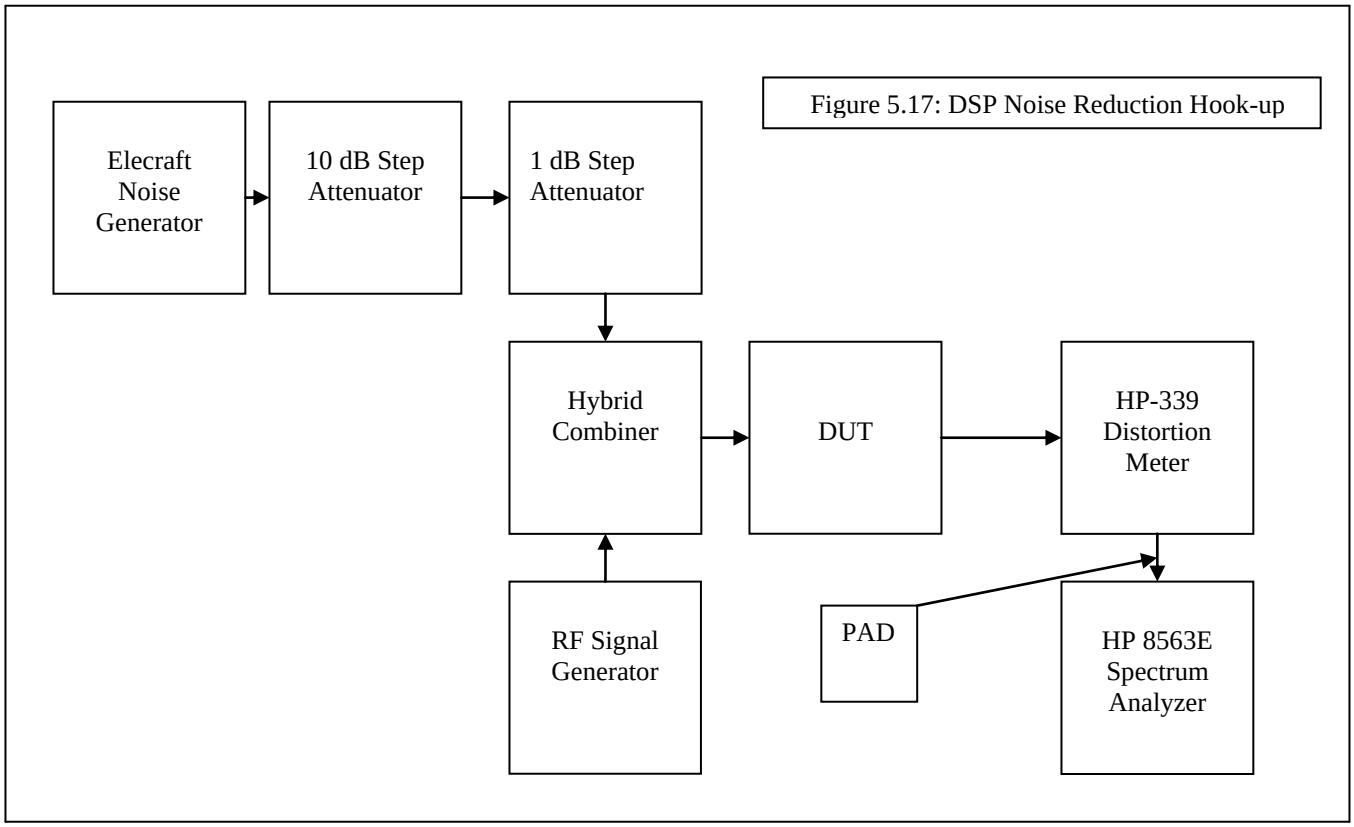



Figure 5.17: DSP Noise Reduction Hook-up

| Connections | Cable Type | Connectors |
|---|---|---|
| 8 Ohm Load / HI-Z Amp across DUT Audio OUT | As Req. | As Req. |
| DUT Audio Output to Audio Attn Pad Input | As Req. to BNC | 50-Ohm Coax |
| Audio Attn Pad Output to Spectrum Analyzer Input | BNC to BNC | Coax |
| Power Source to DUT Power Input | As Req. | As Req. |

5.17.3 Test Procedure

5.17.3.1 Turn the DUT and all test equipment power switches to ON. Set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | USB |
| | Band Selector | 20 Meters |
| | Frequency | 14.200 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | SSB Widest Available |
| | Any Audio Filtering | Disabled |
| | AGC | OFF |
| | AF Gain (Volume Control) | Minimum |
| 10-dB Step Attn | Attenuator | 10 dB |
| 1-dB Step Attn | Attenuator | 0 dB |
| RF Generator | CARRIER FREQ | 14.2000 MHz |
| | RF LEVEL | –63 dBm |
| | CARRIER ON-OFF | OFF |
| | AM | OFF |
| | FM | OFF |
| | NOISE MODE (UTIL) | LOW NOISE |
| Spectrum Analyzer | START FREQ - STOP FREQ | 0 - 5 kHz |
| | REF LEV (AMPLITUDE) | –10 dBm |
| | ATTEN (AMPLITUDE) | 20 dB |
| | RES (BW) | 10 Hz |
| | VIDEO BW (BW) | 10 kHz |
| | THRESHOLD (DISPLAY) | –80 dBm |
| | SWP TIME (SWEEP) | AUTO (1.3 s) |

CAUTION: The input to the spectrum analyzer at no time should be greater than +10 dBm (0.707 Vrms). Damage to this instrument will occur at an input level of +30 dBm (7.07 Vrms) or greater. It is recommended to monitor the audio on the analyzer display at all times during this test - especially when increasing the AF Gain Control or decreasing the variable attenuators.[1]

5.17.3.2 Slowly increase the AF Gain control of the DUT. Receiver hiss should be heard. Adjust the volume to the desired level. Be careful not to exceed the -10 dBm reference limit on the analyzer display. Allow all equipment at least 10 minutes warm-up time before proceeding to step 6.2.3.3.

[1] "Woe be unto he who breaks this thing." ---- Edward F. Hare, W1RFI

5.17.3.3 Increase the Generator RF LEVEL control until the signal is just heard in the receiver. Adjust the INPUT RANGE and receiver controls as required to maintain approximately a midscale meter indication while carefully tuning the receiver for peak signal response on the Audio meter.

5.17.3.4 Increase the RF generator output for approximately an S-9 indication on the DUT S-Meter. (The input signal to the DUT should be approximately 50 μV.) Recheck the receiver tuning for peak response on the Distortion Meter. Turn on the Noise Source and adjust for approximately a 3 dB increase as indicated by the meter.

5.17.3.5 Adjust the Distortion Analyzer INPUT LEVEL and RELATIVE ADJUST Controls for a meter reading of -6 dB. Turn on the DSP Noise Reduction and observe any change in the meter indication. Adjust the Noise Reduction for best results as determined by ear and note the meter indication. Record the best case increase or decrease in 6.2.3.5 of the Data Sheet.

5.17.3.6 Vary the signal level, noise level and DSP Noise Reduction Level for the best case. Record results on data sheet.

5.17.3.7 Turn off the DSP Noise Reduction and Noise Source. Observe the receiver's output audio on the spectrum analyzer. The scan rate may be increased temporarily by adjusting the SWP TIME (SWEEP) control. (NOTE: The analyzer will not be in calibration if the scan rate is increased beyond the point MEAS UNCAL appears on the display.) Adjust the attenuators and DUT AF Gain control for an indication of -10 dB (top line) on the spectrum analyzer.

5.17.3.8 Set the DSP Noise Reduction to ON. Observe any change in the tone as shown by the spectrum analyzer. Readjust, if necessary the decrease in the DUT AF Gain Control for the -10 dB reference on the spectrum analyzer. Turn on the Noise Source and adjust so that the noise floor is 20 dB below the single tone reference. Readjust the AF Gain control if necessary for the -10 dB reference.

## 5.18 Noise Figure & Equivalent Rectangular Bandwidth

This is an automated test procedure.

Hook-up:

Audio Output to HP3561 Input
Signal Generator set to 14.020 MHz
Set level to -94 dBm

Turn on HP-3551

CVI File, open, project
Up folder
FFT plot
ffplot.prg
load (green arrow)
Set Analyzer to one minute sweep
Single sweep

On the computer, pick points where the edges and the tops meet. Ex: 350 and 845 Hz.
Calculate EqivRBW button (502 HZ)

## 5.19 Noise Figure Calculation:

Theory:
Resistor noise is -174 dBm/Hz at room temperature
500 Hz is 27 dB away from 1 Hz
Thus, an ideal receiver's noise figure would be:
-174 dBm/Hz + 27 dB = -147 dBm

Noise Figure of the receiver = 147 dBm – dBm of MDS at 20M
Ex: MDS at 20 meters, preamp off = -129.5 dBm
NF = 147 – 129.5 dBm = 17.5 dB

Calculate the Noise Figure for Preamp off, P1 and P2 if available

## VI. NON-STANDARD AND SPECIAL PURPOSE TESTS

Chapter 6 contains five non-standard and special purpose tests. These include tests for mobile radios, 9600 baud packet and obsolete tests dropped from the standard test list.

### 6.1 LOW VOLTAGE AND TEMPERATURE CHAMBER TEST

6.1.1 The purpose of the Low Voltage and Temperature Chamber Test is to determine the functionality of the mobile type DUT Transceivers at low voltage and its specified temperature limits. The temperatures and voltage used for this test are intended to simulate worst case mobile conditions within the manufacturer's specified limits. Three parameters are checked at each test point, receiver sensitivity, transmit frequency accuracy and RF power output.

6.1.2 Test hook-up (See Fig. 6-1)

6.1.2.1 With all test equipment and DUT power switches in the OFF position and the transceiver in the receive mode, set the DUT in the temperature chamber. Connect the following cables to the DUT and route them through the cable port to the specified termination at the opposite end:

| Connection | Connectors | Cable Type |
|---|---|---|
| DUT RF OUTPUT to Wattmeter INPUT | As Required to Type N | 50-Ohm Coax |
| Wattmeter OUTPUT to RF Power Attn | Type N to Type N | 50-Ohm Coax |
| RF-Power Attenuator Output to Step Attenuator Input | BNC to BNC | 50-Ohm Coax |
| Step Attenuator Output to Frequency Counter Input | BNC to BNC | 50-Ohm Coax |
| Telegraph Key to DUT KEY INPUT | As Required | As Required |
| DUT AUDIO OUTPUT to Dist/Audio Meter Input | As Required | Double Banana |
| 8-Ohm Load /HI-Z Amp Across Dist/Audio Meter Input | As Required | As Required |
| DC Power Only | As Required | As Required |
| 1) | Variable DC Power supply to AC source. | |
| 2) | Set power supply to specified voltage and connect to DUT. | |

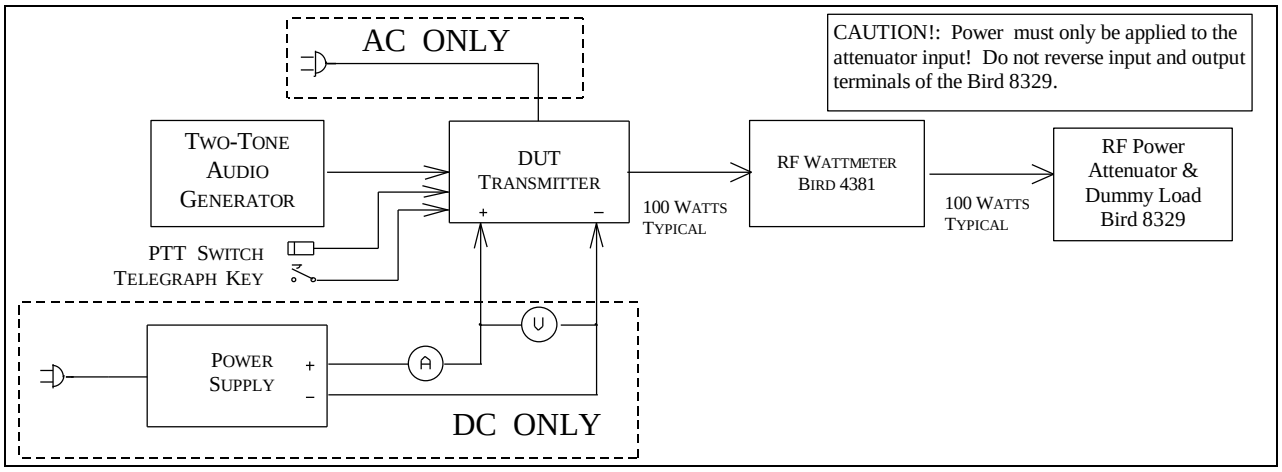



Fig. 6-1 — Chamber Test Hook-up

6.1.3 Test Procedure

6.1.3.1 Turn the DUT and RF wattmeter power switches to ON and set the following controls:

| Instrument | Control | Position |
|---|---|---|
| DUT | Mode | CW |
| | Band Selector | 20 Meters |
| | Frequency | 14.020 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | 500 Hz or Closest Available |
| | AGC | OFF |
| | Preamplifier | OFF |
| RF Generator | CARRIER FREQ | 14.020 MHz |
| | RF LEVEL | –110 dBm |
| | CARR ON-OFF | ON |
| | AM | Off |
| | FM | Off |
| | NOISE MODE | LOW NOISE |
| RF Wattmeter | Push Button mode select | FWD CW |
| | Element | As Required |
| | Forward Element Range | As Required |
| Step Attn | Attenuator | 10 dB |
| Audio/Distortion Meter | FUNCTION | REL LEVEL |
| | RELATIVE ADJUST | Center Rotation |
| | FILTERS | All Off (Out) |
| | INPUT RANGE | 30 V |
| | INPUT/GND SELECT | DIS (Center) |
| | METER RESPONSE | NORM |

6.1.3.2 Receiver hiss should be heard; adjust volume to desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 6.1.3.3

6.1.3.3 Tune the DUT per the DUT operator's manual. Put the DUT in the CW mode Depress telegraph key. Observe that some minimum level of RF power is shown by the DUT Po Meter and Bird Wattmeter. Increase the RF DRIVE/LEVEL control if necessary to the maximum rated power.

6.1.3.4 Adjust the step attenuators for an indication on the Frequency Counter. Record both the power output and the frequency counter indication in 6.1.3.4 of the Data Sheet.

6.1.3.5 Remove the DUT RF Cable from the Power Wattmeter and connect to the Signal Generator Output. Perform an MDS Test and record results on Data Sheet.

6.1.3.5 Reduce the Power Supply voltage to 11.5 V dc. Repeat steps 6.1.3.4 and 6.1.3.5 at this reduced voltage. Return the power supply to voltage to initial setting.

6.1.3.6 Record initial chamber / ambient room temperature. Set chamber for minimum specified temperature. (NOTE: If temperature is not specified, use -10 C.) Once the chamber reaches minimum temperature, let DUT reach thermal equilibrium for at least 1 hour. Repeat steps 6.1.3.4 and 6.1.3.5. Temporarily open chamber and check functionality of several randomly selected main DUT controls.

6.1.3.7 Set chamber for maximum specified temperature. (NOTE: If temperature is not specified, use +60 C.) Once the chamber reaches maximum temperature, let DUT reach thermal equilibrium for at least 1 hour. Repeat the steps of 6.1.3.4 and 6.1.3.5. Temporarily open chamber and check functionality of several randomly selected main DUT controls.

6.1.3.8 Set chamber for initial recorded ambient temperature. Once the chamber reaches initial temperature, let DUT reach thermal equilibrium for at least 1 hour. Repeat steps 6.1.3.4 and 6.1.3.5. Open chamber and check functionality of several randomly selected main DUT controls.

6.1.3.9 Record on data sheet any significant problems or lack of functionality during any of the test points.

**6.2** (Classified)

**6.3 RECEIVER BIT ERROR RATE (BER) TEST,**
**No longer part of our testing, but historical in nature.**

6.3.1 The purpose of the Receiver Bit Error Rate (BER) is to determine the usefulness of 9600 Baud reception at various input signal levels.

6.3.2 Test initialization and hook-up (See Fig. 6.3-1)

NOTE: The IEEE Interface Bus, BER Test Box to Computer COM 1 port and BERT software must be correctly installed before proceeding to this step.

6.3.2.1 Turn on the BER Test box, Marconi Generator and Test Computer. While in DOS, go to the C:\SROOM subdirectory. Type "BERT" and press Enter to start the initializing process. After completion of this process, type the DUT radio and model number when prompted to do so. Next, type "R" for Receiver Tests. The next prompt will be for the test frequency. Use either 146.000 MHz or 440 MHz, whichever is appropriate for the band being tested. Type "I" for Initial Connections and Set-Up when the next prompt appears. Connect the following test hook-up as prompted to do so:

| Connection | Connectors | Cable Type |
|---|---|---|
| BER Test Box DSP OUTPUT to Marconi EXT MOD INPUT | BNC to BNC | 50 Ohm coax |
| BER Test Box AUDIO IN to DUT RX Data Output | BNC to As Required | 50 Ohm coax |
| BER Test Box AUDIO OUT to Oscilloscope CHANNEL 2 INPUT | BNC to BNC | 50 Ohm coax |
| BER Test Box SYNC OUT to Oscilloscope CHANNEL 1 INPUT | BNC to BNC | 50 Ohm coax |
| BER Test Box Clock Input to HP-8116A OUTPUT | BNC to BNC | 50 Ohm coax |
| DUT RF Input to Marconi RF signal OUTPUT | As Required to BNC | 50 Ohm coax |

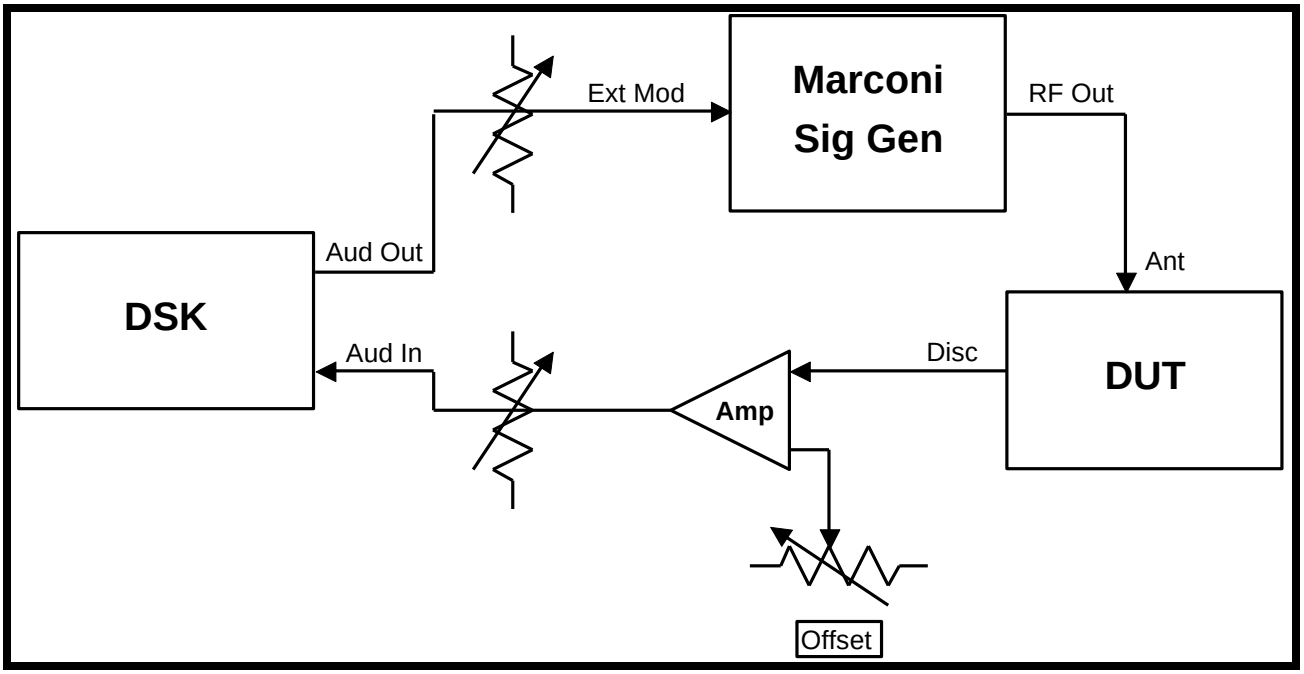



Fig. 6.3-1 — Fig 1.

6.3.3 Test Procedure

6.3.3.1 Turn the on the DUT and set the following controls:

| Instrument | Control | Position |
|---|---|---|
| Oscilloscope | Channel 1 | 5 V / Div, DC |
| | Channel 2 | 0.5 V / DIV, AC |
| | VERT MODE | ALT |
| | TRIGGER | CH 1 ONLY, DC, AUTO |
| | HORIZONTAL TIME BASE | 0.2 ms / DIV |
| Function Generator | Mode | Square Wave |
| | Amplitude | 4.75 +/- 0.25 V to GND |
| | Frequency | 500 kHz |
| BER Test Box | DSP OUTPUT LEVEL | Fully CW (IMPORTANT) |
| | SOURCE SELECT | AUDIO |
| | GAIN ADJUST | 1.5 Vpp on 'scope |
| DUT | Mode | 9600 Baud Data |
| | Band Selector | As Required |
| | Frequency | 146 or 440 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| | IF Filters | As Req for 9600 Baud Data |
| | Preamplifier | OFF |
| RF Generator | All | Set Automatically |

NOTE: Software is configured for use with the older IEEE card. However, this card has a conflict with an unknown device on the IEEE bus. Removal of all IEEE connections except the Marconi is necessary for this and the following test.

6.3.3.2 Receiver hiss should be heard; adjust volume to desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 6.3.3.3

6.3.3.3 Type "C" for Calibrate. The calibration process requires two steps and prompts appear describing each. The first is to set two pulses for simultaneous timing. Figure 2A depicts correct trace appearance for this step. The second sets a 14 us delay as depicted by Figure 2B.

6.3.3.4 A quick check can now be made. Manually set a –50 dBm signal. Type "4" for 10,000 ($10^4$) bits. If 0 or nearly 0 bit errors are detected, the system is functioning correctly. If 10,000 or nearly 10,000 errors are detected, then invert pulse. If approximately 50%, or 5,000 bit errors are detected, there is a problem with the system or test set-up. Recalibrate per step 6.3.3.3. Once 0 or nearly 0 errors are obtained , you may proceed to the next step.

6.3.3.5 Select menu option "A" for Automatic Test Sequence. (NOTE: Hook-up and calibration menus will reappear. Test will run automatically after this.)

6.3.3.6 One the automatic test sequence is completed, type "V" to save data to an appropriately named file that you select. Plots can be displayed by selecting "P" option.

6.3.3.7 Run a -50 dBm test for 1,000,000 bits. Hand record the results as part of the data.

**NOTE:** Although the automatic test sequence will generate considerable data, the published data typically only includes the BER at three test points. In order of ascending signal strength, these points are 12 dB SINAD, 16 dB SINAD and the hand recorded BER at -50 dBm. The first two data points are typically read from the data tables. **The minimum published BER is <1X10$^{-5}$.**

## 6.4 TRANSMITTER BIT ERROR RATE (BER) TEST
**No longer tested, but historical in nature**

6.4.1 The purpose of the Transmitter Bit Error Rate (BER) Test is to determine the usefulness of 9600 Baud transmissions when received at various receive signal levels. The signal will be adjusted for two levels before being applied to the BER Test Box, 12 dB SINAD and 12 dB SINAD plus 30 dB.

6.4.2 Test initialization and hook-up (see Fig 6.4-1).

NOTE: The IEEE Interface Bus, BER Test Box to Computer COM 1 port and BERT software must be correctly installed before proceeding to this step.

6.4.2.1 Turn on the BER Test box, Marconi Generator and Test Computer. While in DOS, go to the C:\SROOM subdirectory. Type "BERT" and press Enter to start the initializing process. After completion of this process, type the DUT radio and model number when prompted to do so. Next, type "T" for Transmitter Tests. The next prompt is for the test frequency. Use either 146 MHz or 440 MHz, whichever is appropriate for the band being tested. Type "I" for Initial Connections and Set-Up when the next prompt appears. Connect the following test hook-up as prompted to do so:

| Connection | Connectors | Cable Type |
|---|---|---|
| TX RF Output to Bird Power Attenuator Input | As Required to BNC | 50-Ohm coax |
| TX Data Input to BER Test Box DSP OUTPUT | As Required to BNC | 50-Ohm coax |

| | | |
|---|---|---|
| Bird Power Attenuator Output to Manual 10 dB Step Attenuator Input | BNC to BNC | 50-Ohm coax |
| Manual 10 dB Step Attenuator Output to HP 10 dB Step Attenuator Input | BNC to BNC | 50-Ohm coax |
| HP 10 dB Step Attenuator Output to HP 1 dB Step Attenuator Input | BNC to BNC | 50-Ohm coax |
| HP 1 dB Step Attenuator OUT to BER Test Box RF INPUT (“RF”) | BNC to BNC | 50-Ohm coax |
| BER Test Box LO to Marconi RF OUT | BNC to BNC | 50-Ohm coax |
| BER Test Box AUDIO OUT to Oscilloscope CHANNEL 2 INPUT | BNC to BNC | 50-Ohm coax |
| BER Test Box DSP Output to DUT TX Data input | RCA to As Required | 50-Ohm coax |
| BER Test Box SYNC OUT to Oscilloscope CHANNEL 1 INPUT | BNC to BNC | 50-Ohm coax |

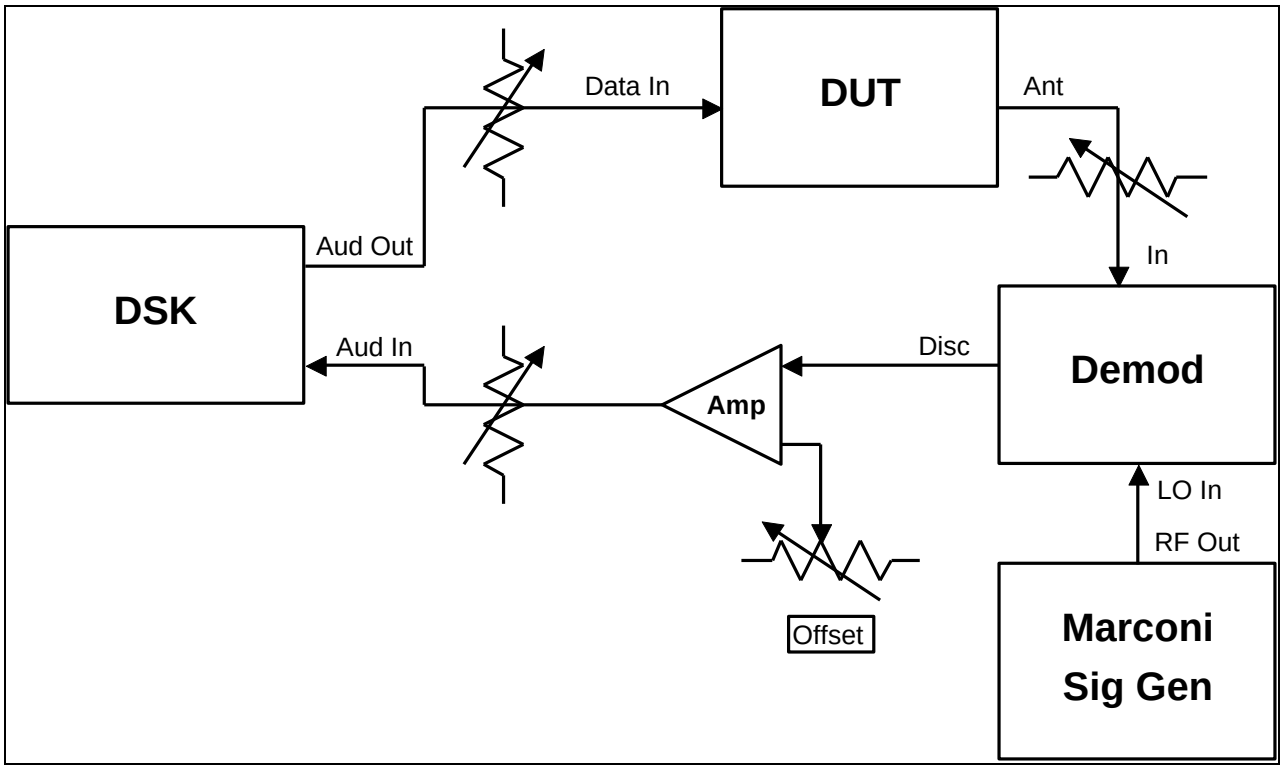



Fig. 6.4-1.

6.4.3 Test Procedure

6.4.3.1 Turn the on the DUT and set the following controls:

| Instrument | Control | Position |
|---|---|---|
| Oscilloscope | Channel 1 | 5 V / Div, DC |
| | Channel 2 | 0.5 V / DIV, AC |
| | VERT MODE | ALT |
| | TRIGGER | CH 1 ONLY, DC, AUTO |
| | HORIZONTAL TIME BASE | 0.2 ms / DIV |
| BER Test Box | DSP OUTPUT LEVEL | Fully CCW |
| | SOURCE SELECT | DEMOD |
| | GAIN ADJUST | Fully CCW |
| Manual 10 dB Step Attenuator | Attenuation | As required (see below) |
| DUT | Mode | 9600 Baud Data |
| | Band Selector | As Required |
| | Frequency | 146 or 440 MHz |
| | XMIT/RCV | RCV |
| | DRIVE or RF LEVEL | Minimum |
| RF Generator | All | Set Automatically |

6.4.3.2 Receiver hiss should be heard; adjust volume to desired level. Allow all equipment at least 10 minutes warm-up time before proceeding to step 6.4.3.3

6.4.3.3 Set DUT Transmitter for minimum power. Set the manual step attenuator for approximately 0 dBm (±10 dBm) into the HP 10 dB step attenuator.

6.4.3.4 Set the BER Test Box GAIN Control to produce 1.5 Vpp as determined by the lower channel 2 trace on the oscilloscope.

6.4.3.5 Use spacebar to toggle between reference signal and DUT signal. Adjust the BER Test Box DSP OUTPUT LEVEL so that both signals are same amplitude as viewed on oscilloscope. Hit any other key to exit.

6.4.3.6 Type "C" for Calibrate. The calibration process requires two steps and prompts appear describing each. The first is to set two pulses for simultaneous timing. Figure 2A depicts correct trace appearance for this step. The second sets a 14 μs delay as depicted by Figure 2B. Repeat the calibration process to verify the calibration.

Note: Automation of software for this test is not yet complete. Manual control and recording of all data is required. Set the software to "debug" mode (hit "Q" to back up a menu, then select Debug option). All program control is now done via the debug menus (see Table 1 for a list of commands). Hit the Esc key to exit debug mode when done.

6.4.3.7 Manually enable the Marconi carrier output and check SINAD level by typing "S". Set attenuators for 12 dB SINAD using < and > commands. Adjust noise level output with potentiometer control on BER test box as necessary.

6.4.3.8 A quick check of the test set-up can now be made if desired. Temporarily decrease the step attenuators by 30 or 40 dB. Type "4" for 10,000 ($10^4$) bits. If 0 or nearly 0 bit errors are detected, the system is functioning correctly. If 10,000 or nearly 10,000 errors are detected, then invert pulse. If approximately 50%, or 5,000 bit errors are detected, there is a problem with the system or test set-up. Recalibrate per step 6.4.3.6 and 6.4.3.7. Once 0 or nearly 0 errors are obtained, you may proceed to the next step.

6.4.3.9 Return the step attenuators for 12 dB SINAD setting. Run the test with $1X10^6$ bits by typing "6". Manually record the resulting Bit Error Rate.

6.4.3.10 Increase the signal by 30 dB (decrease attenuators 30 dB). Again run the test with $1X10^6$ bits by typing "6". Manually record the resulting Bit Error Rate.

**NOTE:** The published BER typically includes two data points, 12 dB SINAD and 12 dB SINAD + 30 dB. **The minimum published BER is $1X10^{-5}$.**

NOTE: If test frequency is 440 MHz or higher, you may wish to down convert to lower frequency. Use an LO frequency of 540 MHz for 100 MHz IF.

---

**TABLE 1**

Transmitter BER Tests
Use Debug mode of BERT program

**Commands**

**Esc** Exit Debug mode

| | | | | | |
|---|---|---|---|---|---|
| **<** | Decrease attenuation (Shift = 10 dB) | | **>** | Increase attenuation (Shift = 10 dB) | |
| **C** | Generate calibration signal | | **I** | Generate idle signal (second stage of calibration) | |
| **+** | Adjust coarse calibration | | **-** | Adjust coarse calibration | |
| **A** | Adjust fine calibration | | | | |
| **4** | BER $10^4$ | **5** | BER $10^5$ | **6** | BER $10^6$ |
| **S** | SINAD measurement | **N** | Sine-wave idle signal | **Q** | Quiet (unmodulated) idle |
| **1** | Data is inverted | **0** | Data is non-inverted | | |

## 6.6 Power Meter Test

6.6.1 The purpose of the power meter test is to measure power output developed by a RF transmitter by the DUT. This measurement directly compared to the HP 432B power meter and the table provided in Appendix XX.X. This table takes into account losses of attenuators and coax cables between the DUT and the HP power meter. Measurements will be recorded on Data Sheet X.XX. Also included are measurements of SWR compared against a known 1:1 and 2:1 load.

Two special RG-8 cables must be used. The dB loss of each cable has been previously measured on the frequencies used during this test. They are labeled with green tags as "Coax 1" and "Coax 2".

An RF linear amplifier must be used along with a transceiver capable of more than 100 watts (105 -110 watts) at the frequencies used must also be located. Inspect the data sheets of each DUT to determine if 6 meters or higher frequencies are needed for your choice of transceiver and linear amplifier. Also, use a linear amplifier that runs off of the 220 volt power receptacle in the screen room.

6.6.2 Test hook-up (See FIG. 6-1)

6.6.2.1 With all test equipment and DUT power switches in the off position and the transmitter in the receive mode, connect the following:

| Connection | Connectors | Cable Type |
|---|---|---|

| | | |
|---|---|---|
| HF/VHF transceiver to Linear Amplifier | PL-259 to PL-259 | 50 Ohm Coax |
| Linear Amplifier to DUT | PL-259 to PL-259 | 50 Ohm Coax |
| DUT to Bird 8329 30 dB Attenuator | PL-259 to Type N | “Coax 1” |
| Bird 8329 to HP 20 dB Attenuator | Type N to Type N | “Coax 2” |
| HP 20 dB Attenuator to Bird 10 dB Attenuator (for high power test) | Direct Connection | |
| HP or Bird Attenuator to HP 432B Power Meter | Direct Connection | |
| DC Power supply to DUT | As Required | |
| Two-Tone Generator to Microphone Jack | As Required | |
| Telegraph Key | As Required | |
| Keying Generator to Transceiver Key Jack | As Required | 50 Ohm Coax |

Keying connection from Transceiver to Linear Amplifier

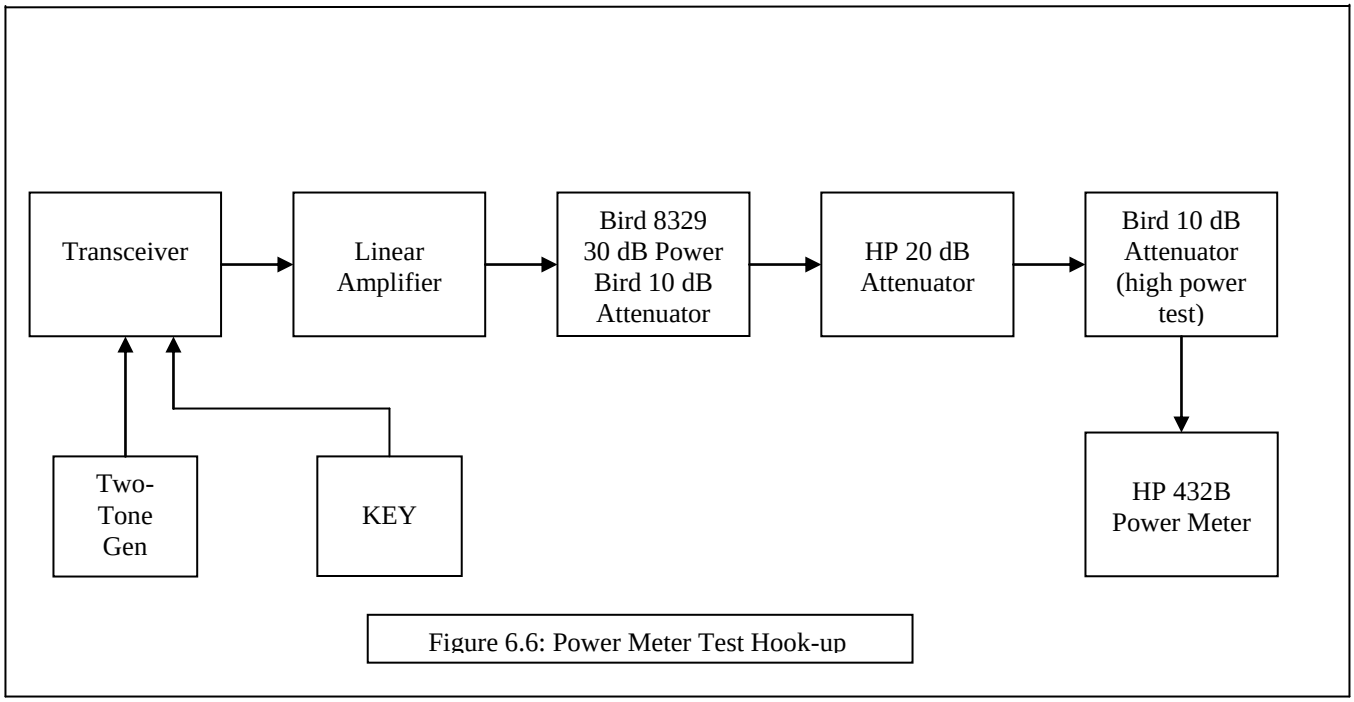



Figure 6.6: Power Meter Test Hook-up

6.6.3 Test Procedure

6.6.3.1 Turn on DUT, transceiver and power meter. Turn on DC supply for DUT. Leave the linear amplifier off until doing the high power tests.

6.6.3.2 Calibrate the HP 432B Power Meter. Make sure the frequency and calibration factor is correct for each individual test. Consult the power meter manual if needed.

| Instrument | Control | Position |
|---|---|---|
| Transceiver | Power Output | Lowest Setting |
| | Frequency | 1.995 MHz |
| | Antenna Port | As Required |
| | Mode | CW |
| | Mic Gain | Lowest Setting |
| Linear Amplifier | AC Power | Off |
| HP Power Meter | Frequency | .002 GHz |
| | Calibration Factor | See Table XX.X |
| | Zero | As Required |
| | dBm/mW | mW |
| DUT | RANGE | 200 or "Mid" |
| | AVE/PEP | AVE. |
| | DC Power | ON |
| | HOLD | Mid Scale |

6.6.3.3 Allow all equipment to warm up at least 10 minutes before proceeding to step 6.1.3.4

6.6.3.4 Tune the transceiver per the operator's manual. Put the transceiver in the CW transmit mode. Depress the telegraph key. Slowly turn up the power until a reading of XX.X mW is read on the power meter, equivalent to 100 watts at the DUT. If this reading cannot be attained, get to as close to XX.X as possible as possible. Note the difference, if any. Transmitting as long as necessary to get a steady reading on the HP power meter, at least 15 seconds. Let go of the telegraph key to turn off transmit mode.

6.6.3.5 Read DUT meter. Add or subtract power from this reading if an exact level could not be attained on the HP 432B power meter with step 6.1.3.4. Example:

HP 432B reads .925 mW but should read .930 mW. The DUT meter should be reading approximately 0.5 lower than it should be when applied 100 watts. Add a half of a watt for this example.
In reality, the difference should be multiplied by the percentage of power reaching the HP power meter, in this case, 0.5 watts X 0.930 = 0.465 watts. This may not make much difference on the 5 and 100 watt tests (especially with analog power meters), but does on the 1000 watt tests.

6.6.3.6 Remove telegraph key from the transceiver. Plug in Keying Generator. Adjust the keying generator for its highest speed. This gives the 50% duty cycle required. Turn generator on. Without adjusting the power output from the previous step, note the power reading on the DUT in the AVE and also in the PEP mode. (add any correction if necessary as in step 6.1.3.5). Record on Data Sheet.

6.6.3.7 Repeat the above steps for the 5 watt level tests. Set the DUT to the lowest power range. Please note the table for the HP432B at this level. Record results of the AVE and PEP and 50 % duty cycle power as in steps 6.1.3.4 through 6.1.3.6 in on Data Sheet.

6.6.3.8 Place transceiver on 14 MHz, Replace the keying generator with the telegraph key. Adjust power until the desired level is reached on the HP power meter for 100 watts at the DUT. Do not forget to adjust the HP power meter for the proper calibration factor and frequency. Zero before making all measurements. Record the AVE and PEP on the Data Sheet. Repeat for 5 watt measurements as in steps 6.1.3.4 through 3.1.3.6.

6.6.3.81 Repeat the above steps for 28, 50, 144, 222, & 432 MHz, if necessary.

6.6.3.9 Remove the telegraph key. Plug in the two-tone generator to the transceiver's microphone jack. Place the transceiver in the USB mode and turn on the VOX. Turn on the generator, both tones on. Turn up the mic gain on the transceiver and note the ALC. Adjust gain so the ALC is one quarter scale. Put the DUT to PEP. Note the power on the DUT and record it on the Data Sheet. Turn off tone generator.

****NOTE: Before the two-tone reading can be made, the two tones of the generator must be exactly even. Use the HP 8563E Spectrum Analyzer with the proper attenuators and step attenuators to match the tones evenly. Adjust the "Balance" control on the tone generator to match the tones on the analyzer. ****

6.6.3.10 Turn the AC power "on" on the Linear Amplifier. Place the amplifier in "standby" Allow to warm up at least 10 minutes. Set the Linear Amplifier and Transceiver for the 160 Meter band, the transceiver frequency adjusted to 1.995 MHz.

6.6.3.11 Place the 10 dB Bird Attenuator in series with the HP 20 dB attenuator. This is now a total of over 50 dB of attenuation for the input of the HP 432B power meter. Please note the set of tables for this extra attenuator for the 1000 watt measurements.

6.6.3.12 Set the transceiver for the lowest power output. Set the DUT to the highest power range. Place the telegraph key in the transceiver. Make sure the transceiver is in the CW mode. Place the Linear Amplifier in the "operate" position. Key the transmitter and slowly adjust the power output of the transceiver. Match the HP power meter to the table provided for 1000 watts at the DUT. Record the AVE and PEP power read by the DUT on the Data Sheet. Let go of the key ! Repeat the above steps for 50 % duty cycle, two-tone (14 MHz only) for 14 and 28 Mhz. (At this time, we do not have linear amplifiers for VHF and UHF testing.) Is it getting hot in there or what ?

Note: Keep an eye on the Linear Amplifier and Transceivers heat dissipation. Let them cool down if necessary.

## SWR TEST

For this test, only the transceiver is used. Make certain the Linear Amplifier AC power is "off". The test is preformed at the 100 watt level. Only the 30 dB Bird 8329 Attenuator is needed and used as a 1:1 load. Two 50 ohm precision carbon resistors in parallel will be used as a 2:1 load. "Coax 1" cable is used between the DUT and the 1:1 load. The 2:1 is connected directly to the DUT via its 6 inch long coax cable. Care must be used to keep the 2:1 load away from on top of or near conductive surfaces.

6.6.3.13 With the linear amplifier AC power off, set the power meter to "MID" or the 200 watt range. Connect the DUT antenna connector to the 1:1 load. Set the frequency of the transceiver to 1.995 Mhz. Set the power output of the transceiver to the lowest output setting. Place the transceiver in CW transmit mode and depress the key. Adjust the power output while watching the DUT. Adjust the transceiver for

100 watts on the DUT. Read the SWR and record on the Data Sheet. Release the key ! Repeat for 14, 28, 50 MHz, if necessary.

There are two kinds of SWR measuring meters, cross needle and single needle. Cross needle meters are read directly, single meters employ a “SET” knob and a forward/ reflected switch. If using a single needle SWR meter, rotate the set knob to full scale in the “forward” position while transmitting. Keep transmitting and switch to “reflected” to read the SWR.

6.6.3.14 Plug in the 2:1 load to the DUT antenna connector to the 2:1 load. Repeat step 6.6.3.13 and record on the data sheet.

**VII. DATA SHEETS/ PRODUCT REVIEW TEST MANUAL**

Manufacturer:_____ Model:_____ Serial #:_____

Date:_____ Test Engineer:_____

## 4.0 TRANSMITTER TESTS

### 4.1 OUTPUT-POWER TEST

**4.1.3.5** DC Power Supply Only/ Maximum Output Power:
Voltmeter Reading: _____V Ammeter Reading: _____A

**4.1.3.6** Receive Current:
Maximum volume; no signal; lights default: _____mA

**4.1.3.7 to 4.1.3.9; 4.1.3.11 to 4.1.3.13**

| Band | Mode | DUT Min | Bird Min | DUT Max | Bird Max | Spurs/ Notes |
|---|---|---|---|---|---|---|
| 160 | CW | _____ | _____ | _____ | _____ | _____ |
| 80 | CW | _____ | _____ | _____ | _____ | _____ |
| | AM | _____ | _____ | _____ | _____ | _____ |
| 60 | SSB | _____ | _____ | _____ | _____ | _____ |
| 40 | CW | _____ | _____ | _____ | _____ | _____ |
| 30 | CW | _____ | _____ | _____ | _____ | _____ |
| 20 | CW | _____ | _____ | _____ | _____ | _____ |
| | SSB | _____ | _____ | _____ | _____ | _____ |
| 17 | CW | _____ | _____ | _____ | _____ | _____ |
| 15 | CW | _____ | _____ | _____ | _____ | _____ |
| 12 | CW | _____ | _____ | _____ | _____ | _____ |
| 10 | CW | _____ | _____ | _____ | _____ | _____ |
| | FM | _____ | _____ | _____ | _____ | _____ |
| 6 | CW | _____ | _____ | _____ | _____ | _____ |
| | SSB | _____ | _____ | _____ | _____ | _____ |
| | FM | _____ | _____ | _____ | _____ | _____ |
| | AM | _____ | _____ | _____ | _____ | _____ |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | CW | _____ | _____ | _____ | _____ | _____ |
| | SSB | _____ | _____ | _____ | _____ | _____ |
| | FM | _____ | _____ | _____ | _____ | _____ |
| | AM | _____ | _____ | _____ | _____ | _____ |
| 440 | CW | _____ | _____ | _____ | _____ | _____ |
| | SSB | _____ | _____ | _____ | _____ | _____ |
| | FM | _____ | _____ | _____ | _____ | _____ |
| | AM | _____ | _____ | _____ | _____ | _____ |

**4.1.3.14** Transverter Output

| Band | µWattmeter Max |
|---|---|
| 20 | _____ |
| 15 | _____ |
| 10 | _____ |

## 4.2 TRANSMIT FREQUENCY RANGE TEST

| | Band | Low Frequency Limit | High Frequency Limit |
|---|---|---|---|
| **4.2.3.4/ 4.2.3.5** | 160 | _____MHz | _____MHz |
| **4.2.3.6** | 80 | _____MHz | _____MHz |
| | 60 | _____MHz | _____MHz |
| | 40 | _____MHz | _____MHz |
| | 30 | _____MHz | _____MHz |
| | 20 | _____MHz | _____MHz |
| | 17 | _____MHz | _____MHz |
| | 15 | _____MHz | _____MHz |
| | 12 | _____MHz | _____MHz |
| | 10 | _____MHz | _____MHz |
| | 6 | _____MHz | _____MHz |
| | 2 | _____MHz | _____MHz |
| | 440 | _____MHz | _____MHz |

All frequencies as measured on transmitter dial read-out.

## 4.3 CW TRANSMIT-FREQUENCY ACCURACY TEST

**Note:** See 6.1 Low-Voltage and Temperature Chamber Test for units designed for mobile and/or portable use.

**4.3.3.3** 20M Unkeyed DUT Frequency: _____MHz

**4.3.3.5/ 4.3.3.6** 20M Max Power Out/ Counter Display: _____MHz

**4.3.3.4/ 4.3.3.7** No significant display variation due to change in DC input Power (12V min) or Power Output ( ).

**4.3.3.8** 6M Unkeyed DUT Frequency: _____MHz
Max Power Out/ Counter Display: _____MHz

No significant display variation due to change in DC input power or Power Output ( ).

2M Unkeyed DUT Frequency: _____MHz
Max Power Out/ Counter Display: _____MHz

No significant display variation due to change in DC input power or Power Output ( ).

440MHz Unkeyed DUT Frequency: _____MHz
Max Power Out/ Counter Display: _____MHz

No significant display variation due to change in DC input power or Power Output ( ).

## 4.4 SPECTRAL PURITY TEST

| | | Spur @ | | Spur Check @ | |
|---|---|---|---|---|---|
| | Band | Max Power | Note | Min Power | Note |
| **4.4.3.9/** | 160 | _____dBc | _____ | _____dBc | _____ |
| **4.4.3.10** | | | | | |
| **4.4.3.11** | 80 | _____dBc | _____ | _____dBc | _____ |
| | 60 | _____dBc | _____ | _____dBc | _____ |
| | 40 | _____dBc | _____ | _____dBc | _____ |
| | 30 | _____dBc | _____ | _____dBc | _____ |
| | 20 | _____dBc | _____ | _____dBc | _____ |
| | 17 | _____dBc | _____ | _____dBc | _____ |
| | 15 | _____dBc | _____ | _____dBc | _____ |
| | 12 | _____dBc | _____ | _____dBc | _____ |
| | 10 | _____dBc | _____ | _____dBc | _____ |
| | 6 | _____dBc | _____ | _____dBc | _____ |
| | 2 | _____dBc | _____ | _____dBc | _____ |
| | 440 | _____dBc | _____ | _____dBc | _____ |

*NOTE: 5W minimum power.

**4.4.3.12** PLOT - WORST CASE/ SPECTRAL PURITY TAKEN ( )

DUT FREQUENCY _____MHz SPUR LEVEL _____dBc

**4.4.3.13** DUT Meets FCC Specifications: _____YES _____NO

**4.5 TWO-TONE TRANSMIT IMD TEST** 100W PEP ac Volt.

**4.5.3.7** 3.900 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____

**4.5.3.8** 14.250 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____

**4.5.3.9** 1.850 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____
5.330 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____
7.250 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____
10.120 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____
18.120 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____
21.250 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____
24.950 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____
28.350 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____
50.200 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____
144.200 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____
432.200 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / ) _____

Half-Power 14.250 MHz - TWO-TONE TRANSMIT IMD 3rd/5th ( / )

**4.6 SSB CARRIER AND UNWANTED SIDEBAND SUPPRESSION TEST**

**4.6.3.7** 20M Level Of Suppressed Carrier/USB Mode: _____dB Below PEP
20M Level Of Unwanted Sideband/USB Mode: _____dB Below PEP

**4.6.3.8** 20M Level Of Suppressed Carrier/LSB Mode: _____dB Below PEP
20M Level Of Unwanted Sideband/LSB Mode: _____dB Below PEP

**4.6.3.9** 6M Level Of Suppressed Carrier/USB Mode: _____dB Below PEP
6M Level Of Unwanted Sideband/USB Mode: _____dB Below PEP

6M Level Of Suppressed Carrier/LSB Mode: _____dB Below PEP
6M Level Of Unwanted Sideband/LSB Mode: _____dB Below PEP

2M Level Of Suppressed Carrier/USB Mode: _____dB Below PEP
2M Level Of Unwanted Sideband/USB Mode: _____dB Below PEP

2M Level Of Suppressed Carrier/LSB Mode: _____dB Below PEP
2M Level Of Unwanted Sideband/LSB Mode: _____dB Below PEP

440 Level Of Suppressed Carrier/USB Mode: _____dB Below PEP
440 Level Of Unwanted Sideband/USB Mode: _____dB Below PEP

440 Level Of Suppressed Carrier/LSB Mode: _____dB Below PEP
440 Level Of Unwanted Sideband/LSB Mode: _____dB Below PEP

**4.7 CW KEYING WAVEFORM TEST**

**4.7.3.6** CW KEYING / QSK OFF ( )
CW KEYING / QSK ON ( )
CW KEYING – QSK & HALF POWER - SPIKE? ( )

**4.7.3.7** Keying sidebands plots saved ( )

**4.8 PTT TO SSB/FM RF OUTPUT TEST**

**4.8.3.5**
20M SSB 50% Delay ON Time: _____ms

**4.8.3.6**
10M FM 50% Delay ON Time: _____ms

**4.8.3.7**
6M FM 50% Delay ON Time: _____ms
2M FM 50% Delay ON Time: _____ms
440 FM 50% Delay ON Time: _____ms

**4.9 TRANSMIT/ RECEIVE TURNAROUND TIME TEST**

**4.9.3.5** S9 Signal * /AGC Slow/PTT Key-Up To 50% Audio Output
Time: _____ms

**4.9.3.6** S9 Signal * /AGC Fast/PTT Key-Up To 50% Audio Output
Time: _____ms

S9 -20 dB /AGC Slow/PTT Key-Up To 50% Audio Output
Time: _____ms

No significant variation due to change in AGC or a 20 dB decrease in input signal ( ).

S9 Signal * /AGC Fast/CW Mode/Key-Up to 50% Audio Output
Time: _____ms

MDS + 10dB Signal/AGC Fast/CW Mode/Key-Up to 50% Audio Output
Time: _____ms

**4.9.3.7** DUT Suitable For AMTOR: _____YES _____NO

*NOTE: If an S meter is not available, use standard 50 μV input signal.

## 4.10 KEYER SPEED AND SIDETONE

**4.10.3.3** DOT to DASH ratio near 1:3 ( ).

**4.10.3.4** Default keyer speed dit to dit time: _____ ms
Code Speed (WPM): _____

**4.10.3.5** Minimum keyer speed dit to dit time: _____ ms
Code Speed (WPM): _____

**4.10.3.6** Maximum keyer speed dit to dit time: _____ ms
Code Speed (WPM): _____

**4.10.3.7** Default sidetone: _____ Hz
Minimum sidetone: _____ Hz
Maximum sidetone: _____ Hz

## 4.11 COMPOSITE NOISE TEST

**4.11.3.13** Plot taken - Composite Noise at 14.025 MHz ( )
Noise level at 2 kHz offset from carrier: _____ dBc/Hz
Noise level at 22 kHz offset from carrier: _____ dBc/Hz

**4.11.3.14** Plot taken - Composite Noise at 50.020 MHz ( )
Noise level at 2 kHz offset from carrier: _____ dBc/Hz
Noise level at 22 kHz offset from carrier: _____ dBc/Hz

Plot taken - Composite Noise at 144.020 MHz ( )
Noise level at 2 kHz offset from carrier: _____ dBc/Hz
Noise level at 22 kHz offset from carrier: _____ dBc/Hz

Plot taken - Composite Noise at 432.020 MHz ( )
Noise level at 2 kHz offset from carrier: _____ dBc/Hz
Noise level at 22 kHz offset from carrier: _____ dBc/Hz

## 5.0 RECEIVER TESTS

### 5.1 CW MINIMUM DISCERNIBLE SIGNAL (MDS) TEST

Standard (STD) Conditions: 10 dB Step Attn: 10 dB
1 dB Step Attn: 0 dB

IF FILTERS: _____Hz

**5.1.3.8/**
**5.1.3.9** Can AGC be set to OFF?: _____

1.020 MHz Preamp Off, MDS: _____dBm
Preamp One, MDS: _____dBm
Preamp Two, MDS: _____dBm

**5.1.3.10**

| | | | | | |
|---|---|---|---|---|---|
| 3.520 MHz | Preamp Off, MDS: | _____dBm | S5 Reference: | _____dBm |
| | Preamp One, MDS: | _____dBm | S5 Reference: | _____dBm |
| | Preamp Two, MDS: | _____dBm | S5 Reference: | _____dBm |

**5.1.3.11**

| | | | | | |
|---|---|---|---|---|---|
| 14.020 MHz | Preamp Off, MDS: | _____dBm | S5 Reference: | _____dBm |
| | Preamp One, MDS: | _____dBm | S5 Reference: | _____dBm |
| | Preamp Two, MDS: | _____dBm | S5 Reference: | _____dBm |

**5.1.3.12**

| | | | | | |
|---|---|---|---|---|---|
| 50.020 MHz | Preamp Off, MDS: | _____dBm | S5 Reference: | _____dBm |
| | Preamp One, MDS: | _____dBm | S5 Reference: | _____dBm |
| | Preamp Two, MDS: | _____dBm | S5 Reference: | _____dBm |
| 144.020 MHz | Preamp Off, MDS: | _____dBm | S5 Reference: | _____dBm |
| | Preamp One, MDS: | _____dBm | S5 Reference: | _____dBm |
| | Preamp Two, MDS: | _____dBm | S5 Reference: | _____dBm |
| 430.020 MHz | Preamp Off, MDS: | _____dBm | S5 Reference: | _____dBm |
| | Preamp One, MDS: | _____dBm | S5 Reference: | _____dBm |
| | Preamp Two, MDS: | _____dBm | S5 Reference: | _____dBm |

### 5.2 AM RECEIVE SENSITIVITY TEST

Standard (STD) Conditions: 10 dB Step Attn: 10 dB
1 dB Step Attn: 0 dB
IF FILTER _____kHz

**5.2.3.7/5.2.3.8**

1.020 MHz Preamp off, 10 dB (S+N)/N: _____μV
Preamp one, 10 dB (S+N)/N: _____μV
Preamp two, 10 dB (S+N)/N: _____μV

**5.2.3.9**

3.800 MHz Preamp off, 10 dB (S+N)/N: _____μV
Preamp one, 10 dB (S+N)/N: _____μV
Preamp two, 10 dB (S+N)/N: _____μV

**5.2.3.10**

53.000 MHz Preamp off, 10 dB (S+N)/N: _____µV
Preamp one, 10 dB (S+N)/N: _____µV
Preamp two, 10 dB (S+N)/N: _____µV

120.000 MHz Preamp off, 10 dB (S+N)/N: _____µV
(AIR) Preamp one, 10 dB (S+N)/N: _____µV
Preamp two, 10 dB (S+N)/N: _____µV

146.000 MHz Preamp off, 10 dB (S+N)/N: _____µV
Preamp one, 10 dB (S+N)/N: _____µV
Preamp two, 10 dB (S+N)/N: _____µV

440.000 MHz Preamp off, 10 dB (S+N)/N: _____µV
Preamp one, 10 dB (S+N)/N: _____µV
Preamp two, 10 dB (S+N)/N: _____µV

## 5.3 FM 12 dB SINAD TEST

Standard (STD) Conditions: 10 dB Step Attn: 10 dB
1 dB Step Attn: 0 dB
IF Filters: _____kHz (Normal/Narrow)

**5.3.3.5/5.3.3.6**

29.000 MHz Preamp off, 12 dB SINAD: _____µV _____dBm
(Normal/ Preamp one, 12 dB SINAD: _____µV _____dBm
Narrow) Preamp two, 12 dB SINAD: _____µV _____dBm

**5.3.3.13**

52.000 MHz Preamp off, 12 dB SINAD: _____µV _____dBm
Preamp one, 12 dB SINAD: _____µV _____dBm
Preamp two, 12 dB SINAD: _____µV _____dBm

100.000 MHz Preamp off, 12 dB SINAD: _____µV _____dBm
(Wide) Preamp one, 12 dB SINAD: _____µV _____dBm
Preamp one, 12 dB SINAD: _____µV _____dBm

146.000 MHz Preamp off, 12 dB SINAD: _____µV _____dBm
Preamp one, 12 dB SINAD: _____µV _____dBm
Preamp two, 12 dB SINAD: _____µV _____dBm

162.000 MHz Preamp off, 12 dB SINAD: _____µV _____dBm
Preamp one, 12 dB SINAD: _____µV _____dBm
Preamp two, 12 dB SINAD: _____µV _____dBm

| | | | |
|---|---|---|---|
| 223.500 MHz | Preamp off, 12 dB SINAD: _____μV | _____dBm |
| | Preamp one, 12 dB SINAD: _____μV | _____dBm |
| | Preamp two, 12 dB SINAD: _____μV | _____dBm |
| 440.000 MHz | Preamp off, 12 dB SINAD: _____μV | _____dBm |
| | Preamp one, 12 dB SINAD: _____μV | _____dBm |
| | Preamp two, 12 dB SINAD: _____μV | _____dBm |

## 5.4 RECEIVE FREQUENCY RANGE TEST

| | Band | Low Frequency Limit | High Frequency Limit |
|---|---|---|---|
| **5.4.3.4/ 5.4.3.5** | 160 | _____MHz | _____MHz |
| **5.4.3.6** | 80 | _____MHz | _____MHz |
| | 60 | _____MHz | _____MHz |
| | 40 | _____MHz | _____MHz |
| | 30 | _____MHz | _____MHz |
| | 20 | _____MHz | _____MHz |
| | 17 | _____MHz | _____MHz |
| | 15 | _____MHz | _____MHz |
| | 12 | _____MHz | _____MHz |
| | 10 | _____MHz | _____MHz |
| | 6 | _____MHz | _____MHz |
| | 2 | _____MHz | _____MHz |
| | 440 | _____MHz | _____MHz |
| | Other | _____MHz | _____MHz |
| | Other | _____MHz | _____MHz |
| | Other | _____MHz | _____MHz |
| | Other | _____MHz | _____MHz |
| | Other | _____MHz | _____MHz |
| | General Coverage | _____MHz | _____MHz |

All frequencies as measured on transmitter dial read-out.

## 5.5 FIRST IF AND IMAGE REJECTION TEST

Specified First Intermediate Frequency _____ MHz

| IF or Calculated Image Frequency | MDS Level in dBm | IF or Image Suppression in dB |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

## 5.7 BLOCKING GAIN COMPRESSION TEST

Standard (STD) Conditions: 10 dB Step Attn: 10 dB
1 dB Step Attn: 0 dB

IF FILTERS _____Hz

**5.7.3.5** Preamp off/ 1 dB Compression Point @ 3.520 MHz: _____dBm
Preamp one/ 1 dB Compression Point @ 3.520 MHz: _____dBm
Preamp two/ 1 dB Compression Point @ 3.520 MHz: _____dBm

**5.7.3.7**

3.520 MHz Block level, Preamp off, lower (3.500 MHz): _____dBm
Block level, Preamp off, lower (3.515 MHz): _____dBm
Block level, Preamp off, lower (3.518 MHz): _____dBm
Block level, Preamp off, upper (3.522 MHz): _____dBm
Block level, Preamp off, upper (3.525 MHz): _____dBm
Block level, Preamp off, upper (3.540 MHz): _____dBm
Block level, Preamp one, lower (3.500 MHz): _____dBm
Block level, Preamp one, upper (3.540 MHz): _____dBm
Block level, Preamp two, lower (3.500 MHz): _____dBm
Block level, Preamp two, upper (3.540 MHz): _____dBm

**5.7.3.9** Preamp off/ 1 dB Compression Point @ 14.020 MHz: _____dBm
Preamp one/ 1 dB Compression Point @ 14.020 MHz: _____dBm
Preamp two/ 1 dB Compression Point @ 14.020 MHz: _____dBm

14.020 MHz Block level, Preamp off, lower (14.000 MHz): _____dBm
Block level, Preamp off, lower (14.015 MHz): _____dBm
Block level, Preamp off, lower (14.018 MHz): _____dBm
Block level, Preamp off, upper (14.022 MHz): _____dBm
Block level, Preamp off, upper (14.025 MHz): _____dBm
Block level, Preamp off, upper (14.040 MHz): _____dBm
Block level, Preamp one, lower (14.000 MHz): _____dBm
Block level, Preamp one, upper (14.040 MHz): _____dBm
Block level, Preamp two, lower (14.000 MHz): _____dBm
Block level, Preamp two, upper (14.040 MHz): _____dBm

Preamp off/ 1 dB Compression Point @ 50.020 MHz: _____dBm
Preamp one/ 1 dB Compression Point @ 50.020 MHz: _____dBm
Preamp two/ 1 dB Compression Point @ 50.020 MHz: _____dBm

50.020 MHz Block level, Preamp off, lower (50.000 MHz): _____dBm
Block level, Preamp off, lower (50.015 MHz): _____dBm
Block level, Preamp off, lower (50.018 MHz): _____dBm
Block level, Preamp off, upper (50.022 MHz): _____dBm
Block level, Preamp off, upper (50.025 MHz): _____dBm
Block level, Preamp off, upper (50.040 MHz): _____dBm
Block level, Preamp one, lower (50.000 MHz): _____dBm
Block level, Preamp one, upper (50.040 MHz): _____dBm
Block level, Preamp two, lower (50.000 MHz): _____dBm
Block level, Preamp two, upper (50.040 MHz): _____dBm

Preamp off/ 1 dB Compression Point @ 144.020 MHz: _____dBm
Preamp one/ 1 dB Compression Point @ 144.020 MHz: _____dBm
Preamp two/ 1 dB Compression Point @ 144.020 MHz: _____dBm

144.020 MHz Block level, Preamp off, lower (144.000 MHz): _____dBm
Block level, Preamp off, lower (144.015 MHz): _____dBm
Block level, Preamp off, lower (144.018 MHz): _____dBm
Block level, Preamp off, upper (144.022 MHz): _____dBm
Block level, Preamp off, upper (144.025 MHz): _____dBm
Block level, Preamp off, upper (144.040 MHz): _____dBm
Block level, Preamp one, lower (144.000 MHz): _____dBm
Block level, Preamp one, upper (144.040 MHz): _____dBm
Block level, Preamp two, lower (144.000 MHz): _____dBm
Block level, Preamp two, upper (144.040 MHz): _____dBm

Preamp off/ 1 dB Compression Point @ 430.020 MHz: _____dBm
Preamp one/ 1 dB Compression Point @ 430.020 MHz: _____dBm
Preamp two/ 1 dB Compression Point @ 430.020 MHz: _____dBm

430.020 MHz Block level, Preamp off, lower (430.000 MHz): _____dBm
Block level, Preamp off, lower (430.015 MHz): _____dBm
Block level, Preamp off, lower (430.018 MHz): _____dBm
Block level, Preamp off, upper (430.022 MHz): _____dBm
Block level, Preamp off, upper (430.025 MHz): _____dBm
Block level, Preamp off, upper (430.040 MHz): _____dBm
Block level, Preamp one, lower (430.000 MHz): _____dBm
Block level, Preamp one, upper (430.040 MHz): _____dBm
Block level, Preamp two, lower (430.000 MHz): _____dBm
Block level, Preamp two, upper (430.040 MHz): _____dBm

**5.7A RECIPROCAL MIXING TEST**

+15 dBm Crystal Oscillator at 14.025 MHz

14.005 MHz, Preamp Off, Input level: _____dBm
14.020 MHz, Preamp Off, Input level: _____dBm
14.023 MHz, Preamp Off, Input level: _____dBm
14.027 MHz, Preamp Off, Input level: _____dBm
14.030 MHz, Preamp Off, Input level: _____dBm
14.040 MHz, Preamp Off, Input level: _____dBm
14.005 MHz, Preamp One, Input level: _____dBm
14.040 MHz, Preamp One, Input level: _____dBm
14.005 MHz, Preamp Two, Input level: _____dBm
14.040 MHz, Preamp Two, Input level: _____dBm

**5.8 TWO-TONE 3rd ORDER DYNAMIC RANGE TEST**

**5.8.3.6/5.8.3.7/5.8.3.8**
3.480/3.500 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
3.510/3.515 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
3.516/3.518 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
3.522/3.524 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
3.525/3.530 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
3.540/3.560 MHz Preamp Off, IMD level: _____dBm S5: _____dBm

3.480/3.500 MHz Preamp One, IMD level: _____dBm S5: _____dBm
3.540/3.560 MHz Preamp One, IMD level: _____dBm S5: _____dBm

3.480/3.500 MHz Preamp Two, IMD level: _____dBm S5: _____dBm
3.540/3.560 MHz Preamp Two, IMD level: _____dBm S5: _____dBm

**5.8.3.11**
13.980/14.000 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
14.010/14.015 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
14.016/14.018 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
14.022/14.024 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
14.025/14.030 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
14.040/14.060 MHz Preamp Off, IMD level: _____dBm S5: _____dBm

13.980/14.000 MHz Preamp One, IMD level: _____dBm S5: _____dBm
14.040/14.060 MHz Preamp One, IMD level: _____dBm S5: _____dBm

13.980/14.000 MHz Preamp Two, IMD level: _____dBm  S5: _____dBm
14.040/14.060 MHz Preamp Two, IMD level: _____dBm  S5: _____dBm

49.980/50.000 MHz Preamp Off, IMD level: _____dBm  S5: _____dBm
50.010/50.015 MHz Preamp Off, IMD level: _____dBm  S5: _____dBm
50.016/50.018 MHz Preamp Off, IMD level: _____dBm  S5: _____dBm
50.022/50.024 MHz Preamp Off, IMD level: _____dBm  S5: _____dBm
50.025/50.030 MHz Preamp Off, IMD level: _____dBm  S5: _____dBm
50.040/50.060 MHz Preamp Off, IMD level: _____dBm  S5: _____dBm

49.980/50.000 MHz Preamp One, IMD level: _____dBm  S5: _____dBm
50.040/50.060 MHz Preamp One, IMD level: _____dBm  S5: _____dBm

49.980/50.000 MHz Preamp Two, IMD level: _____dBm  S5: _____dBm
50.040/50.060 MHz Preamp Two, IMD level: _____dBm  S5: _____dBm

143.980/144.000 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
144.010/144.015 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
144.016/144.018 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
144.022/144.024 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
144.025/144.030 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
144.040/144.060 MHz Preamp Off, IMD level: _____dBm S5: _____dBm

143.980/144.000 MHz Preamp One, IMD level: _____dBm S5: _____dBm
144.040/144.060 MHz Preamp One, IMD level: _____dBm S5: _____dBm

143.980/144.000 MHz Preamp Two, IMD level: _____dBm S5: _____dBm
144.040/144.060 MHz Preamp Two, IMD level: _____dBm S5: _____dBm

429.980/430.000 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
430.010/430.015 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
430.016/430.018 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
430.022/430.024 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
430.025/430.030 MHz Preamp Off, IMD level: _____dBm S5: _____dBm
430.040/430.060 MHz Preamp Off, IMD level: _____dBm S5: _____dBm

429.980/430.000 MHz Preamp One, IMD level: _____dBm S5: _____dBm
430.040/430.060 MHz Preamp One, IMD level: _____dBm S5: _____dBm

429.980/430.000 MHz Preamp Two, IMD level: _____dBm S5: _____dBm
430.040/430.060 MHz Preamp Two, IMD level: _____dBm S5: _____dBm

**5.8.3.13 TWO-TONE 2nd ORDER DYNAMIC RANGE**

RF Generator frequencies: 6.000/8.020 MHz
Preamp off, Second-Order IMD Level: _____dBm (MDS)
Level: _____dBm (S5)
Intercept point: _____dBm

Preamp one, Second-Order IMD Level: _____dBm (MDS)
Level: _____dBm (S5)
Intercept point: _____dBm

Preamp two, Second-Order IMD Level: _____dBm (MDS)
Level: _____dBm (S5)
Intercept point: _____dBm

## 5.9 FM ADJACENT-CHANNEL SELECTIVITY TEST

NOTE: 6 dB SINAD IS EQUAL TO 50% DISTORTION.
DUT Preamp: On
IF FILTERS _____kHz

**5.9.3.4** 29.000 MHz/28.980 MHz
Adjacent Channel Level for 6 dB SINAD: _____dBm

**5.9.3.5** 29.000 MHz/29.020 MHz
Adjacent Channel Level for 6 dB SINAD: _____dBm

**5.9.3.6** 52.000 MHz/51.980 MHz
Adjacent Channel Level for 6 dB SINAD: _____dBm

52.000 MHz/52.020 MHz
Adjacent Channel Level for 6 dB SINAD: _____dBm

146.000 MHz/145.980 MHz
Adjacent Channel Level for 6 dB SINAD: _____dBm

146.000 MHz/146.020 MHz
Adjacent Channel Level for 6 dB SINAD: _____dBm

440.000 MHz/439.980 MHz
Adjacent Channel Level for 6 dB SINAD: _____dBm

440.000 MHz/440.020 MHz
Adjacent Channel Level for 6 dB SINAD: _____dBm

**5.9.3.8** Phase noise limit
Phase Noise Limit Level at 28.980 MHz: _____dBm
Phase Noise Limit Level at 29.020 MHz: _____dBm

**5.9.3.9**
Phase Noise Limit Level at 51.980 MHz: _____dBm
Phase Noise Limit Level at 52.020 MHz: _____dBm
Phase Noise Limit Level at 145.980 MHz: _____dBm
Phase Noise Limit Level at 146.020 MHz: _____dBm
Phase Noise Limit Level at 439.980 MHz: _____dBm
Phase Noise Limit Level at 440.020 MHz: _____dBm

**5.6.3.7** Adjacent Channel Rejection

| #1 Gen | #2 Gen | | #1 Gen | #2 Gen | |
|---|---|---|---|---|---|
| 29.000 | 28.980 | _____dB | 29.000 | 29.020 | _____dB |
| 52.000 | 51.980 | _____dB | 52.000 | 52.020 | _____dB |
| 146.000 | 145.980 | _____dB | 146.000 | 146.020 | _____dB |
| 440.000 | 439.980 | _____dB | 440.000 | 440.020 | _____dB |

## 5.10 FM TWO-TONE 3rd ORDER DYNAMIC RANGE TEST

NOTE: 25% DISTORTION IS EQUAL TO 12 dB SINAD
Standard (STD) Conditions: RF Gen #1 and #2 Output
Level: -17 dBm
DUT Preamp: On
IF FILTERS _____kHz

**5.10.3.5**
Generator frequencies: 28.960/28.980 MHz
Signal Level for 25% Distortion @ 29.000 MHz: _____dBm

**5.10.3.6**
Generator frequencies: 29.020/29.040 MHz
Signal Level for 25% Distortion @ 29.000 MHz: _____dBm

**5.10.3.8**
Generator frequencies: 51.960/51.980 MHz
Signal Level for 25% Distortion @ 52.000 MHz: _____dBm
Generator frequencies: 52.020/52.040 MHz
Signal Level for 25% Distortion @ 52.000 MHz: _____dBm
Generator frequencies: 62.000/72.000 MHz
Signal Level for 25% Distortion @ 52.000 MHz: _____dBm

Generator frequencies: 145.960/145.980 MHz
Signal Level for 25% Distortion @ 146.000 MHz: _____dBm
Generator frequencies: 146.020/146.040 MHz
Signal Level for 25% Distortion @ 146.000 MHz: _____dBm
Generator frequencies: 156.000/166.000 MHz
Signal Level for 25% Distortion @ 146.000 MHz: _____dBm

Generator frequencies: 439.960/439.980 MHz
Signal Level for 25% Distortion @ 440.000 MHz: _____dBm
Generator frequencies: 440.020/440.040 MHz
Signal Level for 25% Distortion @ 440.000 MHz: _____dBm
Generator frequencies: 450.000/460.000 MHz
Signal Level for 25% Distortion @ 440.000 MHz: _____dBm

## 5.11 AUDIO POWER OUTPUT TEST

**5.11.3.6** Specified Distortion: _____% THD

Specified Load Impedance: _____Ohms

**5.11.3.7** Audio Voltage: _____V

**5.11.3.8** Audio Output Power: _____W

**5.11.3.9** Audio voltage at minimum volume (no signal): _____mV (Hiss level)

## 5.12 AUDIO AND IF FREQUENCY RESPONSE TEST

**5.12.3.6** IF Filters: _____Hz (CW Narrow)

Low Frequency: _____Hz

**5.12.3.7** High Frequency: _____Hz Difference: _____Hz

**5.12.3.8** IF Filters: _____Hz (USB Wide)

Low Frequency: _____Hz

High Frequency: _____Hz Difference: _____Hz

IF Filters: _____kHz (LSB Wide)

Low Frequency: _____Hz

High Frequency: _____Hz Difference: _____Hz

**5.12.3.10** IF Filters: _____kHz (AM)

Low Frequency: _____Hz

High Frequency: _____Hz Difference: _____Hz

## 5.13 SQUELCH SENSITIVITY TEST

Standard (STD) Conditions: 10 dB Step Attn: 10 dB
1 dB Step Attn: 0 dB
Preamp On
IF FILTERS _____Hz

**5.13.3.5**
FM
29.000 MHz Squelch Threshold Point: _____μV

**5.13.3.6**
52.000 MHz Squelch Threshold Point: _____μV

146.000 MHz Squelch Threshold Point: _____μV

440.000 MHz Squelch Threshold Point: _____μV

**5.13.3.7**
SSB
14.200 MHz Squelch Threshold Point: _____μV

## 5.14 S METER TEST

Standard (STD) Conditions: 10 dB Step Attn: 10 dB
1 dB Step Attn: 0 dB
IF FILTERS _____Hz

**5.14.3.4/5.14.3.5**
14.200 MHz Preamp Off, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

**5.14.3.6**
14.200 MHz Preamp One, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

14.200 MHz Preamp Two, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

**5.14.3.7**

1.020 MHz Preamp Off, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

1.020 MHz Preamp One, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

1.020 MHz Preamp Two, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

50.020 MHz Preamp Off, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

50.020 MHz Preamp One, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

50.020 MHz Preamp Two, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

144.020 MHz Preamp Off, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

144.020 MHz Preamp One, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

144.020 MHz Preamp Two, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

420.020 MHz Preamp Off, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

420.020 MHz Preamp One, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

420.020 MHz Preamp Two, S9 Indication: _____dBm _____μV
No significant deviation from expected level
at S9+20 dB( ).

## 5.16 NOTCH FILTER TEST

**5.16.3.6** Minimum Notch Depth: _____dB

**5.16.3.11** Maximum Notch Depth: _____dB

**5.16.3.17** Notch Attack Time: _____ms

Test options -
2 tones (S9 each), AGC on, manual notch
2 tones (S9 + S1), AGC on, manual notch
1 tone, AGC on, manual notch
1 tone, AGC off, manual notch
1 tone, AGC on, auto notch
1 tone, AGC off, auto notch

Note changes in tone level and noise level.
Also note changes in S-meter for AGC on tests.

## 5.17 DSP NOISE REDUCTION TEST

**6.2.3.5** S9 Signal, best case noise reduction: _____dB

**6.2.3.6** Overall best case noise reduction: _____dB

## 6.0 NON-STANDARD & SPECIAL PURPOSE TESTS

## 6.1 LOW VOLTAGE AND TEMPERATURE CHAMBER TEST

**6.1.3.4** Room Temperature, 13.8V dc
Power Output (Bird): _____W
Frequency: _____ MHz

**6.1.3.5** Room Temperature, 13.8V dc
Minimum Discernable Signal: _____dBm

**6.1.3.5** Room Temperature, Low Voltage (11.5V nominal)
Power Output (Bird): _____W
Frequency: _____ MHz
Minimum Discernable Signal: _____dBm

**6.1.3.6** Low Temperature (-10 deg. C nominal), 13.8V dc
Power Output (Bird): _____W
Frequency: _____ MHz
Minimum Discernable Signal: _____dBm

**6.1.3.7** High Temperature (+60 deg. C nominal), 13.8V dc
Power Output (Bird): _____W
Frequency: _____ MHz
Minimum Discernable Signal: _____dBm

**6.1.3.8** Room Temperature, 13.8V dc
Power Output (Bird): _____W
Frequency: _____ MHz
Minimum Discernable Signal: _____dBm
DUT Functions Normally ( ).

## 6.3 RECEIVER BIT ERROR RATE TEST

**6.3.3.7** 146 MHz, BER for 12-dB SINAD: _____
146 MHz, BER for 16-dB SINAD: _____
146 MHz, BER for -50 dBm: _____

440 MHz, BER for 12-dB SINAD: _____
440 MHz, BER for 16-dB SINAD: _____
440 MHz, BER for -50 dBm: _____

## 6.4 TRANSMITTER BIT ERROR RATE TEST

**6.4.3.9** 146 MHz, BER for 12 dB SINAD: _____
146 MHz, BER for SINAD + 30 dB: _____

440 MHz, BER for 12 dB SINAD: _____
440 MHz, BER for SINAD + 30 dB: _____

## 6.5 RECEIVER PHASE NOISE TEST

**6.5.3.4** 1 kHz offset, crystal oscillator level

**6.5.3.5**

14.026 MHz, oscillator level: _____dBm
14.027 MHz, oscillator level: _____dBm
14.028 MHz, oscillator level: _____dBm
14.029 MHz, oscillator level: _____dBm
14.030 MHz, oscillator level: _____dBm
14.032 MHz, oscillator level: _____dBm
14.034 MHz, oscillator level: _____dBm
14.036 MHz, oscillator level: _____dBm
14.038 MHz, oscillator level: _____dBm
14.040 MHz, oscillator level: _____dBm
14.042 MHz, oscillator level: _____dBm
14.044 MHz, oscillator level: _____dBm
14.046 MHz, oscillator level: _____dBm
14.048 MHz, oscillator level: _____dBm
14.050 MHz, oscillator level: _____dBm
14.055 MHz, oscillator level: _____dBm
14.060 MHz, oscillator level: _____dBm
14.065 MHz, oscillator level: _____dBm
14.070 MHz, oscillator level: _____dBm
14.075 MHz, oscillator level: _____dBm
14.080 MHz, oscillator level: _____dBm
14.085 MHz, oscillator level: _____dBm
14.090 MHz, oscillator level: _____dBm
14.105 MHz, oscillator level: _____dBm

14.115 MHz, oscillator level: _____dBm
14.125 MHz, oscillator level: _____dBm
14.135 MHz, oscillator level: _____dBm
14.145 MHz, oscillator level: _____dBm
14.155 MHz, oscillator level: _____dBm
14.165 MHz, oscillator level: _____dBm
14.175 MHz, oscillator level: _____dBm
14.200 MHz, oscillator level: _____dBm
14.225 MHz, oscillator level: _____dBm
14.250 MHz, oscillator level: _____dBm
14.275 MHz, oscillator level: _____dBm
14.325 MHz, oscillator level: _____dBm
14.375 MHz, oscillator level: _____dBm
14.425 MHz, oscillator level: _____dBm
14.525 MHz, oscillator level: _____dBm

### 5.1.3.13 Expanded Tests

```
1.820 MHz  Preamp Off, MDS: __________dBm
           Preamp One, MDS: __________dBm
           Preamp Two, MDS: __________dBm

7.020 MHz  Preamp Off, MDS: __________dBm
           Preamp One, MDS: __________dBm
           Preamp Two, MDS: __________dBm

10.120 MHz Preamp Off, MDS: __________dBm
           Preamp One, MDS: __________dBm
           Preamp Two, MDS: __________dBm

18.088 MHz Preamp Off, MDS: __________dBm
           Preamp One, MDS: __________dBm
           Preamp Two, MDS: __________dBm

21.020 MHz Preamp Off, MDS: __________dBm
           Preamp One, MDS: __________dBm
           Preamp Two, MDS: __________dBm

24.910 MHz Preamp Off, MDS: __________dBm
           Preamp One, MDS: __________dBm
           Preamp Two, MDS: __________dBm

28.020 MHz Preamp Off, MDS: __________dBm
           Preamp One, MDS: __________dBm
           Preamp Two, MDS: __________dBm
```

### 5.7.3.11 Swept Blocking Dynamic Range

| Lower Frequencies | Block Level (dBm) | Upper Frequencies | Block Level (dBm) |
|---|---|---|---|
| 13.820 MHz | | 14.220 MHz | |
| 13.840 MHz | | 14.200 MHz | |
| 13.860 MHz | | 14.180 MHz | |
| 13.880 MHz | | 14.160 MHz | |
| 13.900 MHz | | 14.140 MHz | |
| 13.920 MHz | | 14.120 MHz | |
| 13.940 MHz | | 14.100 MHz | |
| 13.960 MHz | | 14.080 MHz | |
| 13.965 MHz | | 14.075 MHz | |
| 13.970 MHz | | 14.070 MHz | |
| 13.975 MHz | | 14.065 MHz | |
| 13.980 MHz | | 14.060 MHz | |
| 13.985 MHz | | 14.055 MHz | |

| | | | |
|---|---|---|---|
| 13.990 MHz | | 14.050 MHz | |
| 13.992 MHz | | 14.048 MHz | |
| 13.994 MHz | | 14.046 MHz | |
| 13.996 MHz | | 14.044 MHz | |
| 13.998 MHz | | 14.042 MHz | |
| 14.000 MHz | | 14.040 MHz | |
| 14.002 MHz | | 14.038 MHz | |
| 14.004 MHz | | 14.036 MHz | |
| 14.006 MHz | | 14.034 MHz | |
| 14.008 MHz | | 14.032 MHz | |
| 14.010 MHz | | 14.030 MHz | |
| 14.012 MHz | | 14.028 MHz | |
| 14.014 MHz | | 14.026 MHz | |
| 14.015 MHz | | 14.025 MHz | |
| 14.016 MHz | | 14.024 MHz | |
| 14.017 MHz | | 14.023 MHz | |
| 14.018 MHz | | 14.022 MHz | |
| 14.019 MHz | | 14.021 MHz | |

**5.8.3.18 Swept IMD Dynamic Range** (Cont.)

| First Generator Frequency | Second Generator Frequency | Level for MDS IMD Response | First Generator Frequency | Second Generator Frequency | Level for MDS IMD Response |
|---|---|---|---|---|---|
| 13.620 MHz | 13.820 MHz | | 14.022 MHz | 14.021 MHz | |
| 13.660 MHz | 13.840 MHz | | 14.024 MHz | 14.022 MHz | |
| 13.700 MHz | 13.860 MHz | | 14.026 MHz | 14.023 MHz | |
| 13.740 MHz | 13.880 MHz | | 14.028 MHz | 14.024 MHz | |
| 13.780 MHz | 13.900 MHz | | 14.030 MHz | 14.025 MHz | |
| 13.820 MHz | 13.920 MHz | | 14.032 MHz | 14.026 MHz | |
| 13.860 MHz | 13.940 MHz | | 14.036 MHz | 14.028 MHz | |
| 13.900 MHz | 13.960 MHz | | 14.040 MHz | 14.030 MHz | |
| 13.910 MHz | 13.965 MHz | | 14.044 MHz | 14.032 MHz | |
| 13.920 MHz | 13.970 MHz | | 14.048 MHz | 14.034 MHz | |
| 13.930 MHz | 13.975 MHz | | 14.052 MHz | 14.036 MHz | |
| 13.940 MHz | 13.980 MHz | | 14.056 MHz | 14.038 MHz | |
| 13.950 MHz | 13.985 MHz | | 14.060 MHz | 14.040 MHz | |
| 13.960 MHz | 13.990 MHz | | 14.064 MHz | 14.042 MHz | |
| 13.964 MHz | 13.992 MHz | | 14.068 MHz | 14.044 MHz | |
| 13.968 MHz | 13.994 MHz | | 14.072 MHz | 14.046 MHz | |
| 13.972 MHz | 13.996 MHz | | 14.076 MHz | 14.048 MHz | |
| 13.976 MHz | 13.998 MHz | | 14.080 MHz | 14.050 MHz | |
| 13.980 MHz | 14.000 MHz | | 14.090 MHz | 14.055 MHz | |
| 13.984 MHz | 14.002 MHz | | 14.100 MHz | 14.060 MHz | |
| 13.988 MHz | 14.004 MHz | | 14.110 MHz | 14.065 MHz | |
| 13.992 MHz | 14.006 MHz | | 14.120 MHz | 14.070 MHz | |
| 13.996 MHz | 14.008 MHz | | 14.130 MHz | 14.075 MHz | |
| 14.000 MHz | 14.010 MHz | | 14.140 MHz | 14.080 MHz | |
| 14.004 MHz | 14.012 MHz | | 14.180 MHz | 14.100 MHz | |
| 14.008 MHz | 14.014 MHz | | 14.220 MHz | 14.120 MHz | |
| 14.010 MHz | 14.015 MHz | | 14.260 MHz | 14.140 MHz | |
| 14.012 MHz | 14.016 MHz | | 14.300 MHz | 14.160 MHz | |
| 14.014 MHz | 14.017 MHz | | 14.340 MHz | 14.180 MHz | |
| 14.016 MHz | 14.018 MHz | | 14.380 MHz | 14.200 MHz | |
| 14.018 MHz | 14.019 MHz | | 14.420 MHz | 14.220 MHz | |

## 5.15 IN-BAND IMD TEST

**5.15.3.6** 14.200 MHz, AGC Fast, S9 Signals, plot taken ( ).
14.200 MHz, AGC Fast, S9 Signals, plot taken ( ).
14.200 MHz, AGC Fast, S9+40/60 Signals, plot taken ( ).
14.200 MHz, AGC Fast, S9+40/60 Signals, plot taken ( ).

# 6.7 HF LINEAR AMPLIFIER TEST

6.7.1 The purpose of the Linear Amplifier Test is to determine if the amplifier meets the manufacturer's power output specification, meets FCC Part 97 regarding Spurious Emissions and measure and record the level of Intermodulation Distortion.

It is very important for the test engineer to be familiar with the Lab test equipment and the Device Under Test (DUT). The manufacturer's manual should be completely read and understood before any testing is performed. At no time should any equipment be operated in a manner that is inconsistent with the manufacturer's recommended procedures or published limits. Failure to understand the DUT could result in test error, damage to the equipment and worse, damage to the test engineer.

6.7.2 List of Equipment:

| Instrument | Manufacturer | Model | Quantity |
|---|---|---|---|
| Spectrum Analyzer | HP | 8563E | 1 |
| In-Line RF Wattmeter | Bird | 4381 | 1 |
| In-Line RF Wattmeter | Bird | 43 | 2 |
| Power Attenuator | Bird | 8329 | 1 |
| RF Step Attenuator | HP | 355C | 1 |
| RF Step Attenuator | HP | 355D | 1 |
| Fixed RF Attenuator | Bird | 8340-100 | 1 |
| Fixed RF Attenuator | Bird | 8340-200 | 1 |
| Signal Generator | Any | Any | 2 |
| Driver Amplifier | ARRL | N/A | 2 |
| 150W Power Amplifier | CCI | EB63 | 2 |
| 14 MHz Power Combiner | ARRL | NA | 1 |
| Inter-Connecting Cables | NA | As Required | As Required |

**6.7A LINEAR AMPLIFIER OPERATION TEST**

6.7A.1 The purpose of the Linear Amplifier Operation Test is to determine the ability of the amplifier to produce its full rated output power in each applicable Amateur Band without exceeding any of the manufacturer's specified limits. The input drive power, the input reflected power, the output forward and reflected power and the spurious emissions are measured at each test point. Measurements are made at the edges of the bands and one or more mid-band points. The spectral display on each band is saved to disc. Additionally, the Linear Amplifier will be tested with an input signal of 27 MHz to ensure the amplifier does not function on the Citizen's Band.

6.7A.2 Test Hook-Up (See FIGURE 6.7A)

Diagram here:

6.7A.3 With all power switches in the OFF position and the signal generator outputs disabled, connect the following:

| CONNECTION | CONNECTORS | CABLE TYPE |
|---|---|---|
| Signal Generator Out to Driver #1 In | Type N to BNC | RG-58 |
| Driver #1 Out to Linear Amp #1 In | BNC to BNC | RG-58 |
| Linear Amp #1 Out to | BNC to BNC | RG-58 |

| | | |
|---|---|---|
| Filter #1 In | | |
| Filter #1 Out to Bird 43 Wattmeter In | BNC to PL-259 | RG-58 |
| Bird 43 Wattmeter Out to DUT In | PL-259 to As Required | RG-58 |
| DUT Out to Bird 4381 Wattmeter In | As required to type N | RG-8 |
| Bird 4381 Wattmeter Out to Power Attenuator In | Type N to type N | RG-8 |
| Power Attenuator Out to 20 dB Attenuator In | Type N to type N | RG-58 |
| 20 dB Attenuator Out to 10 dB Step Attenuator In | Type N to BNC | RG-58 |
| 10 dB Step Attenuator Out to 1 dB Step Attenuator In | BNC to BNC | RG-58 |
| 1 dB Step Attenuator Out to Spectrum Analyzer In | BNC to BNC | RG-58 |

6.7A.4 Set the following controls in the order shown

| INSTRUMENT | CONTROL | POSITION |
|---|---|---|
| DUT | Operate/Standby | Standby |
| DUT | Power On/Off | On |
| DUT | TUNE | Pre-set for 160 m |
| DUT | Load | Pre-set for 160 m |
| DUT | CW/SSB | CW |
| Bird 4381 | Power On/Off | ON |
| Bird 4381 | Function | FWD CW |
| Bird 4381 | Forward Element | As Required |
| Bird 43 | Element | As Required |
| Bird 43 | Element Direction | Forward |
| 10 dB Step Attn | Attenuator | >20 dB |
| 1 dB Step Attn | Attenuator | Any |
| Signal Generator | Modulation | Off |
| Signal Generator | Frequency | 1.800 MHz |
| Signal Generator | Carrier Off/On | Off |
| Signal Generator | RF Level | < -40dBm |
| **Spectrum Analyzer | Reference Level | -10dBm |
| Spectrum Analyzer | Input Attenuation | 20 dB |
| Spectrum Analyzer | Threshold | -80 dBm |
| Spectrum Analyzer | Resolution Bandwidth | 10 kHz |
| Spectrum Analyzer | Video Bandwidth | 30 kHz |
| Spectrum Analyzer | Range MHz | 0 to 50 MHz |
| Spectrum Analyzer | Sweep Time | Auto |

**** NOTE:** The spectrum analyzer is automatically set to these values when using the LabWindows/CVI software.

6.7A.5 Allow the equipment to warm up for at least 10 minutes before proceeding to the next step.

6.7A.6 Check to make sure all power switches are on, including the filament switch if the DUT has one.

6.7A.7 If the DUT is if the vacuum tube type, the tuning of the amplifier must be performed in a manner consistent with all DUT and vacuum tube limits. Be sure not to exceed the maximum plate current, grid current and DUT power output limits while performing the following steps. If the amplifier is of the solid state type, be sure not to exceed the final PA maximum operating temperature and duty cycle limit.

NOTE: Some of the tuning steps may be omitted when the DUT is of the solid state type. Usually, there are no tuning adjustments, such as Plate and Load. Follow the procedure for adjusting the amplifier's drive level for the rated output.

6.7A.8 Switch the DUT to the Operate mode and enable the carrier on the generator. Increase the generator level until a moderate amount of RF is produced but the DUT (consult the operator's manual for the recommended drive level for tune-up). Tune the DUT per the manufacturer's instructions, for the maximum power output without exceeding the maximum rated power output.
Be sure to adjust the generator as necessary if the rated output is exceeded.

6.7A.9 Measure and record on the data sheet the following items:

| MEASUREMENT | INSTRUMENT |
|---|---|
| Forward Power | Bird 43 |
| Reflected Power | Bird 43 |
| Plate Voltage | DUT Meter |
| Plate Current | DUT Meter |
| Grid Current | DUT Meter |
| Output Power | Bird 4381 |

6.7A.9.1 Adjust the step attenuators to provide approximately -10 dBm input to the spectrum analyzer (the fundamental should be just at the top of the frequency sweep display).

**NOTE:** *At no time should the input to the spectrum analyzer exceed 0 dBm. Damage to the instrument will occur at +10 dBm or greater !*

6.7A.9.2 If using the LabWindows/CVI software, click on the READ ANALYZER button to acquire data from the analyzer and display it on the screen. The level of the harmonics and/or spurs, in dB below the fundamental, can be read on the analyzer using the delta function. Note worst case harmonic and spur and record it on the data sheet. Disable the carrier on the signal generator.

6.7A.9.3 Set the generator frequency to the next value on the data sheet and re-tune the DUT as necessary. Repeat step 6.7A.9 for this test frequency. Again, repeat this step incrementing the generator's frequency each time, until the upper limit of the Amateur Band is reached.

6.7A.9.4 Repeat the above steps for each of the available Amateur Bands for each frequency on the data sheet. For 10 to 20 MHz, set the spectrum analyzer for a frequency range of 0 to 100 MHz. For 20 to 30 MHz, set the spectrum analyzer for a frequency range of 0 to 200 MHz.

**6.7B LINEAR AMPLIFIER IMD TEST**

6.7A.1 The purpose of the Linear Amplifier IMD Test is to measure and record the IMD products generated within the DUT when subjected to two simultaneous input signals. This test is performed by generating two RF signals (at 14.020 and 14.0212 MHz), amplifying them to an appropriate drive level, combining the amplified signals and feeding them to the DUT input. The DUT output IMD products are then displayed on the spectrum analyzer.

6.7B.2 Test Hook-Up (See Figure 6.7B)

Diagram here:

6.7B.3 With all power switches in the off position and both signal generator carriers disabled, connect the following:

| CONNECTION | CONNECTORS | CABLE TYPE |
|---|---|---|
| Signal Generator #1 Out to Driver #1 In | Type N to BNC | RG-58 |
| Signal Generator #2 Out to Driver #2 In | Type N to BNC | RG-58 |
| Driver #1 Out to Linear Amp #1 In | BNC to BNC | RG-58 |
| Driver #2 Out to Linear Amp #2 In | BNC to BNC | RG-58 |
| Linear Amp #1 Out to Filter #1 In | BNC to BNC | RG-58 |
| Linear Amp #2 Out to Filter #2 In | BNC to BNC | RG-58 |
| Filter #1 Out to Bird 43 #1 In | BNC to PL-259 | RG-58 |
| Filter #2 Out to Bird 43 #2 In | BNC to PL-259 | RG-58 |
| Bird 43 #1 Out to Power Combiner #1 In | PL-259 to As Required | RG-58 |
| Bird 43 #1 Out to Power Combiner #2 In | PL-259 to As Required | RG-58 |
| Combiner 50 Ohm Out to 100 Watt Dummy Load | BNC to SO-239 | RG-58 |
| Combiner RF Out to DUT RF In | BNC to As Required | RG-58 |
| DUT RF Out to Bird 4381 In | As Required to Type N | RG-8 |
| Bird 4381 Out to Power Attenuator In | Type N to Type N | RG-8 |
| Power Attenuator Out to 20 dB Attenuator In | Type N to Type N | RG-58 |
| 20 dB Attenuator Out to 10 dB Step Attn In | Type N to BNC | RG-58 |
| 10 dB Step Attn Out to 1 dB Step Attn In | BNC to BNC | RG-58 |
| 1 dB Step Attn Out to Spectrum Analyzer In | BNC to Type N | RG-58 |

6.7B.4 Set the following controls in the order shown

| INSTRUMENT | CONTROL | POSITION |
|---|---|---|
| DUT | Operate/Standby | Standby |
| DUT | Power On/Off | On |
| DUT | TUNE | Pre-set for 20 m |
| DUT | LOAD | Pre-set for 20 m |
| DUT | CW/SSB | CW |
| Bird 4381 | Power On/Off | On |
| Bird 4381 | Function | FWD CW |
| Bird 4381 | Forward Element | As Required |
| Bird 43 | Element | As required |
| Bird 43 | Element Direction | Forward |
| 10 dB Step Attn | Attenuator | > 40 dB |
| 1 dB Step Attn | Attenuator | Any |
| Generator #1 | Modulation | Off |
| Generator #1 | Frequency | 14.020 MHz |
| Generator #1 | Carrier On/Off | Off |
| Generator #1 | RF Level | -40dBm |
| Generator #2 | Modulation | Off |
| Generator #2 | Frequency | 14.0212 MHz |
| Generator #2 | Carrier On/Off | Off |
| Generator #2 | RF Level | -40 dBm |
| **Spectrum Analyzer | Input Attenuation | 20 dB |
| Spectrum Analyzer | Reference Level | -40 dBm |
| Spectrum Analyzer | Threshold | -110 dBm |
| Spectrum Analyzer | Resolution Bandwidth | 100 Hz |
| Spectrum Analyzer | Video bandwidth | 10 kHz |
| Spectrum Analyzer | Center Frequency | 14.0206 MHz |
| Spectrum Analyzer | Span | 20 kHz |
| Spectrum Analyzer | Sweep Time | Auto |

** NOTE: The spectrum Analyzer is automatically configured to these values when using the LabWindows/CVI software.

6.7B.5 Allow all equipment to warm up for at least 10 minutes before proceeding to the next step.

6.7B.6 Enable the carrier on Signal Generator #1. Proceed with DUT tuning adjustments as described in the Operation Test. When the DUT tuning is complete, disable the carrier on Signal Generator #1 and reduce its level back to -40 dBm.

6.7B.7 Enable the carrier on both signal generators. Simultaneously, the RF level of both generators slowly until the DUT is operating at about 50% of its rated power output.

6.7B.8 Observe the spectrum analyzer display and adjust the signal generator levels for equal power of the two signals on the spectral display. (Be sure to monitor the DUT's Grid and Plate Current if DUT is a Tube type to keep them within the manufacturer's limits.)

6.7B.9 Adjust the impedance matching controls on the test fixture for minimum SWR (Note that the dummy load matching network shouldn't have to be readjusted after the initial adjustment unless a fixture change has been made).

6.7B.10 It the DUT is not yet at full rated power at this point, slowly increase the signal generator levels simultaneously to bring the power up to the rated full power output. Adjust the step attenuators to adjust the signal peaks to 6 dB below the reference level on the spectrum analyzer.

6.7B.11 Readjust the fixture impedance matching controls if needed and trigger the spectrum analyzer sweep to obtain a static trace. Disable the signal generator carriers to remove the drive to the DUT. *Is it getting hot in there or what ?*

6.7B.12 If using the LabWindows/CVI software, click on the READ ANALYZER button to acquire data from the analyzer and display it on the computer screen. The level of the IMD products, in dB below the PEP level can be directly read off the displayed data. Save the results to a file.

6.7B.13 Record the 3rd and 5th IMD product levels on the data sheet.

# LINEAR AMPLIFIER PRODUCT REVIEW DATA SHEET

Manufacturer:_____ Model:_____Serial #:_____

Date:_____ Test Engineer:_____

| | | DRIVE DUT | | | Bird | | |
|---|---|---|---|---|---|---|---|
| Band (M) | Frequency (MHz) | PFWD (W) | PREF (W) | Po (W) | Po (W) | SP/PUR (dBc) | NOTES |
| 160 | 1.801 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 1.900 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 1.999 | ____ | ____ | ____ | ____ | _____ | _____ |
| 80 | 3.501 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 3.750 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 3.999 | ____ | ____ | ____ | ____ | _____ | _____ |
| 40 | 7.001 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 7.299 | ____ | ____ | ____ | ____ | _____ | _____ |
| 20 | 14.001 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 14.175 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 14.349 | ____ | ____ | ____ | ____ | _____ | _____ |
| 17 | 18.069 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 18.167 | ____ | ____ | ____ | ____ | _____ | _____ |
| 15 | 21.001 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 21.250 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 21.449 | ____ | ____ | ____ | ____ | _____ | _____ |
| 12 | 24.891 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 24.989 | ____ | ____ | ____ | ____ | _____ | _____ |
| 10 | 28.001 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 28.400 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 29.000 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 29.699 | ____ | ____ | ____ | ____ | _____ | _____ |
| 6 | 50.001 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 50.400 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 51.000 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 52.000 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 53.000 | ____ | ____ | ____ | ____ | _____ | _____ |
| | 53.999 | ____ | ____ | ____ | ____ | _____ | _____ |

Worst Case Spectral Purity: _____

Two-Tone IMD, 20 Meters: 3rd:____ 5th____ 7th: ____ 9th: ____ dBc

Test Engineer's Notes:

_____
_

# About the Linear Amplifier IMD Test Fixture

To test for intermodulation distortion (IMD) in power amplifiers, two closely spaced RF signals are necessary. Since a single transmitter would introduce its own IMD into the test setup; two transmitters are used instead, with their outputs combined in a power combiner. An inexpensive hybrid combiner may be constructed using RG-8 and RG-11 coax for a particular design frequency. Slight amounts of reactance at the load port and output port can cause degraded isolation between the input ports, so impedance matching networks are used to adjust these ports for a pure resistive load.

For convenience sake, the entire test setup is mounted in a 6-foot tall equipment rack, mounted on wheels for a limited degree of portability.

In place of two amateur transceivers or transmitters, signal generators and appropriate amplification may also be used. This allows better control of the power levels as well as improved frequency accuracy and stability. This is the approach taken in the ARRL Lab.

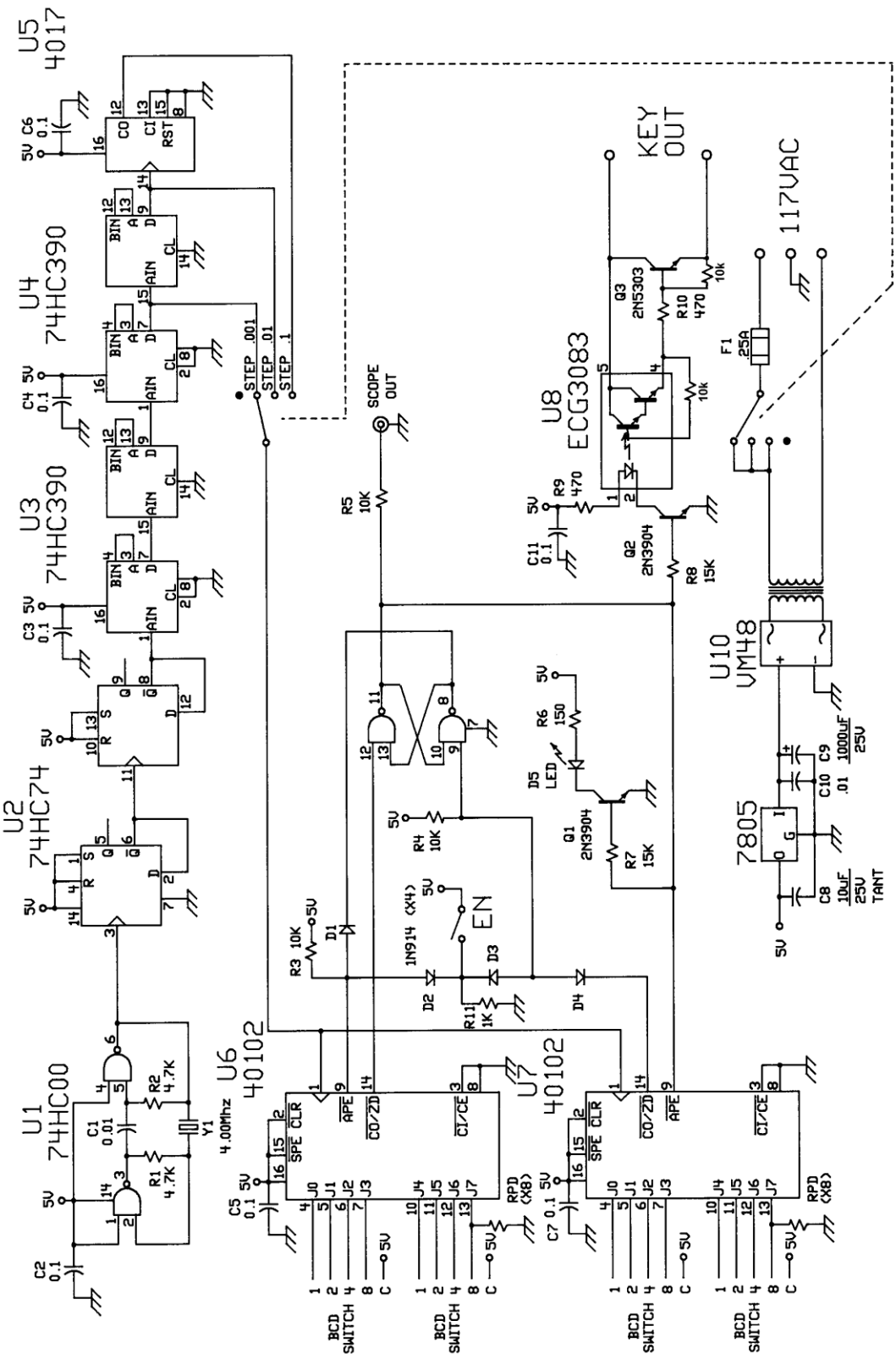
Figure 6.7B provides an overview of the test fixture and interconnections used in the test setup. The two signal generators used must be capable of an output up to about +10dBm. Drivers 1 & 2 are an ARRL design. The schematic of the drivers appears in figure 6.7BA. The linear amplifiers are Communications Concepts model EB63. The filters are 20 meter band pass units made by Ed Wetherhold, W3NQN and were described in the May and June 1998 issues of QST. A copy of this article appears in Appendix B. Similar filters may be substituted. The power combiner is a coaxial hybrid ring adapted from a design for 2 meters and 440 MHz that appeared in the 1998 Central States VHF Conference Proceedings. A copy of this paper also appears in Appendix B.

The RF Sampling Unit is also an ARRL design. The schematic for this circuit appears in Appendix B. The Directional Coupler is a Tandem Match Design. The circuit of the Tandem Match appeared in January, 1987 QST (also in Appendix B). The PI network used to tune out the stray reactance of the dummy load consists of a 0.432 micro Henry coil and two variable capacitors with a nominal value of 104 Pico farads. The T network used to match the input impedance of the amplifiers under test consists of a 1.07 micro Henry coil and two variable capacitors with a nominal value of 70 Pico farads.

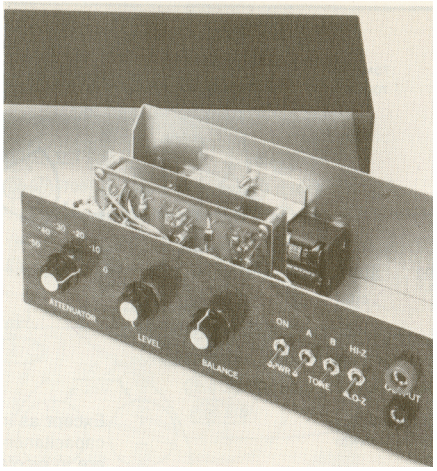
# Appendix A

ARRL custom test circuits:

## Keying Generator


U1
74HC00
U2
74HC74
U3
74HC390
U4
74HC390
U5
4017
U6
40102
U7
40102
U8
ECG3083
U10
VM48
7805
EN
KEY
OUT
117VAC
SCOPE
OUT
STEP .001
STEP .01
STEP .1
4.00Mhz
BCD
SWITCH
RPD
(X8)
1N914 (X4)
LED
2N3904
2N5303
25A
TANT

# Two-Tone Audio Generator




Exterior view of the two-tone audio generator.

## A TWO-TONE AUDIO GENERATOR

The audio frequency generator shown above makes a very convenient signal source for testing the linearity of a single-sideband transmitter. To be suitable for transmitter evaluation, a generator of this type must produce two non-harmonically related tones of equal amplitude. The level of harmonic and intermodulation distortion must be sufficiently low so as not to confuse the measurement. The frequencies used in this generator are 700 and 1900 Hz, both well inside the normal audio passband of an SSB transmitter. Spectral analysis and practical application with many different transmitters has shown this generator to meet all of the requirements mentioned above. While designed specifically for transmitter testing it is also useful any time a fixed-frequency, low-level audio tone is needed. Details on distortion measurement and the two-tone test can be found in Chapter 18.

**Circuit Details**

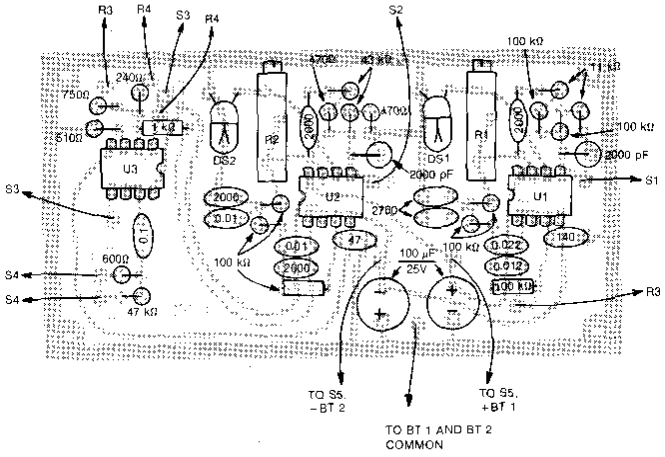
Each of the two tones is generated by a separate Wein bridge oscillator, U1B and U2B. The oscillators are followed by RC active low-pass filters, U1A and U2A. Because the filters require nonstandard capacitor values, provisions have been made on the circuit board for placing two capacitors in parallel in those cases where standard values cannot be used. The oscillator and filter capacitors should be polystyrene or Mylar film types if available. Two tones are combined at op amp U3A. This amplifier has a variable resistor, R4, in its feedback loop which serves as the output LEVEL control. While R4 varies both tones together, R3, the BALANCE control, allows the level of tone A to be changed without affecting the level of tone B. This is necessary because some transmitters do not have equal audio response at both frequencies. Following the summing amplifier is a step attenuator; S3 controls the output level in 10-dB steps. The use of two output level controls, R4 and S3, allows the output to cover a wide range and still be easy to set to a specific level.

The remaining op amp, U3B is connected as a voltage follower and serves to buffer the output while providing a high-impedance load for the step attenuator. Either high or low output impedance can be selected by S4. The values shown are suitable for most transmitters using either high- or low- impedance microphones.

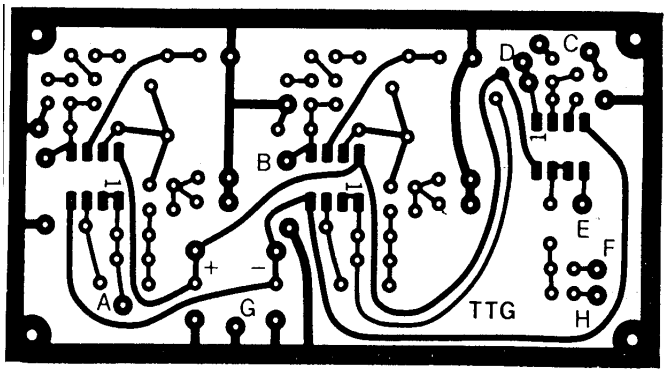
**Construction and Adjustment**

Component layout and wiring are not critical, and any type of construction can be used with good results. For those who wish to use a printed-circuit board, a parts placement guide is shown on the next page. Because the generator will normally be used near a transmitter, it should be enclosed in some type of metal case for shielding. Battery power was chosen to reduce the possibility of RF entering the unit through the ac line. With careful shielding and filtering, the builder should be able to use an ac power supply in place of the batteries.

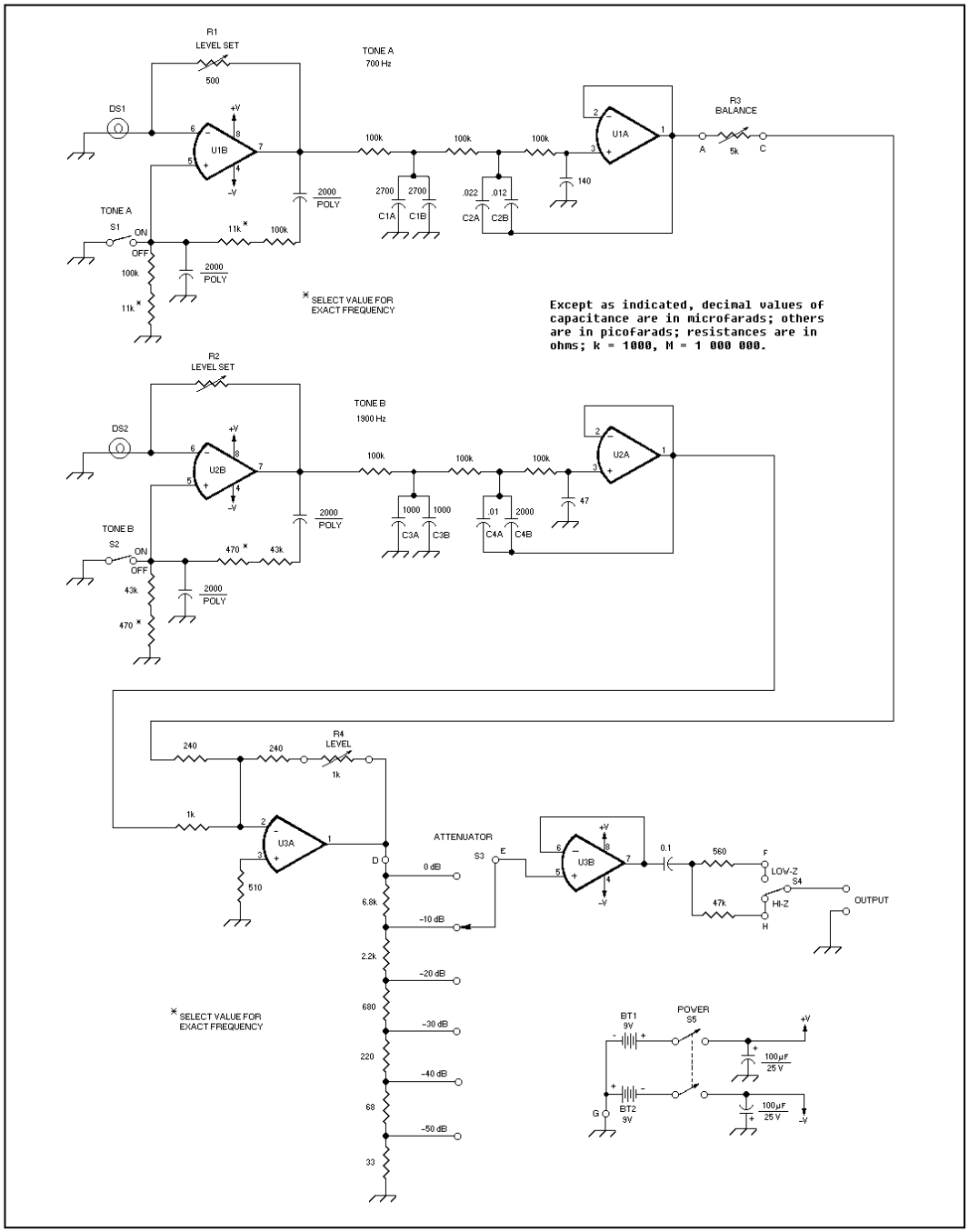
The only adjustment required before use is the setting of the oscillator feedback trimmers, R1 and R2. These should be set so that the output of each oscillator, measured at pin 7 of U1 and U2, is about 0.5 volt RMS. A VTVM or oscilloscope can be used for this measurement. If neither of these is available, the feedback should be adjusted to the minimum level that allows the oscillators to start reliably and stabilize quickly. When the oscillators are first turned on, they take a few seconds before they will have stable output amplitude. This is caused by the lamps, DS1 and DS2, used in the oscillator feedback circuit. This is normal and should cause no difficulty. The connection to the transmitter should be through a shielded cable.




Parts-placement diagram for the two-tone audio generator, shown from the component side of the board.



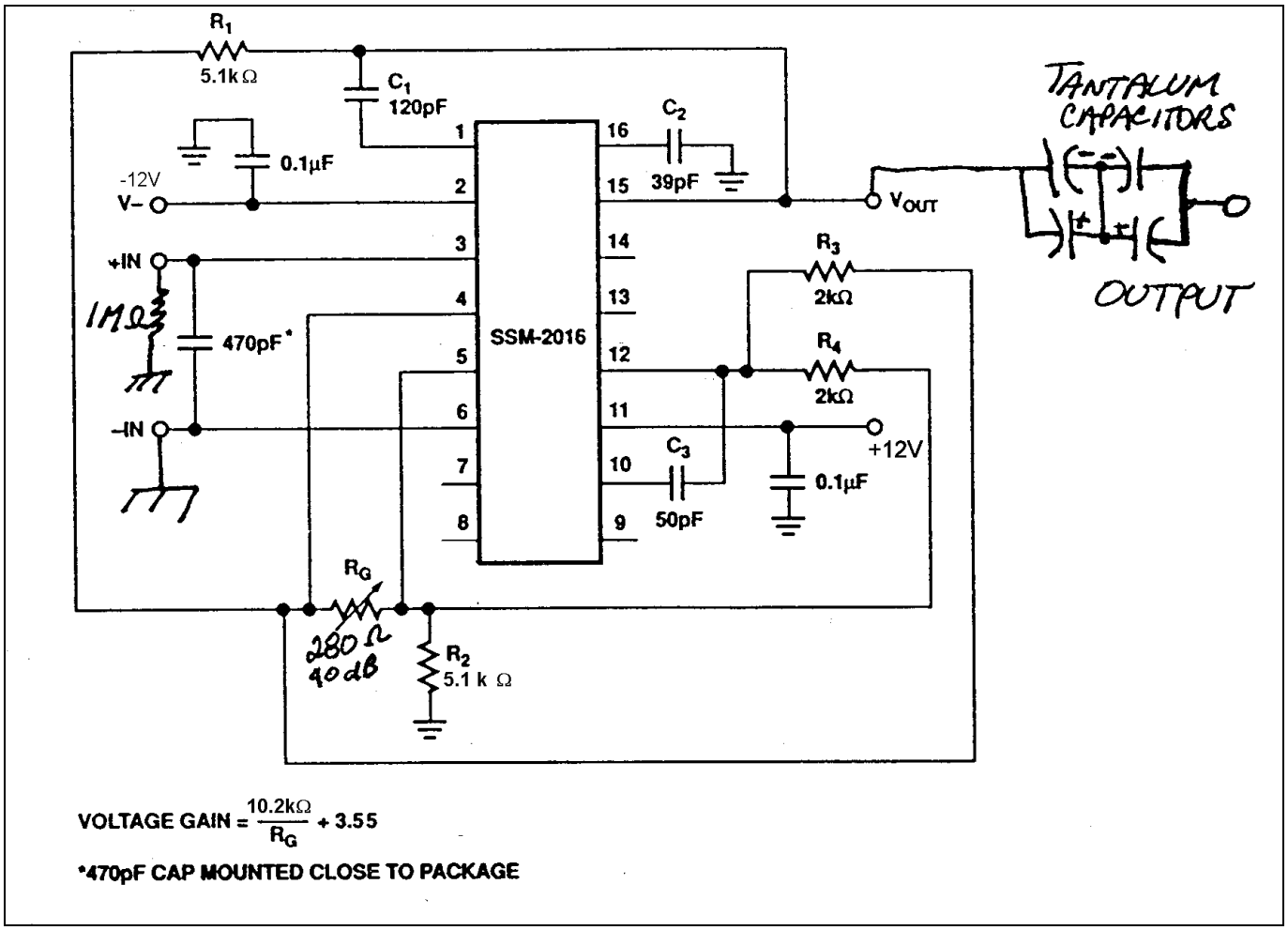

Printed-circuit board etching pattern, shown from the foil side of the board.




Schematic diagram of the two-tone audio generator. All resistors are 1/4-W carbon film.

BT1, BT2--9V Alkaline.
C1A, B--Total capacitance of 0.0054 µF, ±5%
C2A, B--Total capacitance of 0.034 µF, ±5%
C3A, B--Total capacitance of 0.002 µF, ±5%
C4A, B--Total capacitance of 0.012 µF, ±5%
DS1, DS2--12V, 25mA lamp.
R1, R2--500Ω, 10-turn trim potentiometer.
R3--500Ω, panel-mount potentiometer.
R4--1kΩ, panel-mount potentiometer.
S1, S2--SPST toggle switch.
S3--Single-pole, 6-position rotary switch.
S4--SPDT toggle switch.
S5--DPDT toggle switch.
U1, U2, U3--Dual JFET op amp, type LF353N or TL082.

# Low Noise Amplifier




Audio Low Noise Amplifier Calibration Procedure

Connect the output of the Marconi 2041 RF signal generator to the input of the HP-6536E Spectrum Analyzer.

Set the signal generator to -50 dBm at a frequency of 10 kHz (no modulation).

Set the spectrum analyzer center frequency to 10 kHz and set the frequency span to about 20 kHz or less.

Note the peak signal level on the spectrum analyzer display. It should be close to -50 dBm, but will not be exact. Record this level.

Switch the input of the spectrum analyzer to the output of the audio low noise amp (LNA).

Set the signal generator level to -99.5 dBm and switch the output of the generator to the input of the LNA.

Observe the signal level on the spectrum analyzer display and adjust the gain trimmer on the LNA for the same signal level as observed previously. The gain will now be the correct value of 49.5 dB.

# BER Test Box

# Measuring 9600-Baud Radio BER Performance

DSP techniques make testing a
G3RUH-compatible radio easy.

[Reproduced from March 1995 QEX]

By Jon Bloom, KE3Z
225 Main Street
Newington, CT 06111
email: jbloom@arrl.org

One of the jobs of the ARRL Lab is to test the performance of equipment sold to amateurs. With the new crop of 9600-baud radios coming out, we had to develop a technique for testing their performance. The best way to test the performance of a radio used for digital communication is, by far, to test the bit-error rate (BER) that the radio provides under various conditions. BER is a conceptually simple metric that answers the question: How many of the bits get through correctly when a data stream is passed through the system?

The system we developed uses a Texas Instruments DSP Starter Kit (DSK) board that includes a TMS320C26 processor. Figs 1 and 2 show the circuitry of the BER test box built in the ARRL Lab. The DSP generates a test signal that is passed through the system under test. For receiver testing, the signal modulates a low-noise FM signal generator that feeds the radio being tested. The demodulated output of the radio goes to the DSP input so that the demodulated signal can be compared to the transmitted signal. To test transmitters, the DSP output test signal is applied to the modulation input of the transmitter under test. The transmitter RF output is attenuated to a low level, then applied to the Lab-built test box where it is mixed with an unmodulated signal from a signal generator. The resulting IF signal is demodulated by a low-distortion demodulator, and the demodulated signal is routed to the DSP input for comparison with the generated signal.

**G3RUH Signals**

The 9600-baud system used for amateur packet radio, both terrestrially and via the UoSat packet satellites, uses signals handled by the G3RUH modem design. In this system, the binary data stream coming out of a TNC is first "scrambled" to remove any dc component of the signal. Scrambling, which has nothing to do with encryption or data hiding, is simply encoding that ensures that, on average, there are as many 1-bits in the data stream as there are 0-bits. The average voltage of the data signal is thus constant. The scrambled data signal is then used to generate shaped pulses. The shaped-pulse signal is what is applied to the modulation input of the FM transmitter.

On reception, the shaped-pulse signal is filtered and limited, to recover the scrambled data stream. A clock-recovery circuit generates a clock signal that is synchronous with the incoming data. Using this clock, the modem descrambles the data stream, with the end result being a binary data signal identical to the transmitted data signal.

Since our test system has to generate and receive signals like those of the G3RUH modem, some discussion of the pulse shaping used is necessary. The general problem to be solved when sending digital data using FSK is to limit the bandwidth of the baseband (data) signal before applying it to the modulator. But you have to be careful how you do that.

When we limit the bandwidth of a pulsed signal, we necessarily stretch the pulses in time. That is, a signal that is band limited cannot also be time limited. That means that in limiting the bandwidth, we cause each pulse to overlap adjacent pulses. In theory, each pulse overlaps all of the other pulses, but as you go further away in time from a particular pulse, its amplitude gets smaller and smaller.

So, we need to find some way of allowing the pulses to overlap their neighbors without interference. One approach to doing so is to shape the pulses so that, while each pulse does overlap its neighbors, the amplitude of that pulse is zero at the center of each of the other pulses. That way, we can sample the signal at the center of each pulse period and see only the signal from the current pulse; none of the other pulses contributes any amplitude to the signal at that time.

The spectrum of one pulse that has these characteristics is straightforward. It is flat from 0 Hz out to some chosen frequency, then rolls off with a cosine-shaped curve, reaching -6 dB at one-half the baud rate.[1] It continues curve, reaching -6 dB at one-half the baud rate.[1] It continues rolling off with this cosine curve until it reaches zero. In the G3RUH system, the spectrum begins rolling off at 2400 Hz, reaches the -6-dB point at 4800 Hz and reaches zero at 7200 Hz.

In the time domain, the pulse shape that results from such a spectrum has a maximum at the center of the bit period, then decreases in amplitude as we move away from the center of the bit. The pulse signal goes negative, passing through zero at the center of the preceding and following bits. As we go farther away in time from the bit center, the signal alternates between positive and negative values, always passing through zero at the center of each bit. There are other spectra that have pulse shapes that reach zero at the center of all the other pulses, but the benefit of the raised-cosine spectrum is that the pulse amplitude falls off rapidly with time. This is important because any amplitude or phase distortion present in the system is likely to cause the zero-crossing points of the pulse to shift in time, causing ISI. Since the amplitude of the pulse is small near the center of other pulses, the potential for harmful ISI is also small.

Since each bit of the shaped-pulse signal now extends across multiple bit periods -- both preceding and following bits periods -- we must take this into account in generating our signal. And, since this is DSP, what we are generating is a sampled version of the signal. What we must end up with at any sample is a signal that comprises a component from the current bit, preceding bits and following bits. Theoretically, we need components from all of the bits in the data stream, but the amplitude of each pulse falls off so rapidly that only a few successive bits need be used to generate any given sample. In this system, we chose to include components of the current bit and the four preceding and four following bits.

Note that only 1-bits contribute to the signal; 0-bits generate no pulse. If we were to send a continuous stream of 0-bits, we'd get no pulses at all. (Of course, the scrambler circuit ensures that will never occur in a real transmitted data stream.) So, for each sample we need to add up the contribution of the current bit and the contributions of eight other bits, some of which may be 0-bits that make no contribution. The contribution of a particular preceding or following 1-bit is the amplitude of its pulse at the present time. To keep things simple, we use a sampling rate that is a multiple of the baud rate -- and thus a multiple of the pulse rate.

The method used to implement this pulse forming is an FIR filter. The impulse response of the filter is simply the samples that comprise a single shaped pulse, extending over nine bit periods. Our sampling rate is four times the bit rate, so the impulse response is 4 X 9=36 samples long. A Mathcad 5.0 worksheet, shown in Fig 3, calculates the impulse response.

If we feed a single sample of amplitude 1 into the filter, preceded and followed by 0-amplitude samples, the resulting output will be a single copy of our shaped pulse, as shown in the Mathcad graph in Fig 3. To generate our data stream, we input the present data value (1 or 0) at one sample, follow it with three samples of 0, then input the next data value. At any time, the FIR filter contains zeroes in all except (possibly) 9 locations, representing the 9 successive data bits. The output of the filter at any sample time comprises components of 9 data pulses, which is what we want for our shaped-pulse signal.

Now that we know how to generate the shaped-pulse signal from a data stream, we need to think about how to generate the data stream itself. We want to generate a data stream that mimics the scrambled signal of the G3RUH modem. The scrambler in the modem is a tapped shift register with feedback. The logic equation for this circuit is:

$$y = x0 + x12 + x17$$

where y is the output of the scrambler, x0 is the current input bit, x12 is the 12th previous input bit and x17 is the 17th previous input bit. Generating the test data stream is done by making x0 always a 1, implementing a shift register in software, and calculating y for each new data bit -- once every four samples. Note that this is essentially the same signal generated by the G3RUH modem in its BER test mode.

**Counting Bit Errors**

Having generated a test signal, we now need to consider how to compare the signal coming back from the system under test to the signal we transmitted. In passing through the tested system, the signal will be delayed by some amount, and the amount of delay will vary from one system to another. What we need to do is determine what the system delay is, then remember what our transmitted signal was that far back in the past in order to compare it to the samples of the received signal. But we are only interested in the value of the received signal at one time during each bit period: the center of the bit.

The DSP software handles this need using a two-step delay process. First, the operator will tell the DSP system how many samples of delay there are between the transmitted and received signals. Of course, the system delay is not likely to be kind enough to let the center of the received bits fall right onto one of our samples -- there are only four samples per bit, after all. So, the second step is to adjust the phase of the DSP's transmit and receive sample clocks. The combination of these two techniques lets us adjust the DSP's delay with fine resolution.

This calibration procedure is performed using a dual-channel oscilloscope. One channel is connected to the sync signal from the DSK, the other channel is used to display the demodulated signal that is being fed into the DSK input. The operator first commands the DSK to generate a calibration signal. The DSK does so by passing a single 1-bit through the FIR filter described previously, followed by 0-bits. The 1-bit is repeated every 72 samples, resulting in a single shaped pulse every 2 ms. The oscilloscope is triggered on the sync signal and set for 0.2 µsec/div. The operator then sends commands to the DSK to alter the number of samples of delay between the transmitted and received signals.

The sync signal, a short pulse, is output during the sample the DSK believes to be the center of the received bit. Thus the operator simply adjusts the delay value, which causes the received signal trace on the oscilloscope to move relative to the sync pulse. When the sync pulse is aligned with the shaped pulse of the received signal (Fig 4), the DSK is at the proper sample-delay value.

To achieve final calibration, the DSK clock phases must be adjusted to take into account the part of the system delay that is less than one sample period long. This is done by commanding the DSK to output its test signal and switching the oscilloscope to 10 µsec/div. Now the oscilloscope displays a sync pulse at the right side of the screen, along with the eye pattern of the received data (Fig 5). The proper sampling point of the bits is the point where the eye is most "open." However, there is a 14-µsec delay in the DSK system between the actual sampling point and the sync pulse. Thus, the operator commands the DSK to step its clock phase until the center of the eye leads the sync pulse by 14 µsec. This is not a hugely critical setting -- a few-microsecond error isn't normally detectable in the measured BER. The delay does have to be in the ballpark, however.

There is one more detail to consider during calibration. Depending on the system being tested, the received signal may be inverted from the transmitted signal; the positive-going pulses we sent may now be negative-going. If this is the case, the DSK will count all the correctly received bits as bad and all the bad bits as correct. So, the DSK supports a user command to invert the sense of the received data. The polarity of the received signal is most easily seen during the first calibration phase (Fig 4), and if the pulse is seen to be inverted then, the DSK's "invert" command should be issued. That will cause no difference in the displayed signal, but the DSK will know "which way is up."

**DSP Program Operation**

The DSP program used to perform BER measurements is called DBERT. The object file, DBERT.DSK, is downloaded into the DSK from the host computer. Once DBERT is running, it communicates with the host computer via the DSK's serial port. The serial I/O communication mechanism was described in a previous article.[2] When serial I/O is occurring, the DSP interrupts must be disabled. This keeps the DSP from executing its signal-processing operations during serial I/O. For that reason, DSK operation is controlled by the host computer using a command-response sequence: The host sends a command to the DSK, the DSK executes that command -- including any needed signal processing -- and reports completion of the command back to the host, if needed. Each command to the DSK from the host is a single ASCII character. Some commands result in the DSK needing to send data back to the host. In that case, once the command is completed by the DSK, it sends the data back to the host via the serial interface. Table 1 lists the commands supported by the DBERT program.

Because the DSK has to stop its signal I/O while communicating with the host, there is a potential problem in taking a measurement. When the DSK begins generating the test signal, it is possible that the system being tested will exhibit a response to the start-up transient from the DSK that invalidates the result. So, the BER test signal is started and the DSK waits 20,000 samples (about a half-second) before starting to sample the received signal.

We wanted to be able to test at consistent signal-to-noise ratios in order to establish reference levels for comparing different radios to one another. Since each unit will exhibit a unique sensitivity, we needed some way of adjusting the input power level to get a fixed output signal-to-noise ratio. The solution was to include a SINAD measurement function in the DBERT program. When the DSK receives the SINAD command from the host, it generates a 1-kHz sine wave.

About a half-second after it begins generating the sine wave, the SINAD measurement software begins sampling the input signal. It then takes 8192 samples for measurement. Each measured sample is squared and added to a running sum (the signal-plus-noise value). The samples are also run through a narrow 1-kHz IIR notch filter that removes the 1-kHz test signal, leaving only the noise. The output samples from this filter are also squared and added to a (separate) running sum (the noise value). After 8192 samples have been processed, the two 32-bit sum values are returned to the host computer. The host can then calculate the SINAD by expressing the ratio of the signal-plus-noise value to the noise value in dB. A more detailed description of this technique was published in an earlier QEX article.[3]

The analog I/O of the DSK is performed by a TLC32040 integrated analog subsystem. This chip includes a 14-bit D/A, 14-bit A/D, sample-clock generator and input and output programmable switched-capacitor filters (SCF). The TLC32040 is driven by a 10-MHz clock produced by the processor chip. The frequency of this clock, in combination with the programmable dividers in the analog chip, determines the available sampling rates. The design of the BER test software calls for a 38,400 sample-per-second (sps) rate. Unfortunately, this exact rate isn't possible with the 10-MHz clock. The nearest available rate is approximately 38,461.5 sps. This translates to a 0.16% error in the speed of the test signals, which is negligible in the context of BER measurements. However, this error is sufficient to make the test signal unreadable by a G3RUH modem since the clock-recovery loop in that design has a very narrow lock range. I discovered this early on in the development of this system. To check that the problem was in fact the speed difference and not some problem with the signal I was generating, I temporarily removed the 40-MHz master oscillator on the DSK board -- from which the 10-MHz clock is derived -- and connected a signal generator to the clock input through a Schmitt trigger inverter. By setting the signal generator to generate exactly the clock frequency needed to get a 9600-baud signal, I found that the G3RUH modem was quite happy to consider my BER test signal a proper one.

Another problem with the TLC32040 is that the input SCF is a band-pass filter that normally cuts off at about 300 Hz. While the filter cut-off frequency is programmable, it can't be set anywhere near the very low frequency needed for this application. So, the DBERT program configures the TLC32040 to operate without its input filter. This requires that an external antialiasing filter be added. As shown in Fig 2, and described in more detail later, an SCF that cuts off at 10 kHz was added to the BER test box to fill this need. The final characteristic of the TLC32040 to note is that it includes no internal sinx/x correction -- not surprising in a chip with a programmable sample rate. Fortunately, with a sampling rate of 38400 Hz and signals of only up to 7200 Hz, the sinx/x roll-off at the upper end of the signal spectrum is only about 1/2 dB. Still, this seems worth correcting. The TLC32040 data sheet describes a first-order IIR filter that can be used to perform this correction. The Mathcad worksheet of Fig 6 shows the calculation of the coefficients of this sinx/x correction filter, which is implemented in the DBERT program. Fig 6 also shows the predicted output response both before and after the correction, as well as the estimated group delay of the correction filter.

As noted, the calibration procedure requires a sync pulse. The DSK has no uncommitted output signal lines usable for outputting a pulse, so DBERT outputs the sync pulse on the serial data output line. Since the sync pulse is less than a microsecond long, this won't usually bother the serial I/O chip of the host computer. (We did find one computer that was occasionally confused by the presence of the sync pulse.) The RS-232 signal is available on one of the DSK's expansion connectors. A 10-k ohm resistor connected to this point brings the sync signal out to the front panel of the BER test box. The presence of the resistor protects the RS-232 line from accidental shorting or application of another signal.

**Host Application Software**

We used two different programs. One, which we won't cover in detail here, manages the Lab's computer-controlled signal generator for stepped BER measurements at various signal levels and frequencies. The other program, BERT.EXE, is a C program that communicates with the DSK under operator control. Communication with the DSK is normally performed at 19,200 baud, although slower speeds can be used. To ensure that no serial overruns occur, BERT uses interrupt-driven serial I/O.

The BERT program is quite straightforward. It accepts keyboard commands from the operator and sends them to the DSK. The command set is the same as that of DBERT -- BERT just passes these commands through to the DSK -- with a couple of additions. Also, BERT "knows" about those DSK commands that generate a response from DBERT. When such a command is given, BERT waits for the response from the DSK, then converts the incoming serial bytes to binary values. In the case of BER test values (commands 4 and 5), it prints out the reported number of errors and the BER, calculating the latter based on the number of bits the DSK was commanded to use in its test. For the SINAD command (S), BERT computes and prints the SINAD and the distortion percentage using the values returned by DBERT.

The DBERT program can perform BER tests using 10,000 samples or 100,000 samples. The BERT program also provides a 1-million sample BER test (the 6 command), which it performs by commanding the DSK to perform a 100,000-sample BER test 10 times. BERT sums the values from each of these tests to get the final result. As each BER test result is returned from the DSK, BERT prints the running total of samples, errors and BER. The L command performs the same function, except that it stops if 100 or more total errors have been reported. If the operator selects the calibrate (C), idle (I), sine-wave (N) or quiet (Q) commands, BERT remembers the selected command. Whenever a BER or SINAD test command is performed, BERT waits for that command to complete, then sends the appropriate command to place DBERT in the most recently selected mode from among those listed.

**BER Test Box Hardware**

The BER test box contains several op-amp amplifiers used to keep the input signals within the range of the TLC32040 analog input. The box also contains a mixer and a demodulator, shown in Fig 2, designed by ARRL Lab Engineer Zack Lau, W1VT. To test transmitters, the transmitter output is attenuated with a high-power attenuator down to a level that the SBL-1 mixer, U3, can handle. The LO input of the SBL-1 is driven by a +7-dBm signal from a signal generator set to a frequency 373-kHz below or above the transmitter frequency. The output from the SBL-1 passes through a low-pass filter that removes the sum frequency, leaving only the 373-kHz difference frequency. This signal is applied to an NE604 FM IF chip that contains a limiter, IF amplifier and quadrature FM demodulator. The frequency at which the demodulator operates is set by L1 and its associated silver-mica capacitor. When Zack built this circuit, the components he used just happened to fall at 373 kHz. If you reproduce the circuit, it's likely that your copy will work at a slightly different frequency. This shouldn't present a problem; the exact operating frequency isn't critical.

Since the NE604 operates from a single 5-V supply, its output is a positive voltage. This signal is fed into U4A, which amplifies the signal and also removes the positive offset, so that the result is zero volts when the input signal is at 373 kHz. U4B amplifies this demodulated signal or an external input signal (usually the output of a receiver being tested) with adjustable gain. The signal is finally filtered in an MF4 4th-order Butterworth low-pass SCF, U6 that acts as the antialiasing filter for the DSP input. The cut-off frequency of U6 is determined by its input clock, with the clock being 50 times the cutoff frequency. In our test set, this clock is supplied by an external function generator, but it could as easily be provided by a crystal oscillator and divider chain.

**Interpreting BER Measurements**

Bit errors are (or should be) due mainly to corruption of the signal by noise. Thus, they should be random. That being the case, if you make the same measurement several times, you are likely to get different results each time. But the more bits you send through the system, the more the effect of noise is averaged and the more consistent the measurements will be. That raises the question: How many bits do you need to use to get a valid result? The answer to that question depends on what you mean by "a valid result." The more bits you use, the closer your measurement will be to the "true" BER you would get if you sent an infinite number of bits through the system. Since sending an infinite number of bits through the system is a little, well, impractical, we need to come up with some guidelines for selecting a finite number of bits to use.

Unfortunately, in order to do that we have to make some assumptions about the character of the noise in the system. Fortunately, the assumptions we make hold up pretty well for real systems. (There's nothing new about this; we make assumptions about the character of system noise all the time. For example, when we relate noise to system bandwidth we often assume that the noise is uniformly distributed across the spectrum of interest.)

What we find is that with a given number of bits in the sample set, and a given number of errors within those bits, we can establish a confidence interval. The confidence interval tells us how likely it is that our measurement is within some specified factor of the "true" BER. For example, we might find that our measurement gives us an 80% confidence that the value is within a factor of 3 of the true BER. The end result is that we can get as good a measurement as we want if we are willing to wait for enough bits to go through the system. Of course, at a low BER we don't get many errors, so we need to send a lot of bits!

What's interesting about the confidence interval is that it depends on the number of errors detected. Suppose you made two BER measurements. If for the second measurement you double the number of bits sent and get double the number of errors, you end up with the same BER but a higher confidence. On the other hand, if you double the number of bits but measure the same number of errors (you're measuring at a lower noise level, for instance), you get a lower BER but the same confidence interval. What that means is that all we need to do is to ensure that we have at least the number of errors needed to establish the desired confidence interval.

There are two useful sets of numbers we've used here in the ARRL Lab for our BER testing. If you measure 10 bit errors, you are 95% sure that you are within a factor of about 2 of the true BER. And if you get 100 bit errors, you are 99% sure that you are within a factor of about 1.3 of the true BER.[4] If you look at a curve of BER versus signal-to-noise ratio, you'll find that a factor-of-2 difference in BER occurs with a fraction of a dB change in signal-to-noise ratio. That suggests that a measurement that results in 10 bit errors is a pretty good one, and a measurement that results in 100 bit errors is a very good measurement. That's why the L command exists in the BERT program. It pops 100,000 bits at a time through the system, stopping when 100 or more errors have been reported -- because continuing to measure more bit errors is a waste of time -- or when 1 million bits have been sent. Using a million bits means that a BER of $1 \times 10^{-4}$ (100 bit errors) or worse can be measured with great accuracy, and a BER of $1 \times 10^{-5}$ (10 bit errors) can be measured with decent accuracy. Of course, if you have time on your hands you can measure a BER of $1 \times 10^{-6}$ by sending 10 million bits through the system. (That takes over 17 minutes at 9600 baud!)

**Conclusion**

The system described here has been used to measure a number of 9600-baud radios. Some of the test results will be presented in an upcoming QST article, and future QST reviews of 9600-baud radios will include measurements made using this system. The system has proven to be effective and easy to use. I should add that we spot-checked the results obtained with this system by measuring BER using a G3RUH modem. It gave BER results that were consistently slightly better because its input filter cuts off at a lower frequency than the SCF in the BER test box, improving the signal-to-noise ratio.

The software for this system, including the source code, is available for downloading from the ARRL BBS (860-594-0306) and via the Internet using anonymous FTP from ftp.cs.buffalo.edu. The file name is QEXBERT.ZIP.

**Notes**

(1) Couch, L. W., Digital and Analog Communication Systems (New York: Macmillan, 1993), p 179.
(2) Bloom, J., KE3Z, "Measuring System Response with DSP," QEX, February 1995, pp. 11-23.
(3) Bloom, J., KE3Z, "Measuring SINAD Using DSP", QEX, June 1993, pp. 9-13.
(4) Jeruchim, M. C., Balaban,P. and Shanmugan, K. S., Simulation of Communication Systems (New York: Plenum Press, 1992), pp. 492-501.

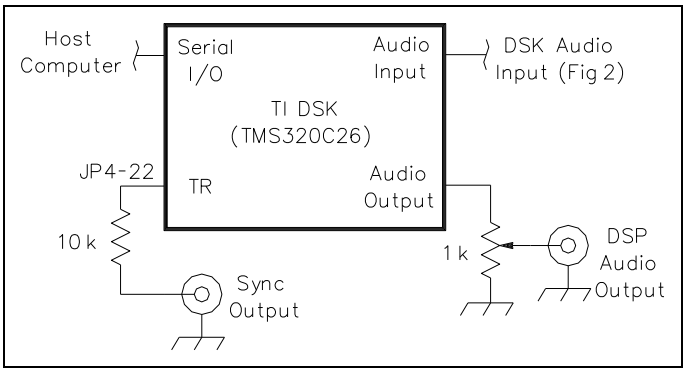


Fig. 1 -- Diagram of the BER test connections to the TI DSK.

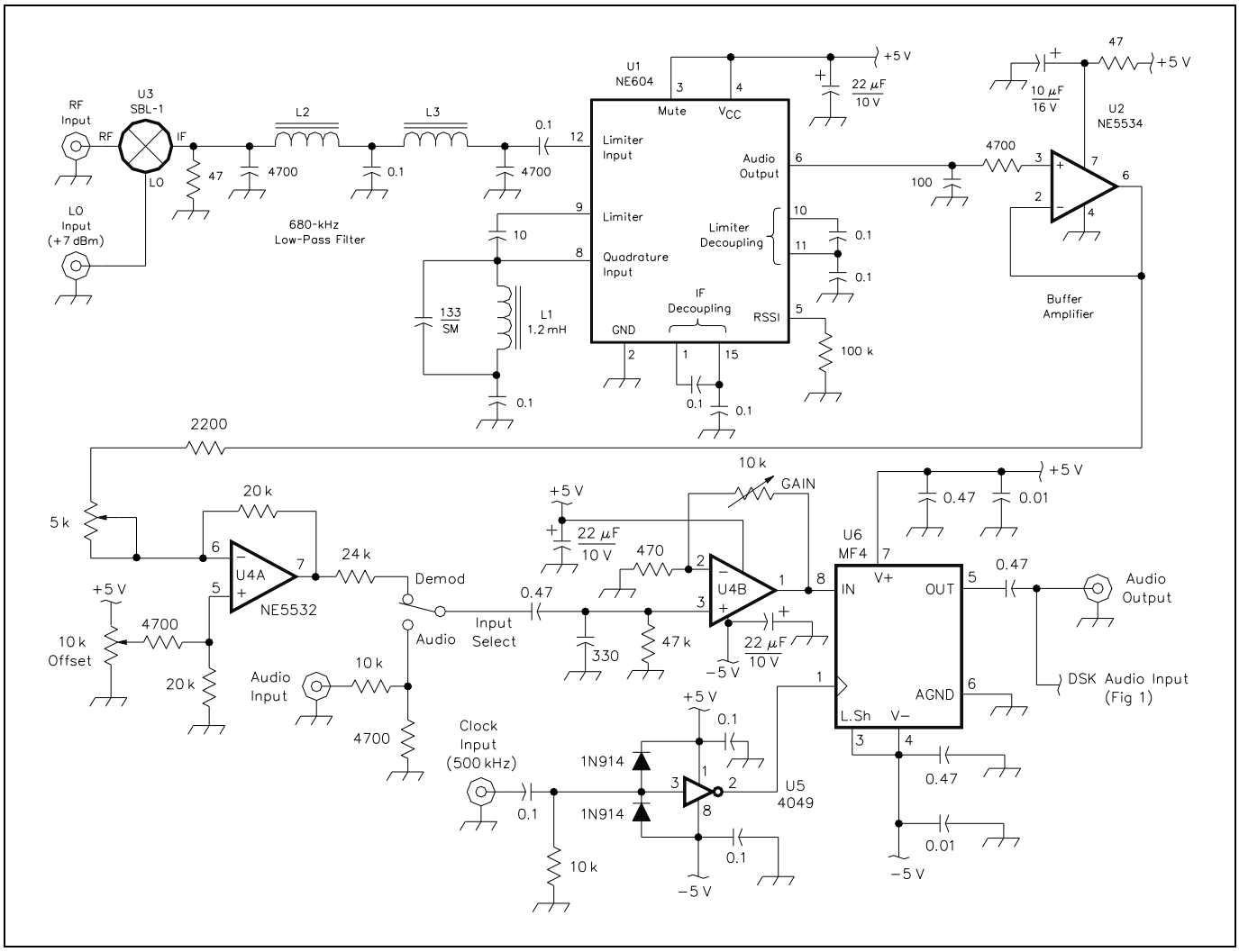



Fig 2 -- Schematic diagram of the BER test box circuit.
L1 -- 1.2-mH Toko 10RB fixed inductor. (Digi-Key part TK4401-ND.)
L2, L3 -- 41 turns #28 enamel wire on a T-50-1 toroid core.

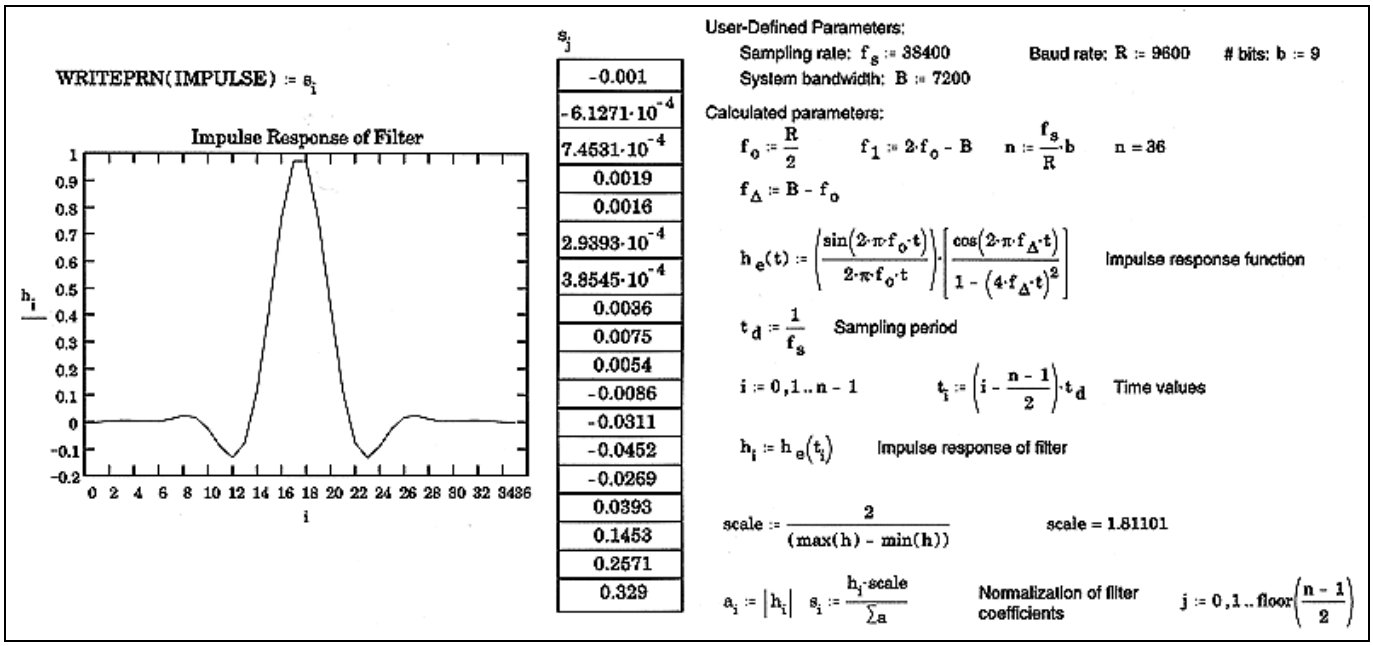



Fig 3 -- This Mathcad worksheet calculates the impulse response of the FIR filter used to generate the shaped-pulse signal.

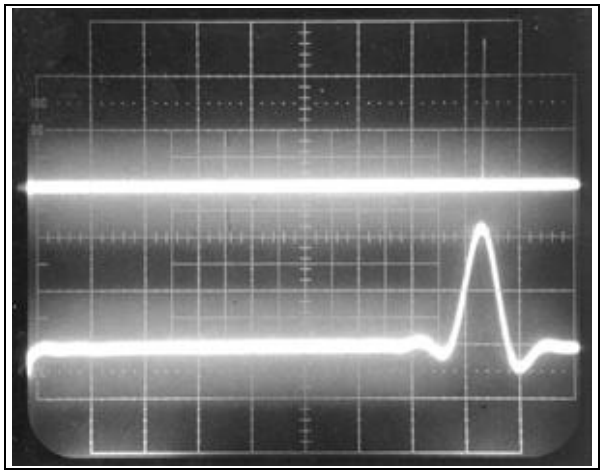


Fig 4 -- The first-phase calibration signal. When properly calibrated, the calibration signal (lower trace) is aligned with the sync pulse (upper trace) as shown. The oscilloscope is set for 0.2 µsec/div. The sync pulse is very narrow and hard to see in this photograph but shows up well on the oscilloscope screen.

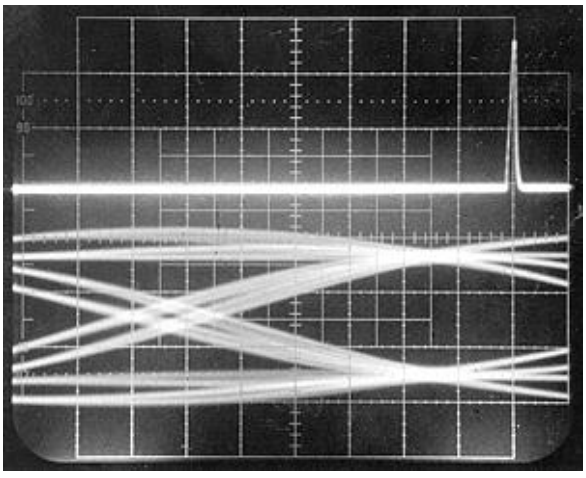


Fig 5 -- Second-phase calibration uses the BER test signal and sync pulse to produce an eye pattern. The oscilloscope is set for 10 µsec/div. The center of the eye is adjusted to lead the sync pulse by 14 µsec as shown.

Fig 6 -- Correction of sinx/x roll-off is performed with a first-order IIR filter having the coefficients calculated by this Mathcad worksheet.

$f_s := 38400$ Sampling frequency $f := 4800$ Upper frequency of passband

$f_n := \frac{f}{f_s}$ $f_n = 0.125$ Normalized upper frequency

$A := \left(\frac{\pi \cdot f_n}{\sin(\pi \cdot f_n)}\right)^2$ $A = 1.0530293$ Amplitude correction at $f_n$

$f_L := \frac{1}{f_s}$ $f_L = 2.6041667 \cdot 10^{-5}$ Normalized lower passband frequency

$A_L := \left(\frac{\pi \cdot f_L}{\sin(\pi \cdot f_L)}\right)^2$ $A_L = 1$ Amplitude correction at $f_L$

Compute needed first-order IIR coefficients

$p_1 := 1$ $p_2 := 2$

Given

$$A \cdot \left(1 - 2 \cdot p_1 \cdot \cos(2 \cdot \pi \cdot f_n) + p_1^2\right) - p_2^2 \cdot (1 - p_1)^2 = 0$$

$$A_L \cdot \left(1 - 2 \cdot p_1 \cdot \cos(2 \cdot \pi \cdot f_L) + p_1^2\right) - p_2^2 \cdot (1 - p_1)^2 = 0$$

$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} := \text{Find}(p_1, p_2)$

$v_1 = -0.1049616$ $v_2 = 1$

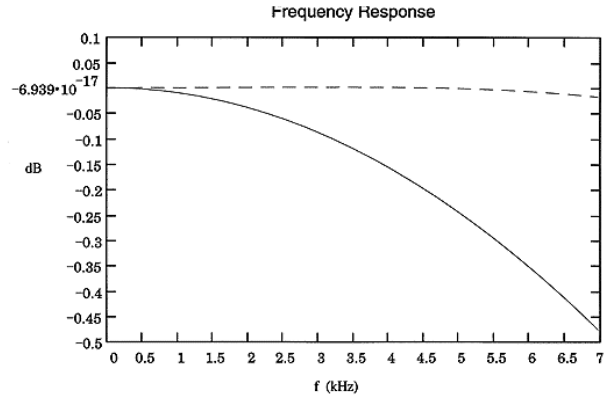
Resulting first-order IIR filter: $y[n]=v_2(1-v_1)x[n]+v_1y[n-1]$

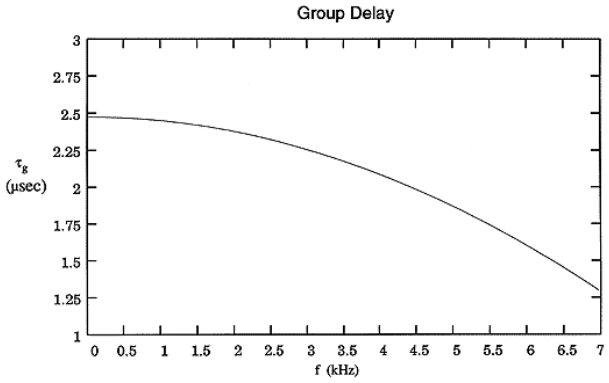
Computation of Corrected System Response

$$S(f) := \frac{\sin\left(\pi \cdot \frac{f}{f_s}\right)}{\pi \cdot \frac{f}{f_s}} \quad M(f) := \frac{v_2 \cdot (1 - v_1) \cdot \exp\left(j \cdot 2 \cdot \pi \cdot \frac{f}{f_s}\right)}{\exp\left(j \cdot 2 \cdot \pi \cdot \frac{f}{f_s}\right) - v_1}$$

$P(f) := \arg(M(f))$

$f_x := 1, 70 .. 7000$ $\omega_x := 2 \cdot \pi, 70 \cdot 2 \cdot \pi .. 7000 \cdot 2 \cdot \pi$



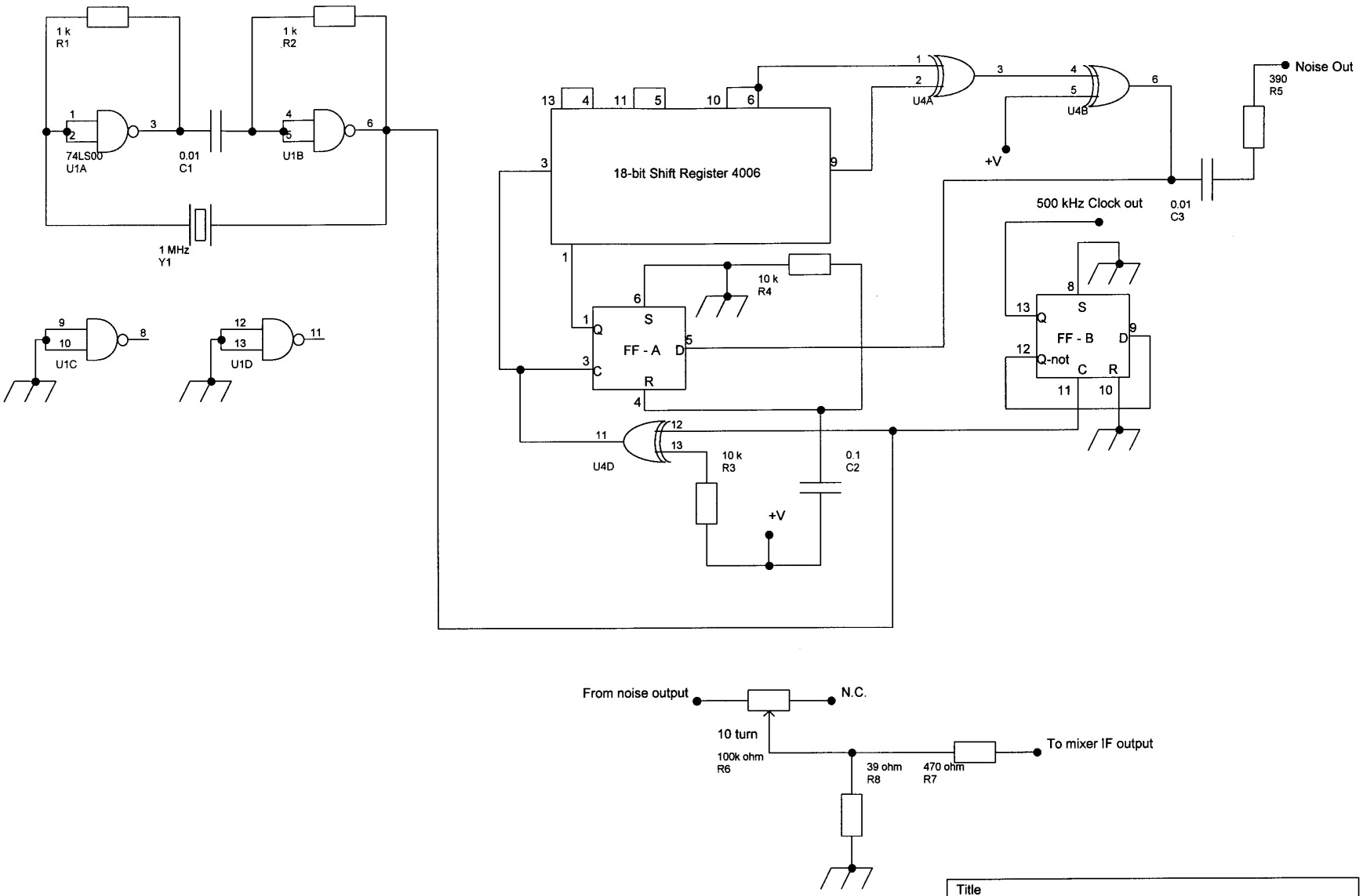



**Update to BER Test Box:**

The ARRL BER test box was modified in March 1999 by the addition of an internal 500-kHz oscillator and high-level pseudo-random noise source. The internal oscillator was added for convenience sake, eliminating the need for an external function generator. The noise source was added to alleviate test setup difficulties imposed by transmitters that generated an excessive amount of common-mode signal.

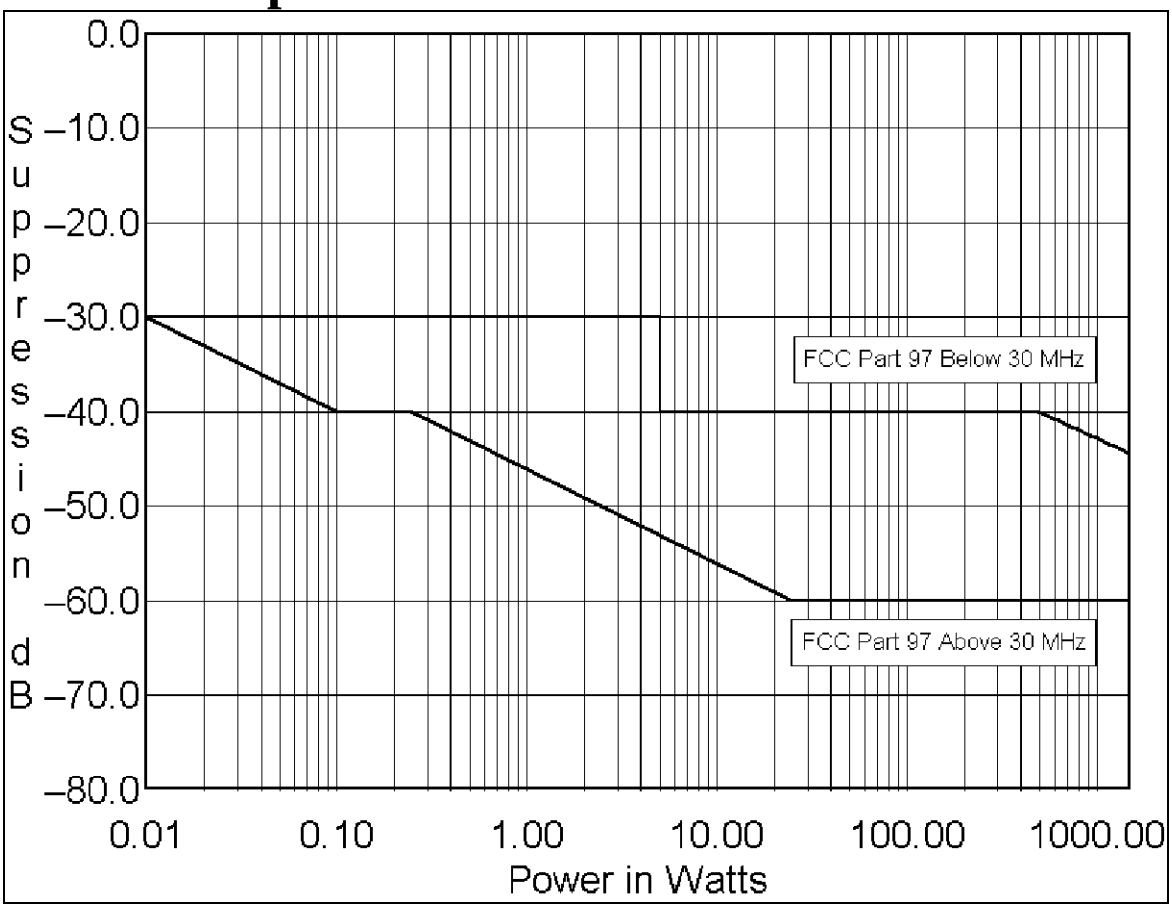
The output of the oscillator is connected to the clock input shown in figure 2 of the QEX article on BER testing. The noise source output is resistor summed into the mixer (U3) in the same figure. A 10-turn potentiometer is used for fine control of noise output so as to achieve as close to 12 dB SINAD as possible.

The schematic of the new circuitry appears on the following page.

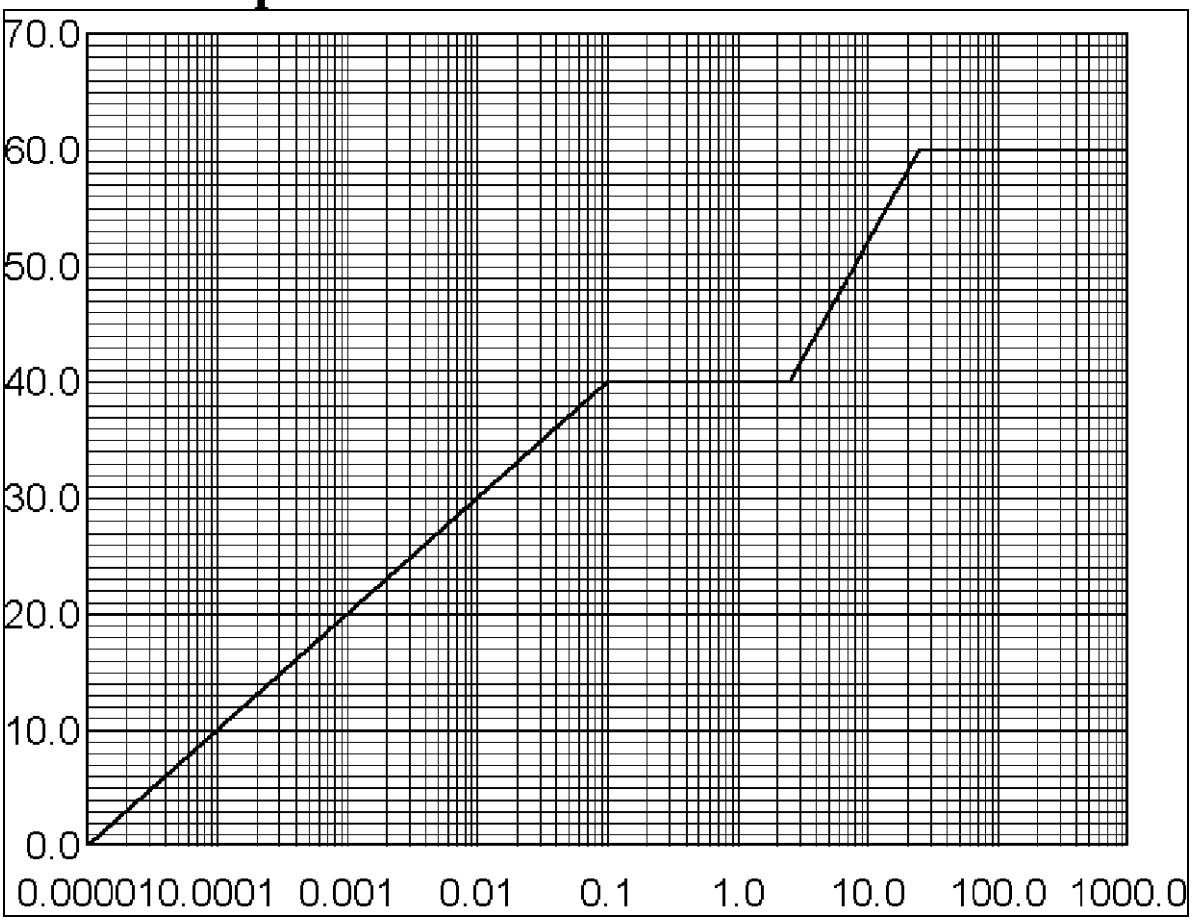



| | | |
|---|---|---|
| Title<br>Pseudo-Noise Source | | |
| Author<br>Michael Tracy<br>American Radio Relay League | | |
| File<br>D:\Program Files\tinycad\noisesrc.dsn | | Document<br>BER1 |
| Revision<br>1.3 | Date<br>15 March 1999 | Sheets<br>1 of 1 |

# Spurious-emission Limits - HF



# Spurious Emission Limits - VHF




**Spurious Attenuation 30-235 MHz**
**X=Power (Watts)**

| **Power Level (output)** | **Attenuation** |
|---|---|
| **X < 0.1W** | **50 + 10LOG(X) dB** |
| **0.1W ≤ X ≤ 0.25W** | **40 dB** |
| **0.25W < X ≤ 25W** | **46 + 10LOG(X) dB** |
| **X > 25W** | **60 dB** |

Given $E$ and $V$ as volts, $P_w$ as power in watts and $dBm$ as decibels relative to a milliwatt, the following relationships are true:

$$E = \sqrt{0.05 \times 10^{\frac{dBm}{10}}} \qquad P_w = 0.001 \times 10^{\frac{dBm}{10}}$$

$$dBm = 10 \times LOG\left(\frac{E^2}{0.05}\right) = 10 \times LOG\left(\frac{P_w}{0.001}\right)$$

$dB\mu = 20 \times LOG(\mu V)$ To reduce μV values by 10 dB, multiply by 0.3162278

S/N in dB = 10 X LOG(10^(0.1 X (S+N/N)) – 1)

Distortion (%) = 100 / 10^(SINAD in dB / 20)
(Assuming meter measures noise as a component of distortion)

$CodeSpeed(WPM) = \frac{2400}{t_{dit-dit}}$ , where $t_{dit\text{-}dit}$ is the time between the rising edge of subsequent dits.

Farnsworth method code speed = $\frac{13.2}{2t_{dot} + 2t_{dash} + 3t_{ele.space}}$

(This equation is based on the letter P in the PARIS word length standard).

**Formulas for SWR, Reflection Coefficient and Return Loss**

For $P_f$ as forward power, $P_r$ as reflected power, $R_l$ as return loss in decibels and $\rho$ as reflection coefficient, the following relationships are true:

$$\rho = \sqrt{\frac{P_r}{P_f}} \qquad SWR = \frac{1+\rho}{1-\rho} \qquad \rho = \frac{SWR-1}{SWR+1} \qquad \rho = e^{\frac{R_l}{-8.68589}}$$

<u>Temperature</u>
+60°C = +140°F
–10°C = +14°F
–20°C = –4°F
–30°C = –22°F

<u>Noise Floor (dBm)</u>
MDS (theoretical) = –174 + NF + 10 LOG(BW)

Where –174 dBm is the noise level of the cosmos, BW is the receiver noise bandwidth in Hertz and NF is the receiver noise figure in dB. MDS is the theoretical noise floor of the receiver, which the actual measured noise floor should be close to.

# ARRL Periodicals Archive – Search Results

## A membership benefit of ARRL and the ARRL Technical Information Service

**ARRL Members:** You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

**Need a higher quality reprint or scan?** Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

**QST on CD-ROM**: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

**Non-Members:** Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.


### Copyright/Reprint Notice

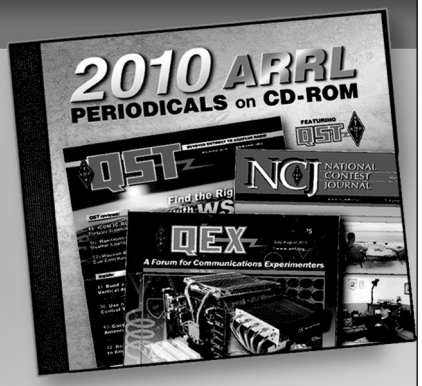
In general, all ARRL content is copyrighted. ARRL articles, pages, or documents--printed and online--are not in the public domain. Therefore, they may not be freely distributed or copied. Additionally, no part of this document may be copied, sold to third parties, or otherwise commercially exploited without the explicit prior written consent of ARRL. You cannot post this document to a Web site or otherwise distribute it to others through any electronic medium.

For permission to quote or reprint material from ARRL, send a request including the issue date, a description of the material requested, and a description of where you intend to use the reprinted material to the ARRL Editorial & Production Department: permission@arrl.org.


**QST Issue:** Sep 1997
**Title:** Build a Return Loss Bridge
**Author:** Jim Ford, N6JF

Click Here to Report a Problem with this File


ADVERTISEMENT




**2010 ARRL Periodicals**
on CD-ROM

**ARRL's popular journals are available on a compact, fully-searchable CD-ROM. Every word and photo published throughout 2010 is included!**

- ***QST*** **The official membership journal of ARRL**
- ***NCJ*** **National Contest Journal**
- ***QEX*** **Forum for Communications Experimenters**

**SEARCH** the full text of every article by entering titles, call signs, names—almost any word. **SEE** every word, photo (including color images), drawing and table in technical and general-interest features, columns and product reviews, plus all advertisements. **PRINT** what you see, or copy it into other applications.

**System Requirements**: Microsoft Windows™ and Macintosh systems, using the industry standard Adobe® Acrobat® Reader® software. The Acrobat Reader is a free download at **www.adobe.com**.

**2010 ARRL Periodicals on CD-ROM**

ARRL Order No. 2001
**Only $24.95***
*plus shipping and handling

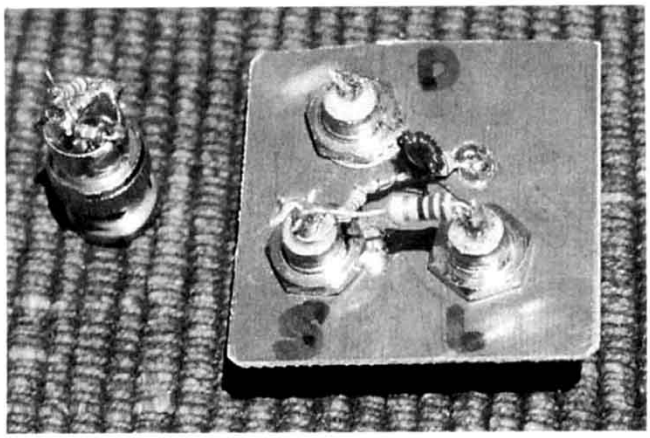
Additional sets available:
2009 Ed., ARRL Order No. 1486, $24.95
2008 Ed., ARRL Order No. 9406, $24.95
2007 Ed., ARRL Order No. 1204, $19.95
2006 Ed., ARRL Order No. 9841, $19.95
2005 Ed., ARRL Order No. 9574, $19.95
2004 Ed., ARRL Order No. 9396, $19.95
2003 Ed., ARRL Order No. 9124, $19.95
2002 Ed., ARRL Order No. 8802, $19.95
2001 Ed., ARRL Order No. 8632, $19.95

ARRL The national association for AMATEUR RADIO™
SHOP DIRECT *or call for a dealer near you.*
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)

Sep 1997 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

By Jim Ford, N6JF

# Build a Return Loss Bridge



Used with your DVM, this simple bridge, diode detector and return loss techniques can help you measure cable loss and SWR at the antenna. The bridge does double duty as a hybrid combiner.

The author's homebrew return loss bridge (right) and peak-reading diode detector built on the back of a male BNC connector (left). The measured bridge directivity is: 2 MHz 24 dB, 4 MHz 26 dB, 10 MHz 28.5 dB, 30 and 50 MHz 30 dB, 146 MHz 34.5 dB.

Return loss (RL) may be a new concept to many radio amateurs. To the uninitiated it may even sound like something bad! It may surprise you, but in this case, *more loss* is *better*. Return loss is the decibel portion of the input power that is taken by the load; greater return loss means less reflected power and lesser SWR. In fact, when the SWR is 1:1 the return loss is infinite!

I've often thought it unfortunate that most amateurs are not familiar with return loss measurements. For some applications, return loss is easier and more useful than SWR measurements. Let's look at a couple of examples. Suppose you know your coax loss and SWR at your transmitter, but want to know the SWR at your antenna feed point. Factoring transmission-line loss into the SWR equations is difficult, but the problem becomes a piece of cake when you use return-loss measurements. Here's the recipe:

$$RL_{Ant} = RL_{TX} - 2\ L_{Coax} \qquad \text{(Eq 1)}$$

Where $RL_{Ant}$ and $RL_{TX}$ are the return losses at the antenna and transmitter, respectively, and $L_{Coax}$ is the coax loss. All three quantities must be in decibels. You may wonder why we double the coax loss in this calculation. The reason is that the power returned by the mismatch travels through the feed line twice—once as forward power, and then again as reflected power.

Let's now consider some real numbers. Suppose the SWR measured at the transmitter is 2:1 and the coax loss is 1.75 dB. The first step of course requires us to convert SWR to return loss. We can translate between several measures of load match by reading a table (see Table 1) or applying math equations.[1] Table 1 shows that a 2:1 SWR equals a return loss of 9.5 dB. Plugging in the numbers, we get:

$$9.5\ \text{dB} - (2 \times 1.75) = 6\ \text{dB} \qquad \text{(Eq 2)}$$

The 6 dB return loss converts to a 3:1 SWR at the antenna feed point.

As you might guess, we can reverse this procedure. Suppose you want to know the SWR at your transmitter if your antenna SWR is 1.5 and coax loss is 1 dB. From Table 1, we see that a 1.5:1 SWR corresponds to a return loss of 14 dB. This time, the calculation looks like this:

$$14\ \text{dB} + (2 \times 1) = 16\ \text{dB}$$

The SWR at the transmitter is about 1.35:1.

## A Return Loss Bridge

By now you may be wondering if it is possible to measure return loss directly. The answer is yes! You can do so with a *return loss bridge* (RLB), which some data sheets call hybrid junctions or hybrid combiners. Figures 1 and 2 show an example RLB. This basic schematic is from Wes Hayward's (W7ZOI) fine book, *Introduction to Radio Frequency Design* (ARRL Order No. 4920; see Note 1 for ordering information).

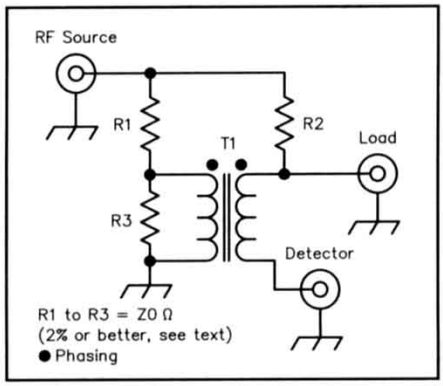



Figure 1—Return loss bridge schematic. T1 is 10 turns bifilar wound #30 AWG enameled wire on an FT-23-77 core. For best accuracy, use 1% tolerance resistors. Use either carbon-composition or metal-film resistors, but keep all leads as short as possible to preserve the useful circuit bandwidth. The cores are available from Amidon.[2]

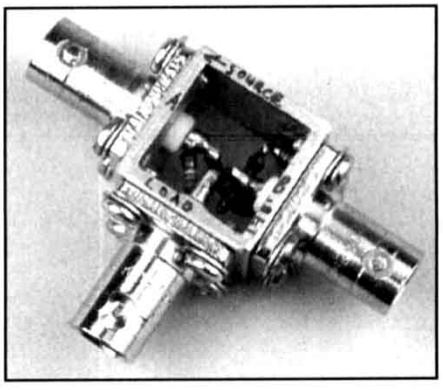


Figure 2—An RLB constructed by ARRL Lab Engineer Zack Lau, W1VT. Zack's construction minimizes lead lengths for maximum bandwidth.

[1]Notes appear on page 36.

Sep 1997 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

### *Advantages*

Return loss bridges offer some advantages over a standard SWR meter. For example, an RLB can measure the mismatch at the input of a receiver or other low-power device without needing 100 W of input power. In fact, 10 mW or so, sometimes less, will do the job (depending on your detector). A signal generator will suffice for most purposes.

The return loss bridge output is RF, compatible with instruments such as spectrum analyzers or log detectors. The amplitude display of a spectrum analyzer, for example, displays dBm (dB relative to 1 mW), and the scale is therefore in decibels. The output of an RLB can therefore be viewed directly on a spectrum analyzer's log (dB) scale.

A return loss bridge can measure coax loss directly, in decibels, much more simply than the standard method (two measurements with power meter and dummy load). The coax must be either unterminated or shorted for this measurement. Figure 3 shows the setup; just connect the coax to the load port of the RLB. The actual coax loss is the return loss divided by two.

A return loss bridge is typically useful over a surprisingly wide spectrum. An RLB and a sweep generator can measure mismatch over a wide frequency range. For example, the commercially made Anzac H9 is rated for use from 2 to 2000 MHz. Such remarkable frequency coverage costs more than $500, and the Anzac is undoubtedly more complex than the RLB shown here.

An RLB is less frequency sensitive than a reflectometer SWR meter. Reflectometers use short, coupled transmission lines to sample $E_f$ and $E_r$, unlike modern, transformer-based directional couplers. The coupling of the short lines increases greatly with frequency, so the instrument sensitivity increases, too. If you have such a meter, you may have noticed that it takes the full calibration-pot (usually labeled **CAL** or **SET**) range to get a full-scale reading on 80 meters, while the same power level and setting easily pegs the meter on 20 meters. In fact, I once had a problem with a QRP transmitter that had (unknown to me) excessive harmonic output. The harmonic energy effectively filled in the SWR meter null because of the instrument's increased sensitivity at the harmonic frequencies. This prevented me from seeing the match at the fundamental frequency.

Return loss bridges can also function as signal combiners. In fact, you may have noticed that the circuit for "A Hybrid Combiner for Signal Generators" in Chapter 26 of *The ARRL Handbook* is identical to an RLB! Combiners can mix two signals into one, while simultaneously isolating the two signal sources. This isolation prevents the generators from interacting and provides a two-tone test signal that is free of intermodulation and other distortion.

The ARRL Laboratory uses combiners with a pair of signal generators for receiver dynamic-range tests. You can perform such tests at home with an RLB. To obtain a two-tone signal, connect signal generator A to the **SOURCE** jack and signal generator B to the **DETECTOR** jack. The two-tone output appears at the **LOAD** jack. Chapter 26 of *The ARRL Handbook* describes the complete procedure under "Receiver Performance Tests."

### *Disadvantages*

Probably the biggest disadvantage of this return loss bridge is that you cannot leave it in the transmission line during normal station operation. There is no straight through path from source to load as most directional couplers for SWR meters. In fact, the RLB has a 6 dB loss from the source to the load. The RLB resistors must dissipate 75% of your transmitter power. For 100 W operation the resistors would be too large for a practical RLB, and losing 75% of your transmitter's output power to a return loss bridge is simply not acceptable to most amateurs.

An RLB has only one output port (**DETECTOR**). Most directional couplers have separate forward and reverse outputs, so they can simultaneously measure both forward and reverse power.

## Theory and Operation

For a 50 Ω ($Z_0$ = 50 Ω)system all three resistors should also be 50 Ω. When the **LOAD** port connects to 50 Ω, the voltages at the **LOAD** port and the R1-R2 junction are equal, resulting in no RF output. In other words the bridge is balanced. T1 serves as a balun; it allows unbalanced measurement of the RF output.

When the **LOAD** port is either open or short circuited, the voltage output from the bridge is maximum. This maximum output is the reference point (0 dB RL) against which we measure the return loss when an antenna, etc is the load. For example, if an open-circuit voltage of 1 (E1) drops to 0.1 V (E2) with the load connected, the reflection coefficient, ρ, is (E2/E1) 0.1, return loss is 20 dB and SWR is 1.22:1 (from Table 1).

The output from the **DETECTOR** port is RF. How do you measure RF of this small power level? There are several solutions. You can use an oscilloscope to measure the peak voltage.[3] Even if you don't have the latest oscilloscope, yours may be adequate. Only the level *change* from open to loaded circuit concerns you; absolute accuracy is not necessary. If you can get near-full-scale deflection on any range, it should work okay. So, even a 50 MHz 'scope should work just fine—even at 2 meters.

An RF micropower meter would also work. Some surplus power meters—such as an HP431 or HP432—are available at reasonable prices, but beware! Often their RF sensor cable is missing or the power head is defective—having been destroyed by an overload (10 mW, maximum, for some models). If you are measuring power, don't forget to find the RL by using the proper column of Table 1.

You can also measure return loss with a diode detector, as shown in Figure 4. The detector and a DVM measure the peak RF voltage. The forward voltage is that with the **LOAD** port open; the reflected voltage is that with the load attached. Diode detectors are not very linear at low power levels, but that is not important if a measured calibration curve relates the dc output voltage to the RF power input. This detector has more than enough accuracy for use with a return loss bridge. This article's lead photo shows that small parts permit us to build the detector *inside* a UHF or BNC connector. It's possible to calibrate a diode detector for good accuracy, but you don't need extreme accuracy to measure power *ratios*.[4] You can use the data in Figure 4 for yours and avoid the calibration. Beware! Keep lead lengths as short as possible to preserve detector bandwidth. Larger connectors can degrade performance by requiring longer leads.

## Caveats

Directivity, the isolation between the **SOURCE** and **DETECTOR** ports, is a key measure of directional-coupler or return-loss-bridge performance. We measure directivity in decibels, and more than 20 dB is desirable. Poor directivity in an RLB shows up as RF leakage at the detector port, even with a perfect load (remember, we want no output).[5] Directivity serves as a floor for mismatch measurements. With 20 dB of directivity, we can measure no more than 20 dB RL; the minimum measurable SWR would be about 1.2:1. With 30 dB of directivity, we could measure an SWR of about 1.07:1.

The impedance of the signal source also limits RLB accuracy. Any differences among

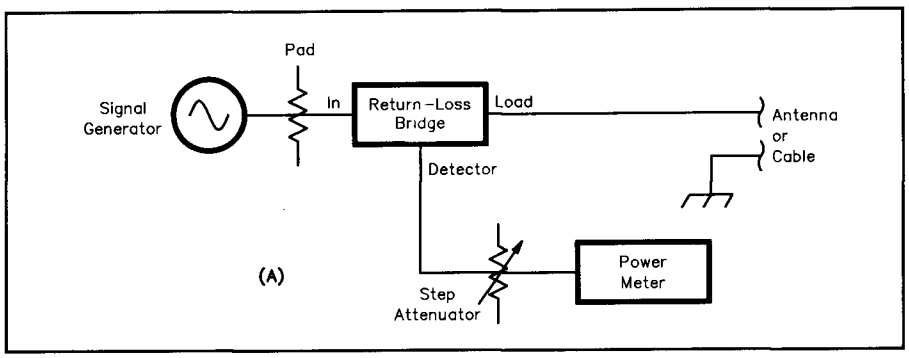



Figure 3—An RLB test setup to measure antenna-system match or coax loss.

Sep 1997 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

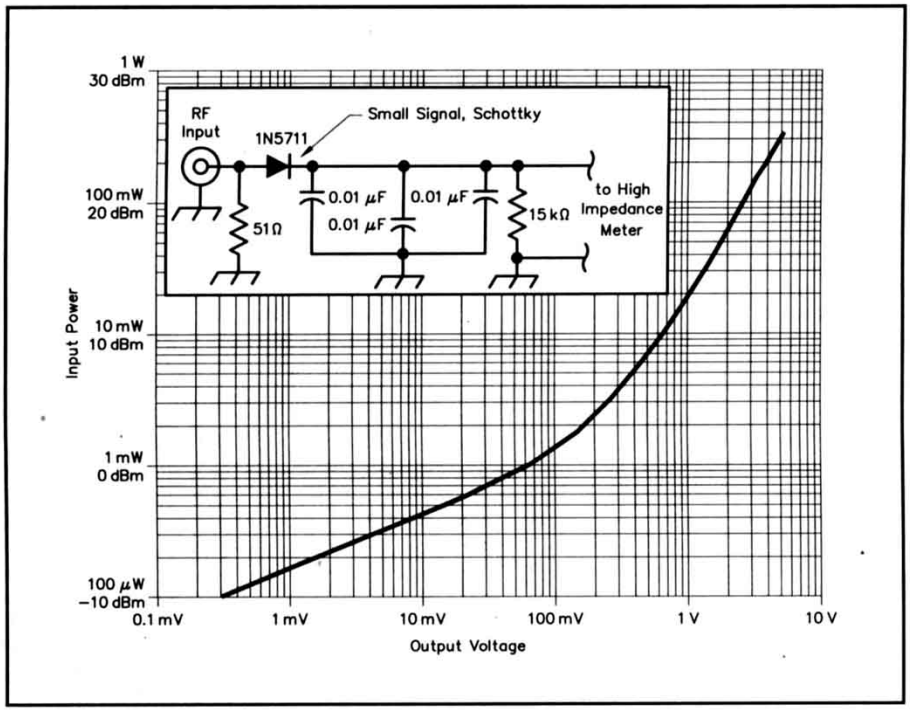



Figure 4—Schematic of the author's diode detector and the detector's calibration curve. All data was taken at 30 MHz. Unless otherwise specified, use 1/4 W, 5% tolerance carbon composition or film resistors.

the source and R1 through R3 (usually 50 Ω) unbalances the bridge, changes the open/short circuit 0 dB reference output and degrades accuracy. Most good signal generators are sufficient in this respect, but if you suspect yours you can improve it. Connect a 3 or 6 dB attenuator in line between the signal generator and the RLB. This improves the return loss of the generator by 6 or 12 dB, but also reduces the available signal level.

**Table 1**

**ρ, ρ², Return Loss and SWR**

| ρ (V) | $P_r/P_f$ (ρ²) | RL (dB) | SWR |
|---|---|---|---|
| 0.00 | 0.00 | ∞ | 1.00 |
| 0.05 | 0.00 | 26.4 | 1.10 |
| 0.09 | 0.01 | 20.8 | 1.20 |
| 0.10 | 0.01 | 20.0 | 1.22 |
| 0.13 | 0.02 | 17.7 | 1.30 |
| 0.15 | 0.02 | 16.5 | 1.35 |
| 0.17 | 0.03 | 15.6 | 1.40 |
| 0.20 | 0.04 | 14.0 | 1.50 |
| 0.23 | 0.05 | 12.7 | 1.60 |
| 0.25 | 0.06 | 12.0 | 1.67 |
| 0.26 | 0.07 | 11.7 | 1.70 |
| 0.29 | 0.08 | 10.9 | 1.80 |
| 0.30 | 0.09 | 10.5 | 1.86 |
| 0.31 | 0.10 | 10.2 | 1.90 |
| 0.33 | 0.11 | 9.5 | 2.00 |
| 0.35 | 0.12 | 9.1 | 2.08 |
| 0.40 | 0.16 | 8.0 | 2.33 |
| 0.45 | 0.20 | 6.9 | 2.64 |
| 0.50 | 0.25 | 6.0 | 3.00 |
| 0.55 | 0.30 | 5.2 | 3.44 |
| 0.60 | 0.36 | 4.4 | 4.00 |
| 0.65 | 0.42 | 3.7 | 4.71 |
| 0.70 | 0.49 | 3.1 | 5.67 |
| 0.75 | 0.56 | 2.5 | 7.00 |
| 0.80 | 0.64 | 1.9 | 9.00 |
| 0.85 | 0.72 | 1.4 | 12.33 |
| 0.90 | 0.81 | 0.9 | 19.00 |
| 0.95 | 0.90 | 0.4 | 39.00 |
| 1.00 | 1.00 | 0.0 | ∞ |

## Conclusion

It is much easier to measure and work with mismatch and attenuation in terms of return loss, rather than SWR. Return loss will probably never replace traditional SWR measurements, but knowledge of both techniques can be very helpful. Despite some limitations, a return loss bridge is a very good way to measure mismatch (especially for low-power applications) and coax loss. With its additional ability as a hybrid combiner, this circuit—just four parts—can be a versatile and valuable addition to your test bench.

### Notes

[1] *The 1997 ARRL Handbook* (ARRL Order No. 1743) contains more information on return loss bridges and hybrid combiners in Chapter 26. You can also find a more detailed mathematical discussion of transmission lines and their associated losses in Chapter 24 of *The ARRL Antenna Book* (18th ed., ARRL Order No. 6133). If you're like me, however, you'll be happy to learn that mathematics beyond simple arithmetic is unnecessary to appreciate or apply return loss.—*Ed.* ARRL publications are available from your local ARRL dealer or directly from ARRL. Mail orders to Pub Sales Dept, ARRL, 225 Main St, Newington, CT 06111-1494. You can call us toll-free at tel 888-277-5289; fax your order to 860-594-0303; or send e-mail to **pubsales@arrl.org**. Check out the full ARRL publications line on the World Wide Web at **http://www.arrl.org/catalog**.

[2] Amidon Inc, 3122 Alpine, Santa Ana, CA 92704; tel 714-850-4660, fax 714-850-1163.

[3] The detector load impedance must match that of the RLB, 50 Ω here.

[4] A discussion of improving diode-detector sensitivity appears under "RF Power Measurement" (pp 146-148) in ARRL's *Solid State Design for the Radio Amateur* by Wes Hayward, W7ZOI, and Doug DeMaw, W1FB (ARRL Order No. 0402). The 1997 *ARRL Handbook* Tandem Match circuit (pp 22.36 to 22.42) uses op amps to compensate for diode nonlinearity. John Grebenkemper's, KI6WX, *QST* article (Jan 1987, pp 18-26) fully describes the technique. For purchasing information, see Note 1.

[5] Make sure to use an accurate load for this measurement. If you use the same resistor as used for R2, the stray reactances will cancel, and erroneously increase the directivity measurement.—*Ed.*

*Jim was first licensed as WV6AXA in 1958, when he was 15. He upgraded to General a few months after getting his Novice "tickey" and remained a General operator until 1965, when he upgraded to Extra class. Ham radio started his career in electronics, and after completing his bachelor degree (a BS in electronics industrial technology from Cal State Long Beach), Jim's first job was in digital design. He had been a digital designer for almost 20 years with many of the local defense companies, when he had an opportunity to join an RF group. There was no trauma in the change because of Jim's Amateur Radio experience. After the defense cutbacks, Jim became an electronics instructor at a local community college. He finds teaching a rewarding experience. You can contact Jim by mail at 2415 College Dr, Costa Mesa, CA 92626 or by e-mail at* **jimford@netcom.com**

# New Products

## THE CELLMATE BATTERY CHARGER/CONDITIONER FROM VCI

◊ Designed to keep 6, 12 and 24 V lead-acid batteries fully charged and in peak condition, Voltage Control Industries' Cellmate is an intelligent charger and battery conditioner for marine, auto, RV and motorcycle batteries. When the Cellmate, powered by 117 V ac, has topped off your battery, an LED blinks to indicate a fully charged condition. The unit compensates for temperature effects and equalizes the voltage across all cells.

Unlike most trickle chargers, the Cellmate does not apply a constant charging current. This extends battery life and eliminates damage from overcharging. Other features include easy installation; automatic voltage switching; solar panel compatibility and a three-year warranty.

Prices: Cellmate 6/12 (conditions on 6 or 12 V battery) $37.95; Cellmate 12/24 (can monitor and charge up to four batteries at once) $87.95. For more information, contact Voltage Control Industries at PO Box 1555, Goleta, CA 93116; tel 805-683-9684, fax 805-967-0111.

# Cathode Ray Tubes (CRT)

A variation of the vacuum tube that is widely used in oscilloscopes and television monitors is the *cathode ray tube (CRT)*, diagrammed in **Fig 1**. The CRT has a cathode and grid much like a triode tube. The plate, usually referred to as the *anode* in this device, is designed to accelerate the electrons to very high velocities, with anode voltages that can be as high as tens of thousands of volts. The anode of the CRT differs from the plates of other vacuum tubes, since it is designed as a set of plates that are parallel to the electron beam. The anode voltage accelerates the electrons but does not absorb them. The electron beam passes by the anode and continues to the face of the tube. The cathode, grid and anode are all located in the neck of the CRT and are collectively referred to as the *electron gun*.

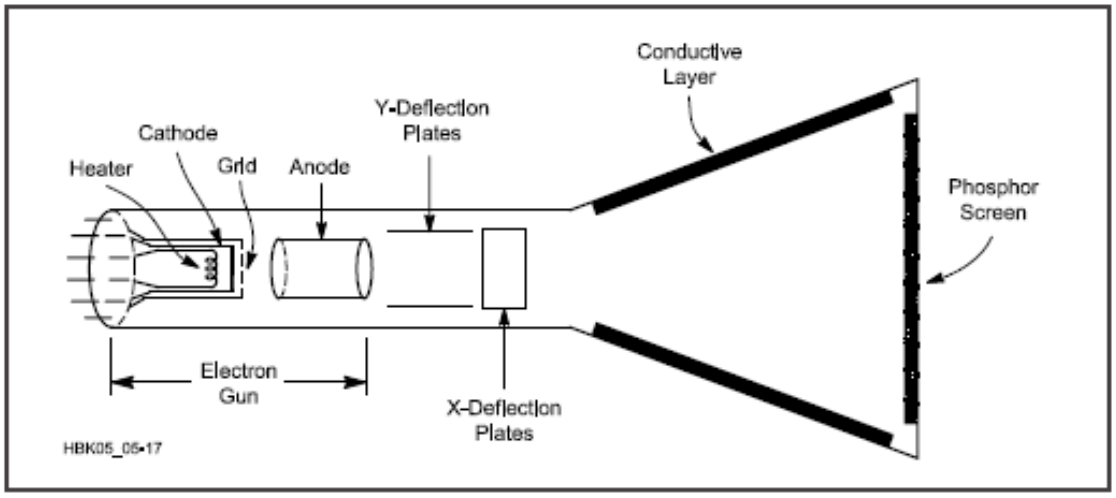



**Fig 1 — Cross section of CRT. The electron gun generates a stream of electrons and is made up of a heater, cathode, grid and anode (plate). The electron beam passes by two pairs of deflection plates that deviate the path of the beam in the vertical (y) direction and then the horizontal (x) direction. The deflected electron beam strikes a phosphor screen and causes it to glow at that spot. Any electrons that bounce off the screen are absorbed by the conductive layer along the sides of the tube, preventing spurious luminescence.**

The electron beam is deflected from its path by either magnetic deflectors that surround the yoke of the tube or by electrostatic deflection plates that are built into the tube neck just beyond the electron gun. A CRT typically has two sets of deflectors: vertical and horizontal. When a potential is applied to a set of deflectors, the passing electron beam is bent, altering its path. In an oscilloscope, the time base typically drives the horizontal deflectors and the input signal drives the vertical deflectors, although in many oscilloscopes it is possible to connect another input signal to the horizontal deflectors to obtain an X-Y, or vector, display. In televisions and some computer monitors the deflectors typically are driven by a raster generator. The horizontal deflectors are driven by a sawtooth pattern that causes the beam to move repeatedly from left to right and then retrace quickly to the left. The vertical deflectors are driven by a slower sawtooth pattern that causes the beam to move repeatedly from top to bottom and then retrace quickly to the top. The relative timing of the two sawtooth patterns is such that the beam scans from left to right, retraces to the left and then begins the next horizontal trace just below the previous one.

Beyond the deflectors, the CRT flares out. The front face is coated with a phosphorescent material that glows when struck by the electron beam. To prevent spurious phosphorescence, a conductive layer along the sides of the tube absorbs any electrons that reflect off the glass.

Vector displays have better resolution than raster scanning. The trace lines are clearer, which is the reason oscilloscope displays use this technique. It is faster to fill the screen using raster scanning, however. This is why TVs use raster scanning.

Some CRT tubes are designed with multiple electron beams. The beams are sometimes generated by different electron guns that are placed next to each other in the neck of the tube. They can also be generated by splitting the output of a single electron gun into two or more beams. Very high quality oscilloscopes use two electron beams to trace two input channels rather than the more common method of alternating a single beam between the two inputs. Color television tubes use three electron beams for the three primary colors (red, green and blue). Each beam is focused on only one of these colored phosphors, which are interleaved on the face of the tube. A metal shadow mask keeps the colors separate as the beams scan across the tube.

# PROJECT: A Compensated, Modular RF Voltmeter

This versatile and yet simple voltmeter measures levels from 100 mV to 300 V, from 30 MHz to audio frequencies, and is accurate to ±0.5 dB. The design allows for either a built-in analog meter or an externally connected DVM—or both! This portable, simple, and inexpensive project was originally featured in the March/April 2001 issue of *QEX*. Like the original article, the project described below includes several refinements implemented by its author, Sid Cooper, K2QHE that evolved from a suggestion by then *QEX* Managing Editor, Robert Schetgen, KU7G, to increase the voltage range of the instrument.

The requirements for this RF voltmeter (RFVM, **Fig 25.46**) are many:

- Provide accurate, stable measurements
- Measure voltage at frequencies from audio through HF
- Measure voltage levels from QRP to QRO
- Measure voltages inside equipment or in coax
- Operate portably or from ac lines
- Be flexible: work with digital voltmeters already in the shack, or independently
- Be inexpensive

## DESIGN APPROACH

To achieve stability, the RFVM uses op amps and any drift in the probe's diode detector is compensated by a matched diode in the RF op amp. This stretches the sensitivity to QRP levels. The dynamic range extends linearly from 0.1 V to 300 V (RMS) by using a series of compensated voltage multipliers.

The frequency response was flattened by using Schottky diodes, which easily reach from 60 Hz to 30 MHz. Both the basic probe and the multipliers very simply adapt to function as a probe, clip to various measurement points on the chassis or to screw onto UHF connectors. The active devices are only two op amps. They draw 800 µA of resting current and a maximum of 4.0 mA during operation from a 9-V battery, which is disconnected when a 9-V wall unit is plugged into a jack on the back panel. As designed, the RFVM has a small sloping panel cabinet with a microammeter display. A pair of front-panel banana OUTPUT jacks can be used with a DVM; this eliminates one op amp, the meter and a multiposition switch. Since the meter and switch are the most expensive parts (when they are not bought at a hamfest), how low-cost can this RF voltmeter get?

## THE PROBE

The low-voltage probe is a detector circuit that uses a Schottky diode and a high-impedance filter circuit (**Fig 25.45**). The diode is matched with the one in the feedback circuit of the CA3160 shown in **Fig 25.64**. This match reduces the diode's threshold voltage from about 0.34 V to less than 0.1 V, making the voltage drop comparable to that of a germanium diode. Since I think 0.1 V adequately covers QRP requirements, I made no further tests with germanium diodes to determine how much lower in voltage we could go.[1]

Use the low voltage probe and the output of the CA3160 in the meter unit (Fig 25.64) to find a matched diode pair. Build the CA3160 circuit first for this purpose. Select a diode and place it in the feedback loop of the CA3160, then test each of the remaining diodes in the probe with the three pots set about midrange. Test each diode, first at probe inputs of 100 mV, then at 3.00 V, at 400 Hz. Record the dc output from the CA3160, using the OUTPUT terminals. A bag of 20 diodes from Mouser Electronics[2] contained seven matched pairs, with identical readings at both low and high voltages. My tests used the ac scale (good to 500 Hz) of a Heath 2372 DVM to read the input voltage, and its dc scale to read the output. The number of matched pairs is surprising, but the diodes in the bag may all be from the same production run.

## RF PROBE ASSEMBLY

After selecting matched diodes, all components of the RF probe are mounted on a piece of perf board that is inserted in an aluminum tube (**Fig 25.47**). Since the circuit board is very light, it is supported on each end only by its input and output wires. The input RF wire is soldered to the center pin of the SO-239 connector. The two dc-output wires are held to the board by a small tie wrap, consisting of a piece of the #26 insulated wire, after having been passed through the plug cap and grommet. Cut a slit in the aluminum tube, perpendicular to its end. Make it at least ⅛-inch long and 1/32-inch wide to allow a bare

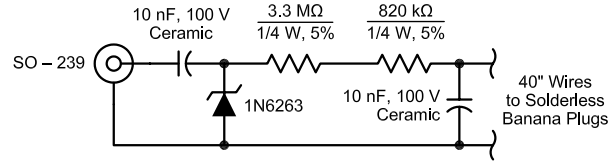



**Fig 25.45 — A schematic diagram of the RF probe.**

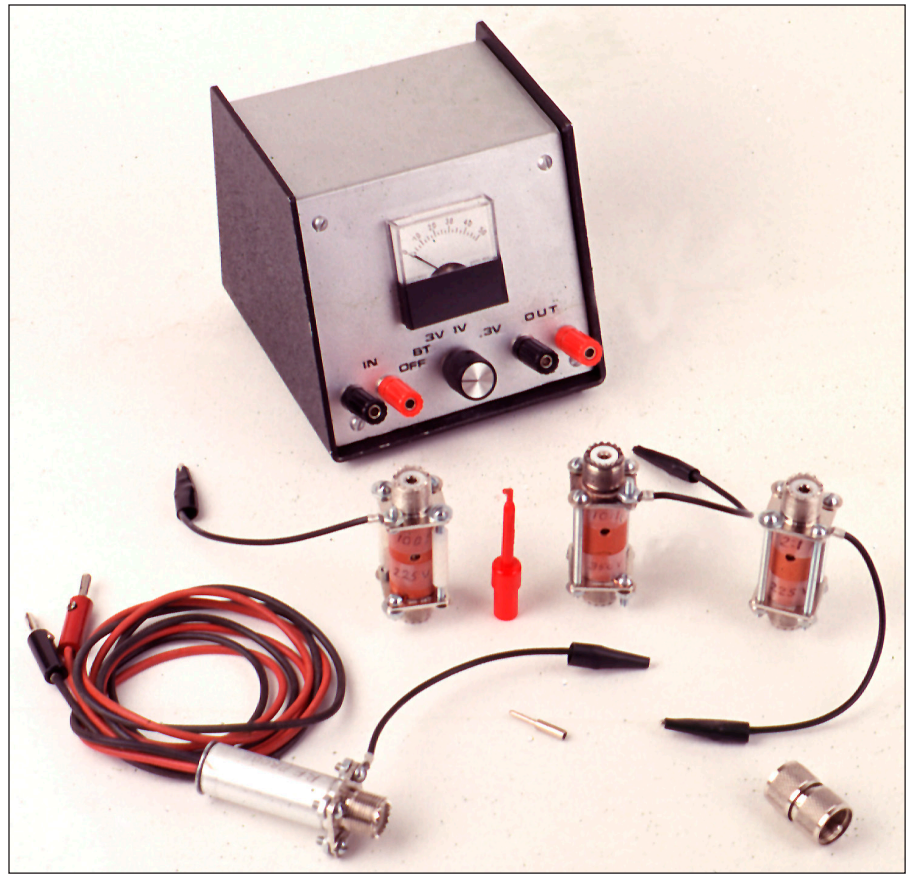



**Fig 25.46 — The RF voltmeter with meter unit, RF probe and voltage multipliers.**

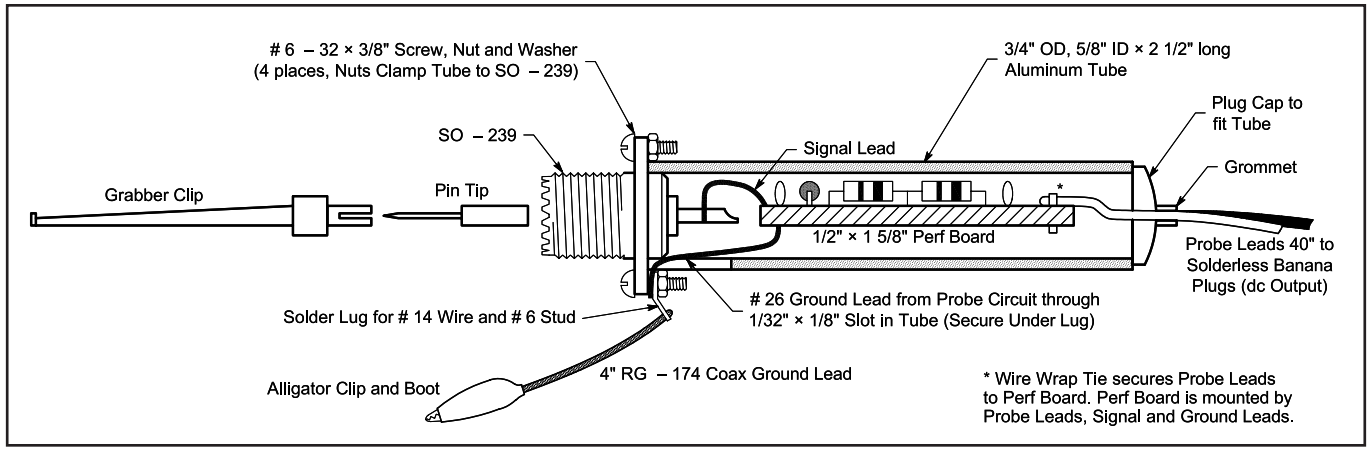



**Fig 25.47 — RF probe construction showing the grabber clip (RS 270-334), pin tip (Mouser 534-1600), lug (Mouser 571-34120), for #14-#16 wire and a #6 stud), alligator clip (RS 270-1545), plug cap (Mouser 534-7604), grommet (Mouser 5167-208) and 4-in ground lead made from shield of RG-174.**

#26 ground wire to pass through it.

Let's look at how the probe is assembled. The aluminum tube just fits around the back end of the SO-239 connector. When the four #6-32 screws, nuts and washers are installed in the four holes of the SO-239 connector, the flat face of the nut bears down strongly on the aluminum tube and rigidly holds it in place. Most standard SO-239 connector holes easily accept #6-32 screws; if yours do not, enlarge them with a #27 bit. The slit cut in the aluminum tube allows the bare ground wire from the perf board to pass from inside the tube to the outside and around the screw that secures the ground lead of the probe. As can be seen in Fig 25.47, the threaded part of the SO-239 connector faces away from the aluminum tube.

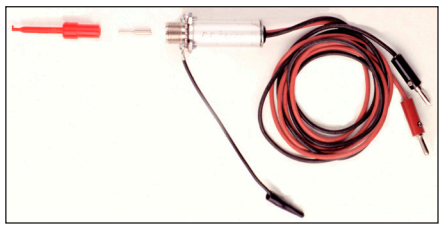


**Fig 25.48 — A photo of the grabber clip, pin tip and RF probe.**

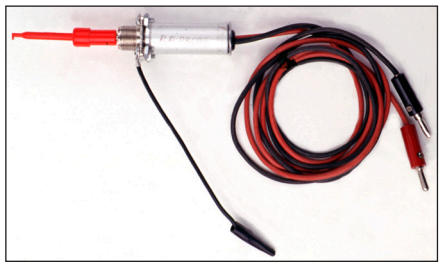


**Fig 25.49 — The RF probe with the pin tip and grabber clip in place. The pin tip is not visible, but it provides the physical bridge and electrical connection between the clip and SO-239 of the RF probe.**

This probe allows easy voltage measurements in a coaxial cable when the probe SO-239 is secured to a mating T connector. To make a probe reading at any point on a chassis or PC board with this connector, simply insert the larger end of a nicely mating pin tip into the SO-239. The pin tip's diameter is 0.08 inches at the small end and — fittingly — 0.14 inches at the mating end. I made this purely fortuitous discovery while rummaging through a very ancient junk box. Further, to make measurements with the probe clipped to a part or test point on a PC board or chassis, a clip or grabber can be mated to the pin tip that protrudes from the SO-239. This is shown in Figs 25.47 and **25.48**. All this may have been known to people in the connector industry, but it appears not to have been known or used elsewhere. **Fig 25.49** shows the grabber connected to the RF probe using the pin tip to join the two.

## PROBE MEASUREMENTS

The probe can accept RF signals up to 20 V (RMS) when there is no dc voltage or a combination of dc voltage and peak ac of about 28 V without exceeding the reverse-voltage rating of the diode. I made linearity measurements of the RF probe from 100 mV to 8.0 V at 400 Hz. (The range was limited by my available signal generator.) **Fig 25.50** is a plot of the error at the probe output versus the input amplitude. This was measured using a digital voltmeter with a 10-MΩ input resistance. This input resistance and the 4.1-MΩ resistor in the probe converts the peak voltage of a sine wave signal to the RMS reading of the DVM. The error is –44% at 100 mV, then 10% (0.9 dB) at 1.0 V and finally 2.9% at 8.0 V. The RF probe and DVM are obviously intended for higher voltage readings where the diode voltage drop doesn't affect the accuracy significantly.

The frequency response of the probe was measured at the midpoint of each ham band from 1.9 to 30 MHz with 650 mV input (see **Fig 25.51**). This time, because the input voltage was low and the absolute error would have been high, the probe was connected to the meter unit, which will be described later. The compensating diode loop of the meter unit reduces the error that would otherwise be read by the DVM from 90 mV (14%) to 4 mV (0.6%). The frequency response, however, is controlled by the probe. Use of the meter unit—which operates entirely on the dc signal—does not affect the frequency response, but it does improve the sensitivity of the readings. The measured response shown goes from –3.6% (–0.32 dB) to +1.2% (+0.1 dB) [A larger input capacitor in the probe will improve accuracy at lower frequencies. Pat McGuire, NO5O, reports good results from 50 kHz to 30 MHz by changing the input capacitor from 10 nF to 0.1 μF and performing the calibration described later at 50 kHz instead of 400 Hz. — Ed].

## MULTIPLIERS

To extend the voltage range of the RF probe, use a 10× multiplier, which is a compensated divider, shown in **Fig 25.52**. The electrical design is straightforward and includes a small trimmer capacitor to adjust for a flat frequency response. These components are very lightweight, so they require no perf board and are easily supported by their leads, which are anchored at the two SO-239 connectors. To make measurements, the input connector can be mated to connectors in a coaxial cable, a pin tip can be inserted for probing PC boards or a grabber can be added for connections to components on a chassis. The multiplier can connect to the RF probe

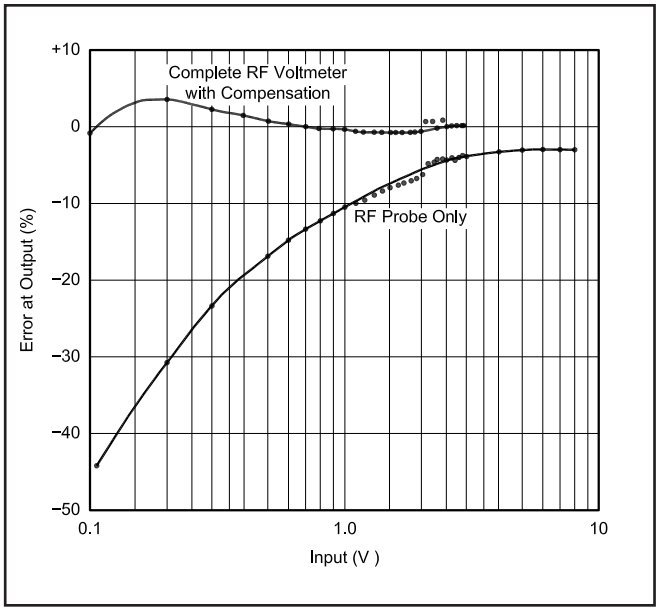



**Fig 25.50 — Amplitude linearity of the RF probe alone and in combination with the RF voltmeter.**

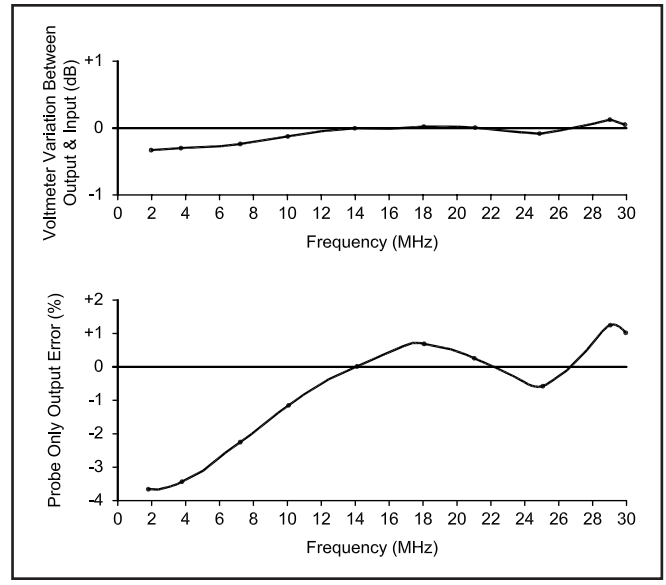



**Fig 25.51 — A frequency-response plot for the RF voltmeter. The response of the probe determines the overall system response.**

either through a male-to-male UHF fitting or by a piece of coax with a UHF male connector at each end.

The typical way to adjust the trimmer requires a square-wave input and an oscilloscope on the output. The trimmer is then set to produce a flat square-wave output with no overshoots on the leading edges and no droop across the top. An RF signal generator that covers 1.9 MHz to 30 MHz can be used instead, if a square wave generator is not available. Since this is a 10× multiplier, it is reasonable to expect the input to handle up to 200 V (RMS), but the limit is 150 V because the trimmer capacitor is rated at 225 V dc. You may be able to find small trimmers with higher voltage ratings that will take the multiplier to 200 V and still fit inside the tube housing.

Multiplier construction follows the same methods used for the RF probe but with a few differences. My first model used an aluminum tube like that in the RF probe with #6-32 screws to hold it all together, but the frequency response began to drop off at 21 MHz. By replacing the aluminum tube with a phenolic tube[3] having the same dimen-

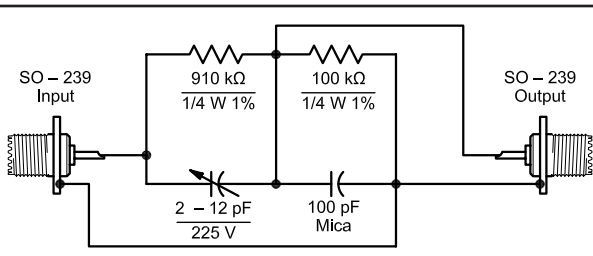



**Fig 25.52 — A schematic of the 10× multiplier. The trimmer capacitor is from Ocean State Electronics (see Note 10).**

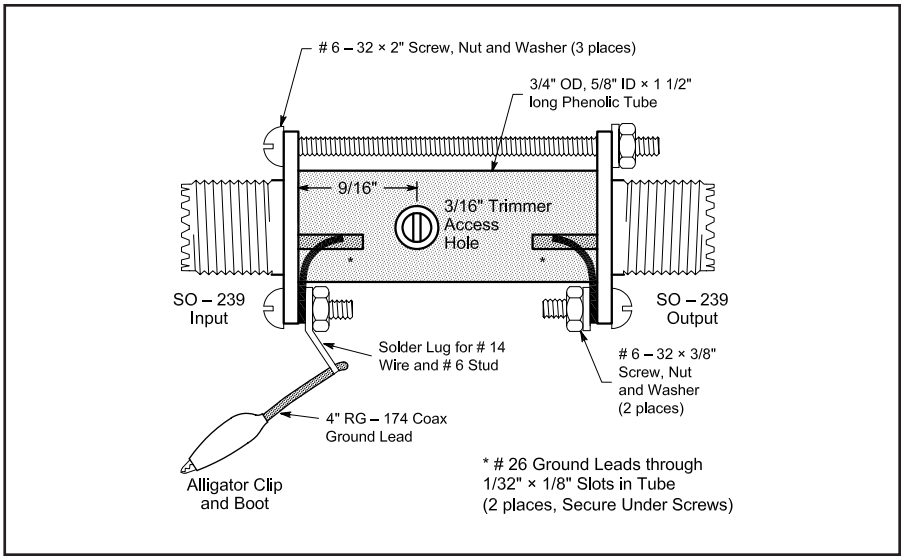



**Fig 25.53 — Construction of the 10x multiplier. The phenolic tube is US Plastics #47081, see Note 3. The lug is Mouser 571-34120.**

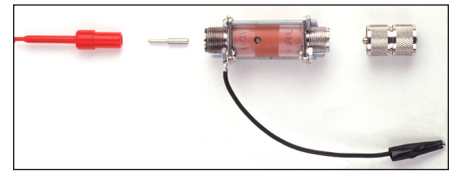


**Fig 25.54 — A view showing the 10× multiplier with grabber, pin tip and a male UHF coupler that connects the multiplier to the RF probe.**

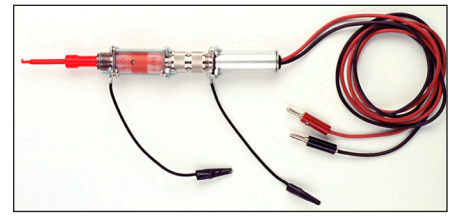


**Fig 25.55 — The complete RF probe and 10× multiplier assembly.**

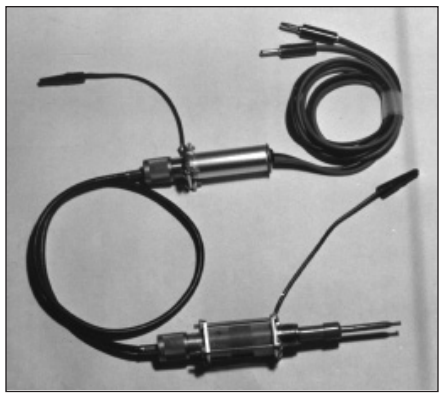


**Fig 25.56 — The RF probe and multiplier can be separated by a piece of coax less than 12 inches long if it makes measurements more convenient.**

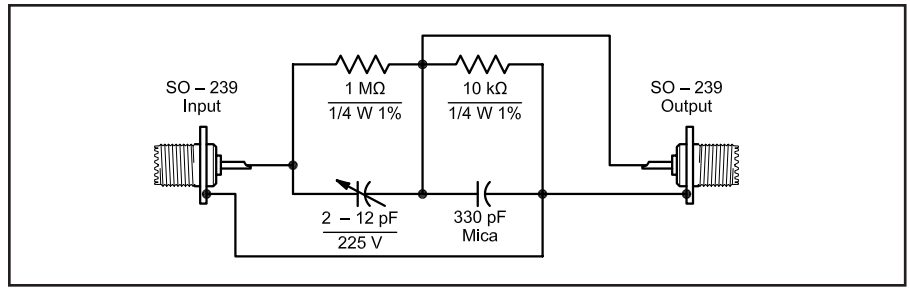



**Fig 25.57 — Schematic of the 100x multiplier (maximum allowable input is 150 V RMS). The trimmer capacitor is from Ocean State Electronics (see Note 10).**

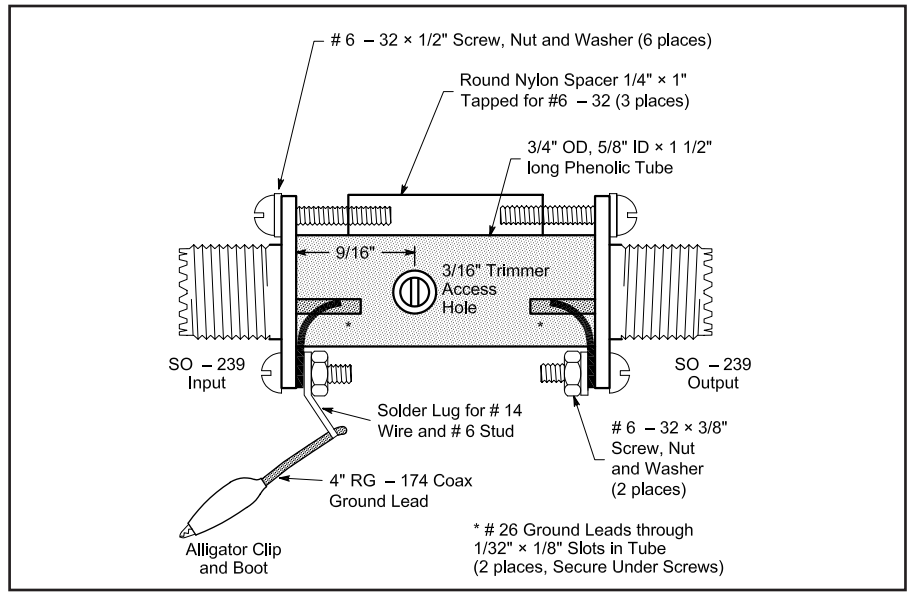



**Fig 25.58 — 100× multiplier assembly showing the round Nylon spacer (Mouser 561-TSP10), phenolic tube (US Plastics #47081, see Note 3). The lug is Mouser 571-34120.**

sions, the response is adequate to 30 MHz.

**Fig 25.53** shows construction methods. Two slits at the ends of the tube allow the ground wires at the input and output ends to exit and wrap around the screws of each connector. A hole in the phenolic tube allows adjustment of the trimmer capacitor.

**Fig 25.54** shows the exploded view of the pin tip and grabbers at the input end of the multiplier and a UHF male-to-male connector on the output side to mate with the RF probe. The assembled multiplier and probe are shown in **Fig 25.55**. The whole assembly is only 3½ inches long and is comfortable in the hand. If more flexibility is desired, the two pieces may be separated by as much as one foot of coax (**Fig 25.56**) without doing too much damage to the accuracy of the measurement.

The 100× multiplier follows the same design concepts as the 10× multiplier including the obligatory trimmer capacitor, as in **Fig 25.57**. Here too, the 225 V dc rating of the trimmer limits the maximum RF to 150 V RMS. So, this is not useful should it be used only with the probe and your DVM because the 10× multiplier already covers this range. Later, we will see how it is useful with the RF probe and the meter unit.

The 100× unit also uses a phenolic-tube housing with a hole to adjust the trimmer and a slit at each end to bring out the ground wires, which then wrap around screws in the connectors. In order to reach at least 30 MHz, however, the two-inch steel screws between the two SO-239 connectors must be insulated. Using plastic washers at the connector holes could do this job but would have required #4-40× 2-inch-long screws. After a long search at the biggest hardware stores and catalogs, I found nothing longer than 1½ inches. That's the best way to assemble this multiplier, if you can find the screws. Otherwise, use nylon spacers and shorter screws as shown in **Figs 25.58** and **25.59**. The pin tip and grabber

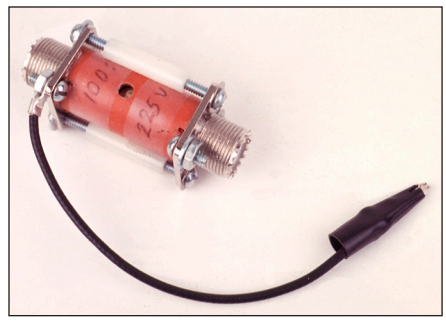


**Fig 25.59 — The assembled 100× multiplier.**

at the input and UHF male-to-male connector or coaxial cable at the output are used as with the 10× multiplier.

The only purpose of the 2× multiplier is to create a 200× multiplier when used with the 100× multiplier in series, so as to extend the voltage range from 150 V to 300 V (1800 W at 50 Ω only under matched conditions).[4] The trimmer in each multiplier now divides in half the input voltage. **Fig 25.60** shows the 2× multiplier with only one RC section since it relies on the RC sections in the 100× multiplier to complete the division (multiplication).

**Table 25.4**
**Multiplier Range versus Capacitance**

| Multiplier | 10× | 100× | 200× |
|---|---|---|---|
| Input Cap (pF) | 13.3 | 10.9 | 13.5 |

The construction, **Fig 25.61**, returns to the tubular format as before, but this time there are no slots to pass the ground wires; the four screws are able to do the job. The two multipliers are shown in series in **Fig 25.62**. The input capacitance for each multiplier is shown in **Table 25.4**.

When measurements are made on a load resistance of 50 Ω, the input capacity of the multipliers has no affect on readings below 30 MHz. When the impedance of the load resistance is larger, consider any error it introduces. To get a sense of the affects of the load resistance, use the equation below.

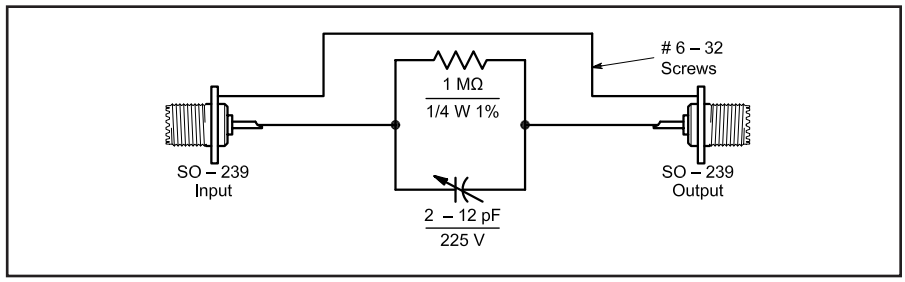



**Fig 25.60 — A schematic diagram of the 2× multiplier. Its output connects to the input of the 100× multiplier for 200x measurements up to 300 V RMS. This multiplier is used *only* with the 100× multiplier.**

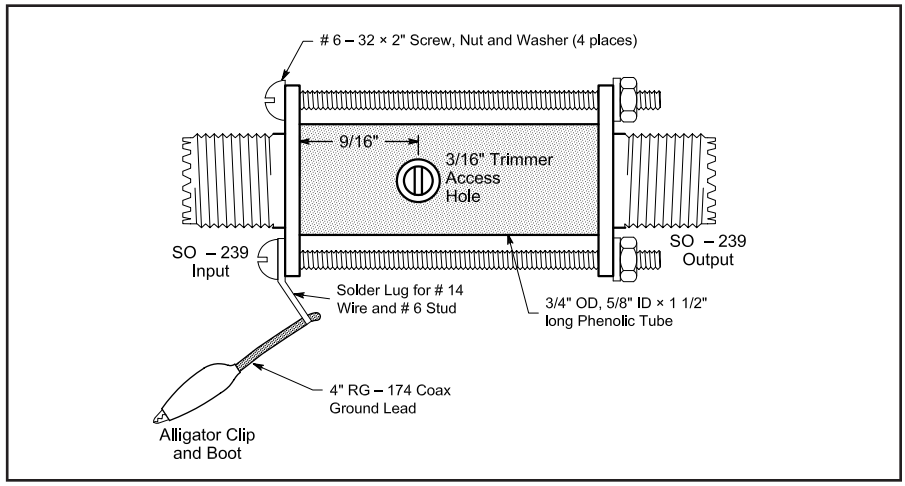



**Fig 25.61 — 2× multiplier assembly showing the phenolic tube (US Plastics #47081, see Note 3). The lug is 16-14 #6 stud (Mouser 571-34120).**

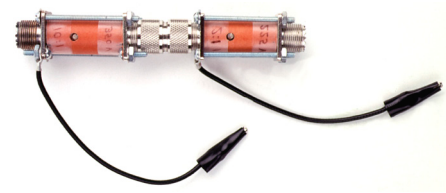


**Fig 25.62 — The 2x and 100x multipliers are joined by a male UHF coupler for readings up to 300 V (RMS, 1800 W, see Note 4).**

It shows the relationship between the resistance, capacitance and frequency when the measured voltage is 3 dB down from what it would be if it were read by a meter with no input capacity:

$$f = \frac{1}{2\pi R_L C_P} \qquad (21)$$

where

f = 3-dB loss frequency

$R_L$=load resistance

$C_P$=probe capacitance

For example, at a load of 1 kΩ and a probe capacity of 13.3 pF, the 3-dB down frequency is about 12 MHz, where the error is 30%. If an attenuator were used instead of the multiplier, it would not have this problem because it works with either a fixed load of 50 Ω at high frequencies or 600 Ω at low frequencies. Probes cannot select their frequency or load impedance and are thereby more flexible in use, so measurements must be made with consideration, but these multipliers would not have a problem at 50 Ω or 600 Ω either.

**Table 25.5** summarizes the voltage ranges using only the multiplier with the RF probe and a DVM that has a 10-MΩ input resistance that's available in the shack.

The multipliers are intended for use at high voltages where safety precautions are a primary consideration to avoid personal injury. See the **Safety** chapter of this *Handbook*. In Tektronics' *ABC's of Probes,*[5] an entire section thoroughly covers the hazards and necessary precautions when making measurements with probes. It is worth the little effort to get a copy of it and also the Pomona Catalog,[6] which has some good information on probe use.

The multipliers have panel connectors at both ends and the RF probe has one at one end. Their grounds are connected and brought to the meter unit. This unit is grounded only when the ac-powered 9-V power supply that may be used with it, is grounded. When it is battery operated, the meter unit relies on the ground-clip connection to the ground of the equipment under test. This is satisfactory if there is no unknown break in the chain of ground connections that would make the panel connector hot. Furthermore, as insurance, it is good to wrap these connectors with vinyl tape or to cover the multiplier with a plastic boot to prevent contact with either hands or equipment under test. This is not shown in any photographs because it would have obscured the appearance and construction of the probe and multipliers.

The frequency responses of the multipliers were determined using an RF signal generator set at the midpoint of each ham band from 160 to 10 meters, including 30 MHz. An error of 5.42% (0.46 dB) was measured from 160 to 17 meters, which then decreased to 4.13% (0.35 dB) at 15 meters and then to zero through 30 MHz. The constant error from 160 to 17 meters is probably due to the inherent errors in the test equipment. The oscilloscope has a 60 MHz bandwidth, an input impedance of 1 MΩ and 30 pF, which introduces a load effect on the multiplier outputs. It also has a reading accuracy of ±3%. When the multiplier-components accuracy of 1% is included, it is not surprising to find the overall inaccuracy to be at most 5.42%.[7] The error of the multiplier itself could be inherently less than this. Before the overall frequency response

**Table 25.5**

**Multiplier versus Voltage Range**

| *Probe Configuration* | *Range (V)* | *Maximum Voltage* |
|---|---|---|
| Probe Alone* | 1.0-20 V ac | 20 V ac |
| With 10× Multiplier* | 10-150 V ac | 225 V (ac + dc) |
| With 100× and 2× Multiplier† | 150-200 V ac | 450 V (ac + dc) |
| With 100× and 2× Multiplier†† | 200-300 V ac | 450 V (ac + dc) |

*error less than –10% or –0.83 dB

†with error of –45% to –10% or less than 3.2 dB. A trimmer capacitor with a rating of 300 V dc increases the 10× range to 200 V ac and reduces the error from 45% to 10%. The RF probe and meter unit reduces the error to less than –3.5% or less than –0.3 dB.

††with error less than –6% or –0.54 dB

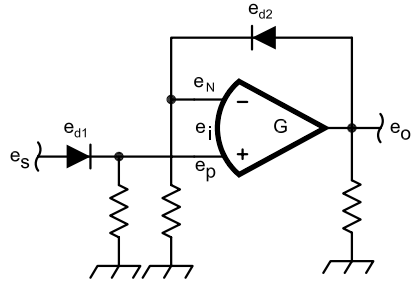
$$e_s - e_{d1} = e_p$$

$$e_o - e_{d2} = e_n$$

$$e_i = e_p - e_n = e_s - e_{d1} - e_o + e_{d2}$$

$$e_i = \frac{e_o}{G} = 0$$

$$e_s - e_o = e_{d1} - e_{d2}$$




$e_s$ represents the input of the RF probe as a dc signal.
$e_o$ represents the output from the high-gain CA3160 op amp.
$e_{d1}$ is the voltage drop across the diode in the RF probe.
$e_{d2}$ is the voltage drop across the diode in the feedback loop.
G = 320,000 for the CA3160.
The last equation shows that any difference between the signal and output results from unequal diode drops when the diodes are not matched.

**Fig 25.63 — Basics of the diode-compensation method to improve measurement accuracy at low input voltages. It also provides a measure of temperature compensation and drift reduction. See Fig 25.50.**

was measured, the trimmer capacitor was adjusted with the input frequency set mid-frequency at 15 MHz. If your interest in RF probes and multipliers has been raised and you have more questions, see the references in Notes 5 and 6.

## THE METER UNIT

The meter unit serves several purposes when used with the RF probe. It increases the accuracy of the probe and DVM from –45% to –3.5% at a voltage of 100 mV RF. At higher voltages, it maintains a minimum advantage of 5:1 in reducing the error, when the probe is used with a DVM. This is an increase in both sensitivity and accuracy. The usual non-linearity caused by the diode in the probe is reduced when used with the meter unit. The technique also incidentally provides temperature compensation for diode drift. The meter unit has a ±5% panel meter to display measurements, but it also contains a pair of OUTPUT terminals for a DVM. Add a DVM when more accuracy is desired at the low end of the range or when you want a bit more resolution. Finally, since the meter unit is fully portable, low-level field-strength measurements are possible with a whip antenna at the probe input connector. Due to its sensitivity and accuracy, the probe can be adapted for use in many places around the shack.

The non-linear response of the probe diode is compensated (improved) by a circuit in the meter unit. The feedback loop of a CA3160 op amp contains a diode matched to the one in the probe (see Note 1). **Fig 25.63** and its sequence of equations present a very simple sketch of how matched diodes do this when dc is applied through a diode. The final equation shows that any difference between the op amp input and output is due to a difference in voltage drops across the two diodes. When the diodes are matched, the error disappears. When RF is applied, the average currents through the diodes must be equal to keep the voltage drops equal. Articles by Kuzdrall (Note 1) Grebenkemper[8] and Lewallen[9] are first-class descriptions of the principles used in this RF voltmeter.

Fig 25.64 shows three pots for calibrating the meter unit. This should be performed at 400 Hz to avoid any effects due to RF. The 100-kΩ pot is typically used to

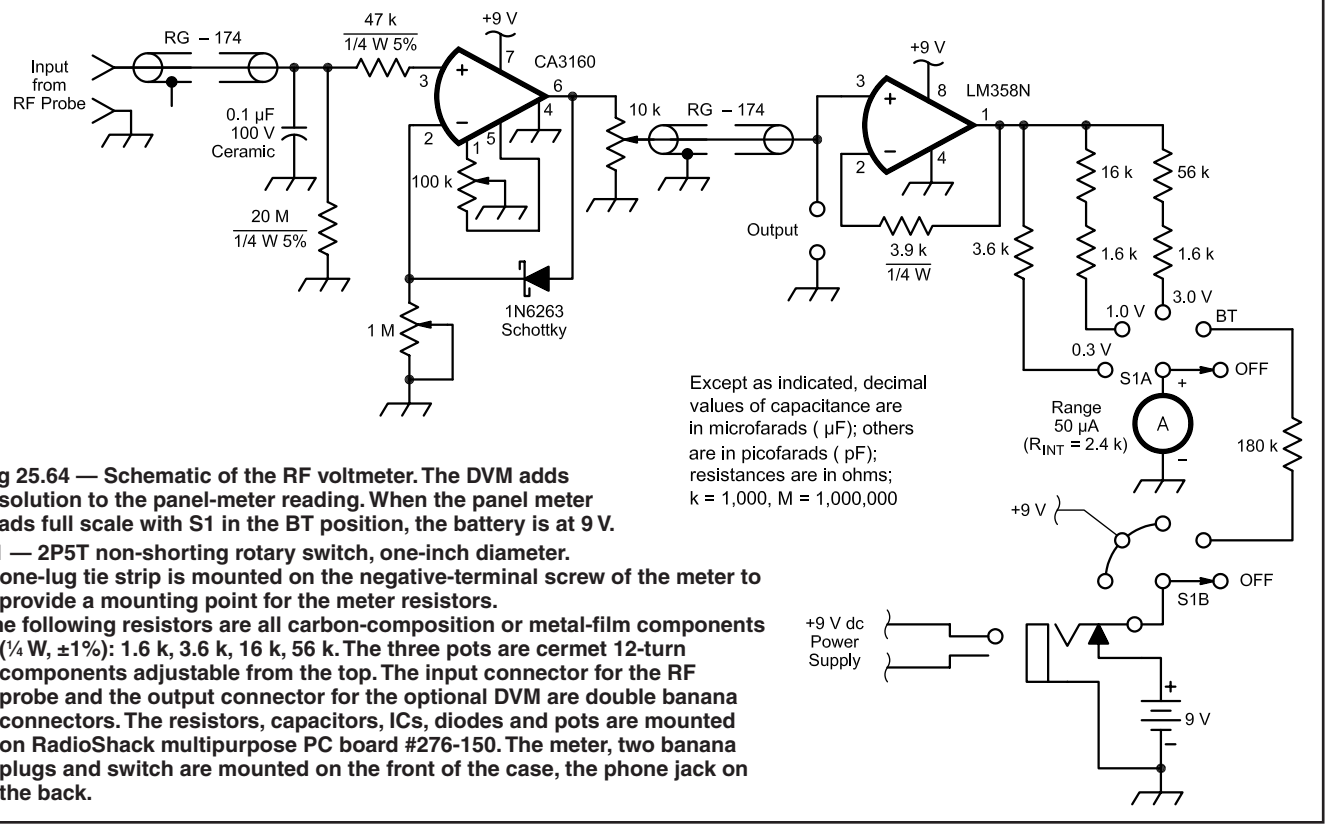



**Fig 25.64 — Schematic of the RF voltmeter. The DVM adds resolution to the panel-meter reading. When the panel meter reads full scale with S1 in the BT position, the battery is at 9 V.**
**S1 — 2P5T non-shorting rotary switch, one-inch diameter.**
**A one-lug tie strip is mounted on the negative-terminal screw of the meter to provide a mounting point for the meter resistors.**
**The following resistors are all carbon-composition or metal-film components (¼ W, ±1%): 1.6 k, 3.6 k, 16 k, 56 k. The three pots are cermet 12-turn components adjustable from the top. The input connector for the RF probe and the output connector for the optional DVM are double banana connectors. The resistors, capacitors, ICs, diodes and pots are mounted on RadioShack multipurpose PC board #276-150. The meter, two banana plugs and switch are mounted on the front of the case, the phone jack on the back.**

null the offset of the CA3160, but is used here to initially set the offset to about 0.5 mV to 1.0 mV with 100 mV input. Then the 1-MΩ pot sets the output to 100 mV with 100 mV input, and the 10 kΩ pot sets the output to 3.0 V when the input is 3.0 V. Finally, the three pots are alternately adjusted until the 100 mV and 3.0 V set points occur together. The 100-kΩ pot is helpful in fine tuning the 100-mV point. A DVM was used at the optional OUTPUT sockets during calibration.

A CA3160 was selected for the input op amp because of its high input impedance of 1.5 TΩ and high gain of 320,000. Although it has diodes that provide protection, I think it could be sensitive to electrostatic discharge, so handle it carefully. Use IC sockets to permit easy replacement of the ICs in case of damage. An LM358N IC follows the CA3160 (see Fig 25.64), primarily to drive the panel meter. If you use a DVM as the display, you can omit the LM358N circuit, meter and multiposition switch. The only functions lost are the battery-voltage check and power on/off switch. Add an on/off switch to the circuit when the multi-position switch is omitted. The CA3160 op amp circuit easily spans the range from 100 mV to 3.0 V without the need for a range switch.

## CONSTRUCTION

A 5×5×4½-inch sloping-front instrument case was used to house the components of the meter unit. The meter on the front panel has a 2×2-inch face and requires a 1½-inch hole in the panel. Although a 50-μA meter from the junk box is used here, a 1-mA movement will work as well, provided the series resistors are changed accordingly.

The rotary switch has two poles and five positions for changing the meter range, testing the battery condition and switching the power off. Two sets of double banana binding posts are used, the INPUT pair accepts the dc signal from the RF probe. The OUTPUT pair provides a voltage for a DVM display, whether the panel meter is used or not. On the rear of the case, a miniature phone jack accepts 9-V power from either a battery or a 9-V dc supply. The ICs and other parts are mounted on a RadioShack multipurpose PC board that has very convenient holes and traces. The board is bolted to the back of the case via standoff insulators and wired to the front panel components (see **Fig 25.65**).

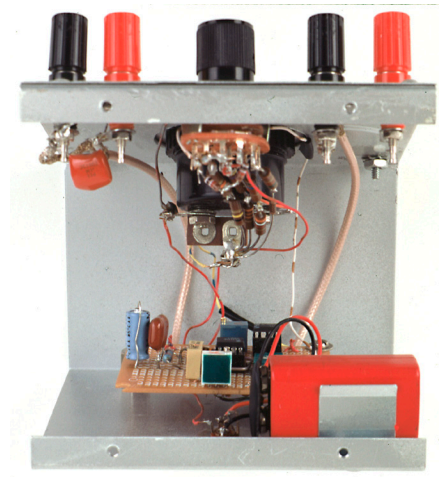


**Fig 25.65 — Inside view of the meter unit. RG-174 connects the INPUT and OUTPUT banana binding posts to the circuit board. The rotary switch, one-lug tie strip and battery are also visible.**

**Notes**

[1]J. A. Kuzdrall, "Linearized RF Detector Spans 50-to-1 Range," Analog Applications Issue, *Electronic Design*, June 27, 1994.

[2]Mouser Electronics, 2401 Hwy 287 N, Mansfield, TX 76063; tel 800-346-6873, fax 817-483-0931; E-mail **sales@mouser.com**; **www.mouser.com**.

[3]United States Plastics Corp, 1390 Neubrecht Road, Lima, OH 45801-3196, tel 419-228-2242, fax 419-228-5034.

[4]Load mismatch changes the measured voltage depending on the phase angle of the load impedance. In the worst-case, the mismatch multiplies the voltage by the square root of the SWR. With 1800 W, a 2:1 SWR (100-ohm load) increases the voltage from 300 to 424 $V_{RMS}$ and a 3:1 SWR (150-ohm load) would give 520 $V_{RMS}$.

[5]*ABCs of Probes*, Tektronix Inc, Literature number 60W-6053-7, July 1998. Tektronix, Inc. Export Sales, PO Box 500 M/S 50-255 Beaverton, OR 97707-0001; 503-627-6877. Johnny Parham, "How to Select the Proper Probe," Electronic Products, July 1997.

[6]Pomona Test and Measurement Accessories catalog. ITT Pomona Electronics, 1500 E Ninth St, Pomona, CA 91766-3835.

[7]A. Frost, "Are You Measuring Your Circuit or Your Scope Probe?" *EDN*, July 22, 1999. E. Feign, "High-Frequency Probes Drive 50-Ω Measurements," *RF Design*, Oct 1998.

[8]J. Grebenkemper, KI6WX, "The Tandem Match — An Accurate Directional Wattmeter," *QST*, Jan 1987, pp 18-26; and "Tandem Match Corrections," *QST*, Jan 1988, p 49.

[9]R. Lewallen, W7EL, "A Simplified and Accurate QRP Directional Wattmeter," *QST*, Feb 1990, pp 19-23, 36.

[10]Ocean State Electronics, 6 Industrial Dr, PO Box 1458, Westerly, RI 02891; tel 800-866-6626, fax 401-596-3590.

# A Compensated, Modular RF Voltmeter

*Build this voltmeter—it measures voltages from 100 mV to 300 V, from 30 MHz down to audio, on chassis or in coax, accurate to ±0.5 dB, with built-in meter or with your DVM. It's portable, simple and inexpensive.*

By Sid Cooper, K2QHE

This project was started to replace a reconditioned commercial RF voltmeter that required calibration every six months. It was quite accurate when it was in calibration, but was expensive to recalibrate. The idea then was to build a stable, solid-state RF voltmeter in less than six months so that the commercial one could then calibrate the homebrew unit, but—it has long been known what happens to your best-laid plans.

The requirements for my RF voltmeter (RFVM, Fig 1) are many:

- Provide accurate, stable measurements
- Measure voltage at frequencies from audio through HF
- Measure voltage levels from QRP to QRO
- Measure voltages inside equipment or in coax
- Operate portably or from ac lines
- Be flexible: work with digital voltmeters already in the shack, or independently
- Be inexpensive

This last goal is an old fashioned idea; most equipment today is not in that category.

5 Belaire Dr
Roseland, NJ 07068-1220

## Design Approach

To achieve stability, the RFVM uses op amps and any drift in the probe's diode detector is compensated by a matched diode in the RF op amp. This stretches the sensitivity to QRP levels. The dynamic range extends linearly from 0.1 V to 300 V (RMS) by using a series of compensated voltage multipliers.

The frequency response was flattened by using Schottky diodes, which easily reach from 60 Hz to 30 MHz. Both the basic probe and the multipliers very simply adapt to function as a probe, clip to various measurement points on chassis or to screw onto UHF connectors. The active devices are only two op amps. They draw 800 µA of resting current and a maximum of 4.0 mA during operation from a 9-V battery, which is disconnected when a 9-V wall unit is plugged into a jack on the back panel. As designed, the RFVM has a small sloping panel cabinet with a microammeter display. A pair of front-panel banana **OUTPUT** jacks can be used with a DVM; this eliminates one op amp, the meter and

a multiposition switch. Since the meter and switch are the most expensive parts (when they are not bought at a hamfest), how low-cost can this RF voltmeter get?

*The Probe*

The low-voltage probe is a detector circuit that uses a Schottky diode and a high-impedance filter circuit (Fig 2). The diode is matched with the one in the feedback circuit of the CA3160 shown in Fig 20. This match reduces the diode's threshold voltage from about 0.34 V to less than 0.1 V, making the voltage drop comparable to that of a germanium diode. Since I think 0.1 V adequately covers QRP requirements, I made no further tests with germanium diodes to determine how much lower in voltage we could go.[1]

Use the low voltage probe and the output of the CA3160 in the meter unit (Fig 20) to find a matched diode pair. Build the CA3160 circuit first for this purpose. Select a diode and place it in the feedback loop of the CA3160, then test each of the remaining diodes in the probe with the three pots set about midrange. Test each diode, first at probe inputs of 100 mV, then at 3.00 V, at 400 Hz. Record the dc output from the CA3160, using the **OUTPUT** terminals. A bag of 20 diodes from Mouser Electronics[2] contained seven matched pairs, with identical readings at both low and high voltages. My tests used the ac scale (good to 500 Hz) of a Heath 2372 DVM to read the input voltage, and its dc scale to read the output. The number of matched pairs is surprising, but the diodes in the bag may all be from the same production run.

[1]Notes appear on page 34.




**Fig 1—The RF voltmeter with meter unit, RF probe and voltage multipliers.**

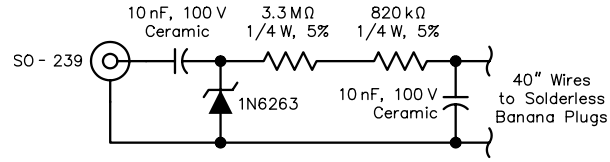



**Fig 2—A schematic diagram of the RF probe.**

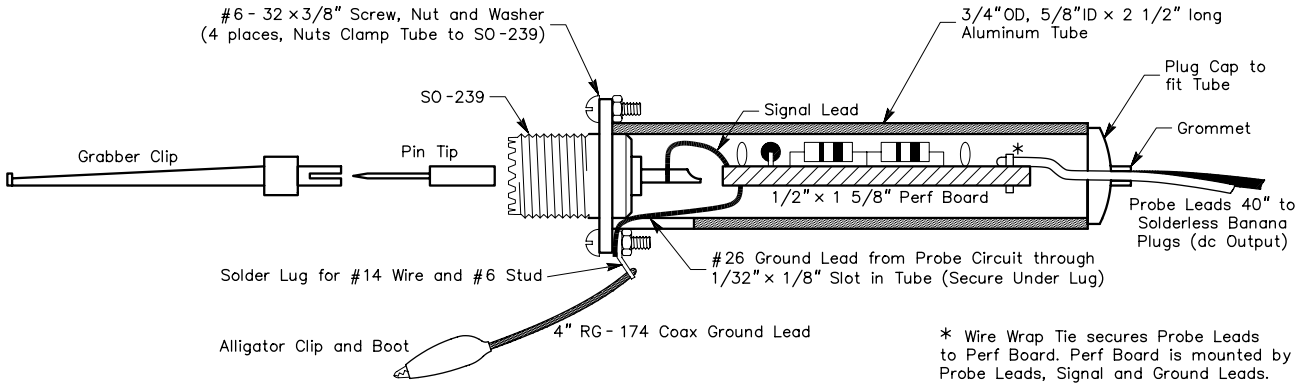



**Fig 3—RF probe construction showing the grabber clip (RS 270-334), pin tip (Mouser 534-1600), lug (Mouser 571-34120, for #14-#16 wire and a #6 stud), alligator clip (RS 270-1545), plug cap (Mouser 534-7604), grommet (Mouser 5167-208) and four-inch ground lead made from shield of RG-174.**

*RF Probe Assembly*

After selecting matched diodes, all components of the RF probe are mounted on a piece of perf board that is inserted in an aluminum tube (Fig 3). Since the circuit board is very light, it is supported on each end only by its input and output wires. The input RF wire is soldered to the center pin of the SO-239 connector. The two dc-output wires are held to the board by a small tie wrap, consisting of a piece of the #26 insulated wire, after having been passed through the plug cap and grommet. Cut a slit in the aluminum tube, perpendicular to its end. Make it at least 1/8-inch long and 1/32-inch wide to allow a bare #26 ground wire to pass through it.

Let's look at how the probe is assembled. The aluminum tube just fits around the back end of the SO-239 connector. When the four #6-32 screws, nuts and washers are installed in the four holes of the SO-239 connector, the flat face of the nut bears down strongly on the aluminum tube and rigidly holds it in place. Most standard SO-239 connector holes easily accept #6-32 screws; if yours do not, enlarge them with a #27 bit. The slit cut in the aluminum tube allows the bare ground wire from the perf board to pass from inside the tube to the outside and around the screw that secures the ground lead of the probe. As can be seen in Fig 3, the threaded part of the SO-239 connector faces away from the aluminum tube.

This probe allows easy voltage measurements in a coaxial cable when the probe SO-239 is secured to a mating T connector. To make a probe reading at any point on a chassis or PC board with this connector, simply insert the larger end of a nicely mating pin tip into the SO-239. The pin tip's diameter is 0.08 inches at the small end and—fittingly—0.14 inches at the mating end. I made this purely fortuitous discovery while rummaging through a very ancient junk box. Further, to make measurements with the probe clipped to a part or test point on a PC board or chassis, a clip or grabber can be mated to the pin tip that protrudes from the SO-239. This is shown in Figs 3 and 4. This was another lucky strike. All this may have been known to people in the connector industry, but it appears not to have been known or used elsewhere. Fig 5 shows the grabber connected to the RF probe using the pin tip to join the two.

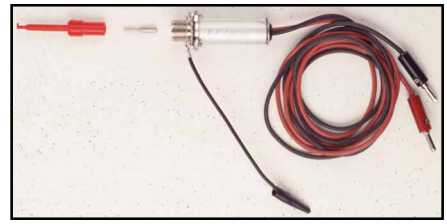


**Fig 4—A photo of the grabber clip, pin tip and RF probe.**

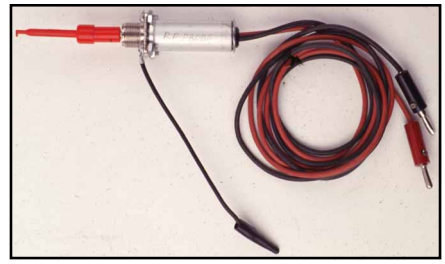


**Fig 5—The RF probe with the pin tip and grabber clip in place. The pin tip is not visible, but it provides the physical bridge and electrical connection between the clip and SO-239 of the RF probe.**

## Probe Measurements

The probe can accept RF signals up to 20 V (RMS) when there is no dc voltage or a combination of dc voltage and peak ac of about 28 V without exceeding the reverse-voltage rating of the diode. I made linearity measurements of the RF probe from 100 mV to 8.0 V at 400 Hz. (The range was limited by my available signal generator.) Fig 6 is a plot of the error at the probe output versus the input amplitude. This was measured using a digital voltmeter with a 10-MΩ input resistance. This input resistance and the 4.1-MΩ resistor in the probe converts the peak voltage of a sine wave signal to the RMS reading of the DVM. The error is –44% at 100 mV, then 10% (0.9 dB) at 1.0 V and finally 2.9% at 8.0 V. The RF probe and DVM are obviously intended for higher voltage readings where the diode voltage drop doesn't affect the accuracy significantly.

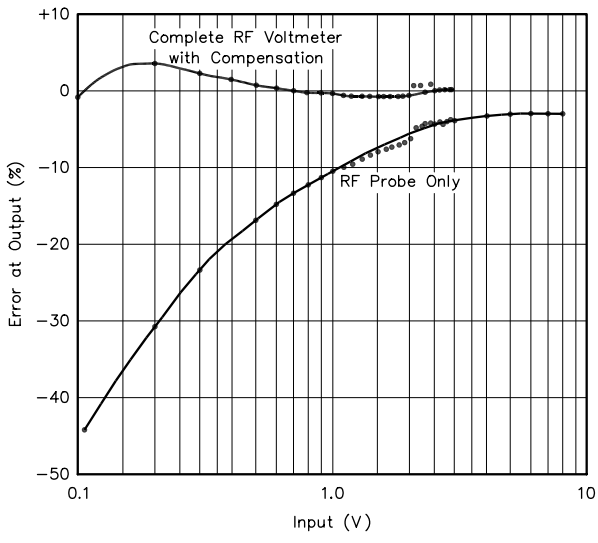



**Fig 6—Amplitude linearity of the RF probe alone and in combination with the RF voltmeter.**

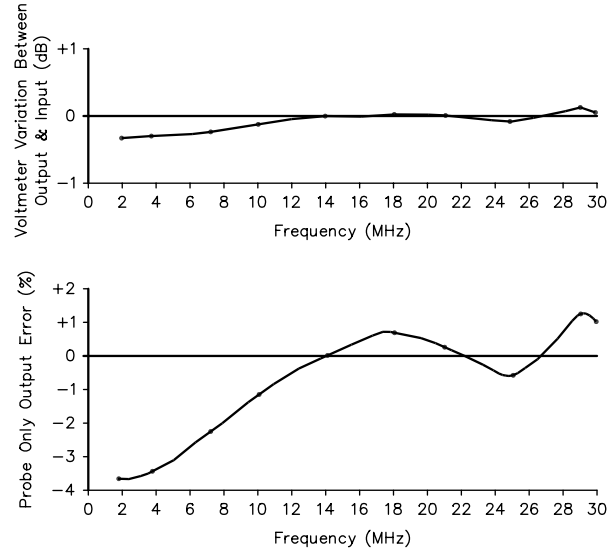



**Fig 7—A frequency-response plot for the RF voltmeter. The response of the probe determines the overall system response.**

The frequency response of the probe was measured at the midpoint of each ham band from 1.9 to 30 MHz with 650 mV input (see Fig 7). This time, because the input voltage was low and the absolute error would have been high, the probe was connected to the meter unit, which will be described later. The compensating diode loop of the meter unit reduces the error that would otherwise be read by the DVM from 90 mV (14%) to 4 mV (0.6%). The frequency response, however, is controlled by the probe. Use of the meter unit—which operates entirely on the dc signal—does not affect the frequency response, but it does improve the sensitivity of the readings. The measured response shown goes from –3.6% (–0.32 dB) to +1.2% (+0.1 dB).

## Multipliers

To extend the voltage range of the RF probe, use a 10× multiplier, which is a compensated divider, shown in Fig 8. The electrical design is straightforward and includes a small trimmer capacitor to adjust for a flat frequency response. These components are very lightweight, so they require no perf board and are easily supported by their leads, which are anchored at the two SO-239 connectors. To make measurements, the input connector can be mated to connectors in a coaxial cable, a pin tip can be inserted for probing PC boards or a grabber can be added for connections to components on a chassis. The multiplier can connect to the RF probe either through a male-to-male UHF fitting or by a piece of coax with a UHF male connector at each end.

The typical way to adjust the trimmer requires a square-wave input and

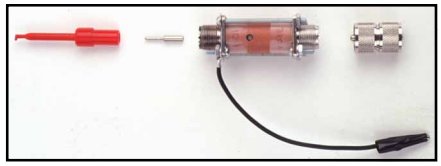


**Fig 10—A view showing the 10× multiplier with grabber, pin tip and a male UHF coupler that connects the multiplier to the RF probe.**

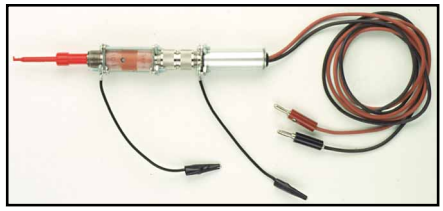


**Fig 11—The complete RF probe and 10× multiplier assembly.**

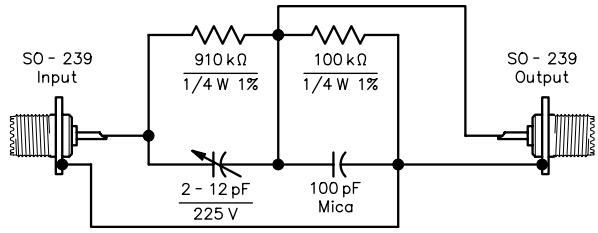



**Fig 8—A schematic of the 10× multiplier. The trimmer capacitor is from Ocean State Electronics (see Note 10).**

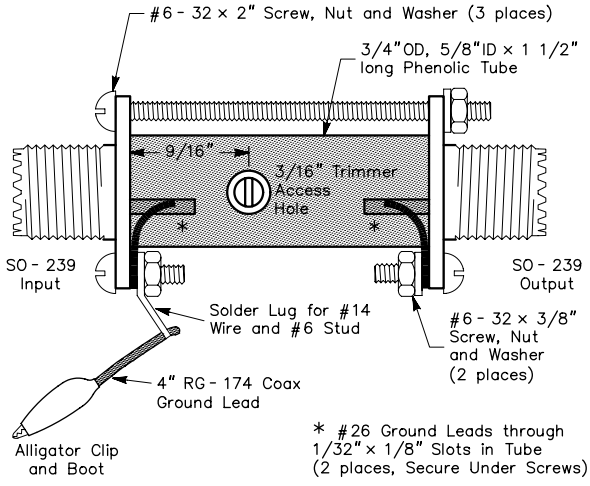



**Fig 9—Construction of the 10× multiplier. The phenolic tube is US Plastics #47081, see Note 3. The lug is Mouser 571-34120.**

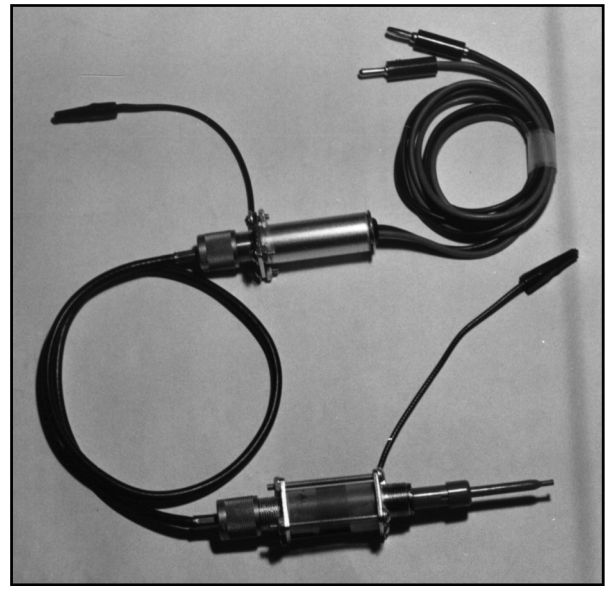


**Fig 12—The RF probe and multiplier can be separated by a piece of coax less that 12 inches long if it makes measurements more convenient.**

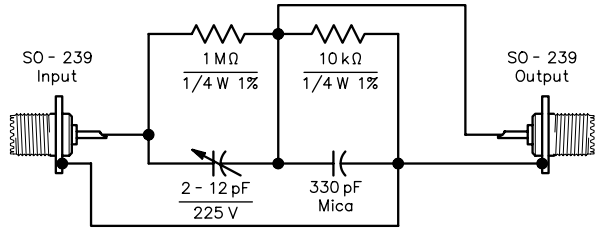



**Fig 13—Schematic of the 100× multiplier (maximum allowable input is 150 V RMS). The trimmer capacitor is from Ocean State Electronics (see Note 10).**

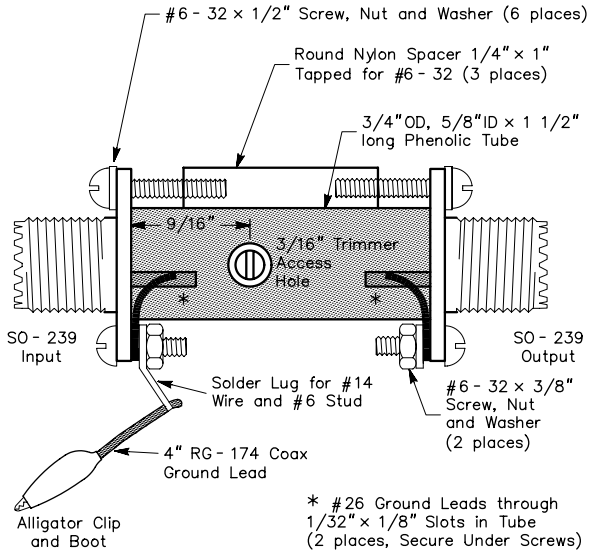



**Fig 14—100× multiplier assembly showing the round Nylon spacer (Mouser 561-TSP10), phenolic tube (US Plastics #47081, see Note 3). The lug is Mouser 571-34120.**

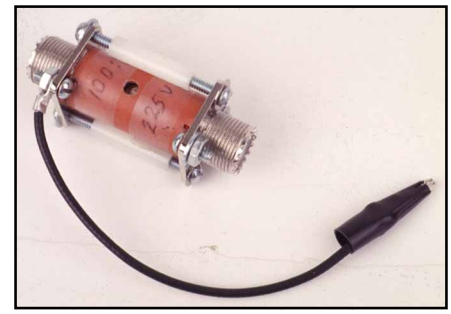


**Fig 15—The assembled 100× multiplier.**

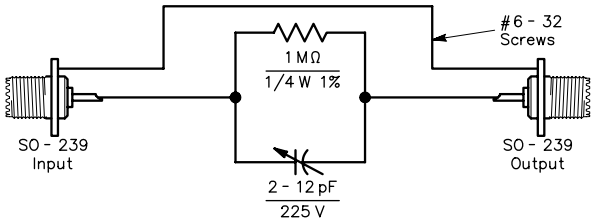



**Fig 16—A schematic diagram of the 2× multiplier. Its output connects to the input of the 100× multiplier for 200× measurements up to 300 V RMS. This multiplier is used *only* with the 100× multiplier.**

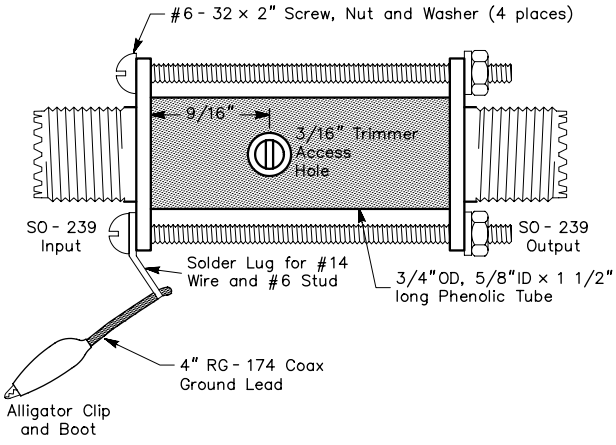



**Fig 17—2× multiplier assembly showing the phenolic tube (US Plastics #47081, see Note 3). The lug is 16-14 #6 stud (Mouser 571-34120).**

an oscilloscope on the output. The trimmer is then set to produce a flat square-wave output with no overshoots on the leading edges and no droop across the top. An RF signal generator that covers 1.9 MHz to 30 MHz can be used instead, if a square wave generator is not available. Since this is a 10× multiplier, it is reasonable to expect the input to handle up to 200 V (RMS), but the limit is 150 V because the trimmer capacitor is rated at 225 V dc. You may be able to find small trimmers with higher voltage ratings that will take the multiplier to 200 V and still fit inside the tube housing.

Multiplier construction follows the same methods used for the RF probe but with a few differences. My first model used an aluminum tube like that in the RF probe with #6-32 screws to hold it all together, but the frequency response began to drop off at 21 MHz. By replacing the aluminum tube with a phenolic tube[3] having the same dimensions, the response is adequate to 30 MHz.

Fig 9 shows construction methods. Two slits at the ends of the tube allow the ground wires at the input and output ends to exit and wrap around the screws of each connector. A hole in the phenolic tube allows adjustment of the trimmer capacitor.

Fig 10 shows the exploded view of the pin tip and grabbers at the input end of the multiplier and a UHF male-to-male connector on the output side to mate with the RF probe. The assembled multiplier and probe are shown in Fig 11. The whole assembly is only 3½ inches long and is comfortable in the hand. If more flexibility is desired, the two pieces may be separated by as much as one foot of coax (Fig 12) without doing too much damage to the accuracy of the measurement.

The 100× multiplier follows the same design concepts as the 10× multiplier including the obligatory trimmer capacitor, as in Fig 13. Here too, the 225 V dc rating of the trimmer limits the maximum RF to 150 V RMS. So, this is not useful should it be used only with the probe and your DVM because the 10× multiplier already covers this range. Later, we will see how it is useful with the RF probe and the meter unit.

The 100× unit also uses a phenolic-tube housing with a hole to adjust the trimmer and a slit at each end to bring out the ground wires, which then wrap around screws in the connectors. In order to reach at least 30 MHz, however, the two-inch steel screws between the two SO-239 connectors must be insulated. Using plastic washers at the connector holes could do this job but would have required #4-40×2-inch-long screws. After a long search at the biggest hardware stores and catalogs,

I found nothing longer than 1½ inches. That's the best way to assemble this multiplier, if you can find the screws. Otherwise, use nylon spacers and shorter screws as shown in Figs 14 and 15. The pin tip and grabber at the input and UHF male-to-male connector or coaxial cable at the output are used as with the 10× multiplier.

The only purpose of the 2× multiplier is to create a 200× multiplier when used with the 100× multiplier in series, so as to extend the voltage range from 150 V to 300 V (1800 W at 50 Ω only under matched conditions[4]). The trimmer in each multiplier now divides in half the input voltage. Fig 16 shows the 2× multiplier with only one RC section since it relies on the RC sections in the 100× multiplier to complete the division (multiplication).

The construction, Fig 17, returns to the tubular format as before, but this time there are no slots to pass the ground wires; the four screws are able to do the job. The two multipliers are shown in series in Fig 18.

The input capacitance for each multiplier is shown in Table 1.

When measurements are made on a load resistance of 50 Ω, the input capacity of the multipliers has no affect on readings below 30 MHz. When the impedance of the load resistance is larger, consider any error it introduces. To get a sense of the affects of the load resistance, use the equation below. It shows the relationship between the resistance, capacitance and frequency when the measured voltage is 3 dB down from what it would be if it were read by a meter with no input capacity:

$$f = \frac{1}{2\pi R_L C_p} \qquad \text{(Eq 1)}$$

where

$R_L$=load resistance

$C_p$=probe capacitance

For example, at a load of 1 kΩ and a probe capacity of 13.3 pF, the 3 dB down frequency is about 12 MHz, where the error is 30%. If an attenuator were used instead of the multiplier, it would not have this problem because it works with either a fixed 50-Ω load at high frequencies or a 600-Ω at low frequencies. Probes cannot select their frequency or load impedance and are thereby more flexible in use, so measurements must be made with consideration, but these multipliers would not have a problem at 50 Ω or 600 Ω either.

Table 2 summarizes the voltage ranges using only the multiplier with the RF probe and a DVM that has a 10-MΩ input resistance that's available in the shack.

The multipliers are intended for use at high voltages where safety precautions are a primary consideration to avoid personal injury. *The ARRL Handbook* has an entire chapter devoted to safety, for good reason. In Tektronix' *ABC's of Probes,*[5] an entire section thoroughly covers the hazards and necessary precautions when making measurements with probes. It is worth the little effort to get a copy of it and also the Pomona Catalog,[6] which has some good information on probe use.

The multipliers have panel connectors at both ends and the RF probe has one at one end. Their grounds are connected and brought to the meter unit. This unit is grounded only when the ac-powered 9-V power supply that may be used with it, is grounded. When it is battery operated, the meter unit relies on the ground-clip connection to the ground of the equipment under test. This is satisfactory if there is no unknown break in the chain of ground connections that would make the panel connector hot. Furthermore, as insurance, it is well to wrap these connectors with vinyl tape or to cover the multiplier with a plastic boot to prevent contact with either hands or equipment under test. This is not shown in any photographs because it would have obscured the appearance and construction of the probe and multipliers.

The frequency responses of the multipliers were determined using an RF signal generator set at the midpoint of each ham band from 160 to 10 meters, including 30 MHz. An error of 5.42% (0.46 dB) was measured from 160 to 17 meters, which then decreased to 4.13% (0.35 dB) at 15 meters and then to zero through 30 MHz. The constant error from 160 to 17 meters is probably due to the inherent errors in the test equipment. The oscilloscope has a 60 MHz bandwidth, an input impedance of 1 MΩ and 30 pF, which introduces a load effect on the multiplier outputs. It also has a reading accuracy of ±3%. When the multiplier-components accuracy of 1% is included, it is not surprising to find the overall inaccuracy to be at most 5.42%.[7] The error of the multiplier itself could be inherently less than this. Before the overall frequency response was measured, the trimmer capacitor was adjusted with the input frequency set mid-frequency at 15 MHz. If your interest in RF probes and multipliers has been raised and you have more questions, see the references in Notes 5 and 6.

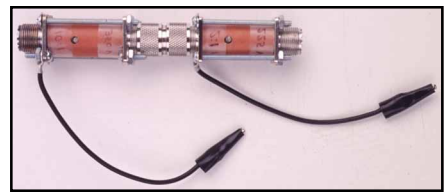


**Fig 18—The 2× and 100× multipliers are joined by a male UHF coupler for readings up to 300 V (RMS, 1800 W, see Note 4).**

### The Meter Unit

The meter unit serves several purposes when used with the RF probe. It increases the accuracy of the probe and DVM from –45% to –3.5% at a voltage of 100 mV RF. At higher voltages, it maintains a minimum advantage of 5:1 in reducing the error, when the probe is used with a DVM. This is an increase in both sensitivity and accuracy. The usual non-linearity caused by the diode in the probe is reduced when used with the meter unit. The technique also incidentally provides temperature compensation for diode drift. The meter unit has a ±5% panel meter to display measurements, but it also contains a pair of **OUTPUT** terminals for a DVM. Add a DVM when more accuracy is desired

**Table 1—Multiplier Range versus Capacitance**

| Multiplier | 10× | 100× | 200× |
|---|---|---|---|
| Input Cap (pF) | 13.3 | 10.9 | 13.5 |

**Table 2—Multiplier versus Voltage Range**

| *Probe Configuration* | *Range (V)* | *Maximum Voltage* |
|---|---|---|
| Probe Alone* | 1.0-20 V ac | 20 V ac |
| With 10× Multiplier* | 10-150 V ac | 225 V (ac + dc) |
| With 100× and 2× Multiplier† | 150-200 V ac | 450 V (ac + dc) |
| With 100× and 2× Multiplier†† | 200-300 V ac | 450 V (ac + dc) |

*error less than –10% or –0.83 dB

†with error of –45% to –10% or less than 3.2 dB. A trimmer capacitor with a rating of 300 V dc increases the 10× range to 200 V ac and reduces the error from 45% to 10%. The RF probe and meter unit reduces the error to less than –3.5% or less than –0.3 dB.

††with error less than –6% or –0.54 dB

at the low end of the range or when you want a bit more resolution. Finally, since the meter unit is fully portable, low-level field-strength measurements are possible with a whip antenna at the probe input connector. Due to its sensitivity and accuracy, the probe can be adapted for use in many places around the shack.

The non-linear response of the probe diode is compensated (improved) by a circuit in the meter unit. The feedback loop of a CA3160 op amp contains a diode matched to the one in the probe (see Note 1). Fig 19 and its sequence of equations present a very simple sketch of how matched diodes do this when dc is applied through a diode. The final equation shows that any difference between the op amp input and output is due to a difference in voltage drops across the two diodes. When the diodes are matched, the error disappears. When RF is applied, the average currents through the diodes must be equal to keep the voltage drops equal. Articles by Kuzdrall (Note 1) Grebenkemper[8] and Lewallen[9] are first-class descriptions of the principles used in this RF voltmeter.

Fig 20 shows three pots for calibrating the meter unit. This should be performed at 400 Hz to avoid any effects due to RF. The 100-kΩ pot is typi-

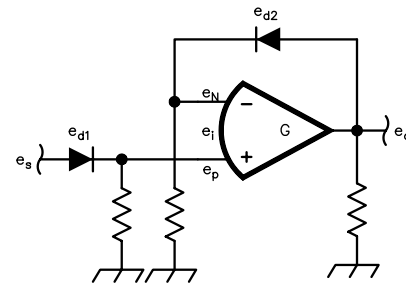
$$e_s - e_{d1} = e_p$$

$$e_o - e_{d2} = e_n$$

$$e_i = e_p - e_n = e_s - e_{d1} - e_o + e_{d2}$$

$$e_i = \frac{e_o}{G} \approx 0$$

$$e_s - e_o = e_{d1} - e_{d2}$$




**$e_s$ represents the input of the RF probe as a dc signal.**
**$e_o$ represents the output from the high-gain CA3160 op amp.**
**$e_{d1}$ is the voltage drop across the diode in the RF probe.**
**$e_{d2}$ the voltage drop across the diode in the feedback loop.**
**G = 320,000 for the CA3160**
**The last equation shows that any difference between the signal and output results from unequal diode drops when the diodes are not matched.**

**Fig 19—Basics of the diode-compensation method to improve measurement accuracy at low input voltages. It also provides a measure of temperature compensation and drift reduction. See Fig 6.**

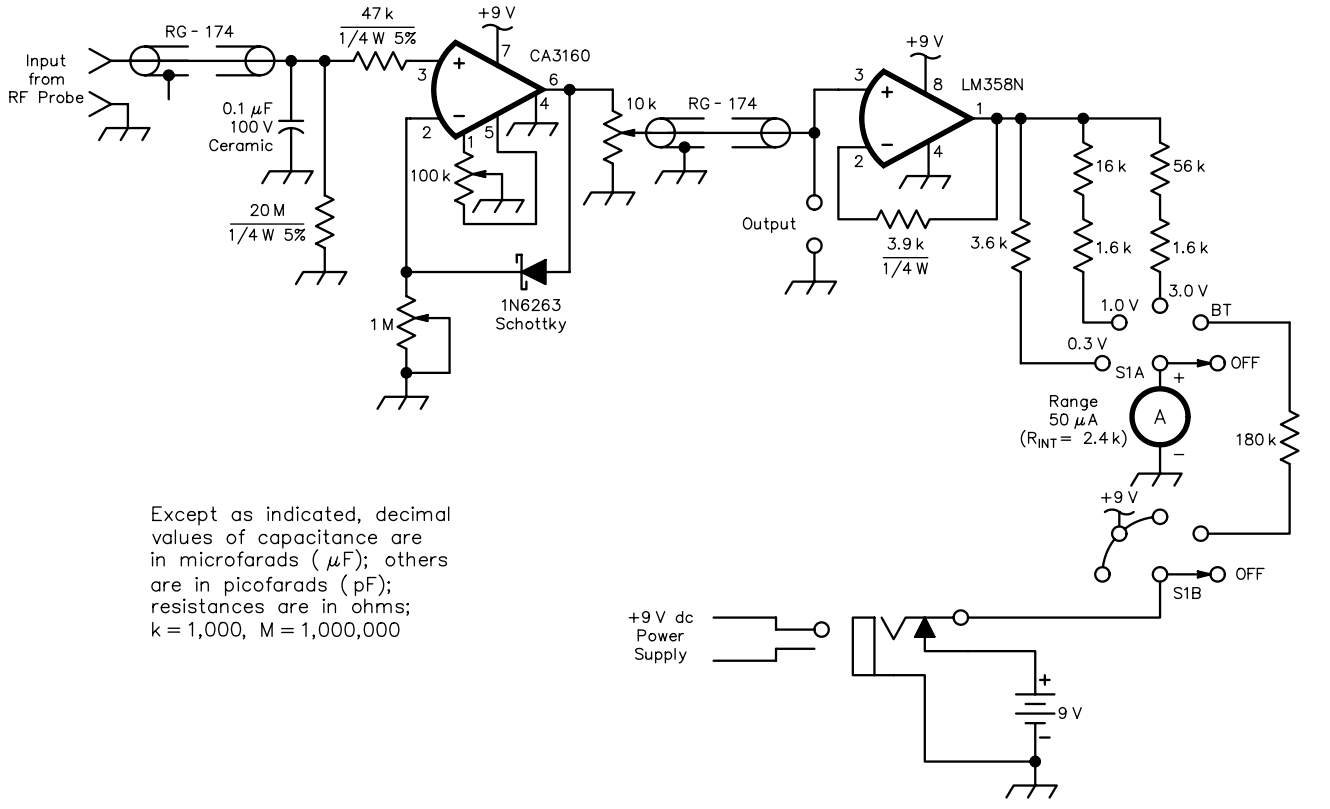
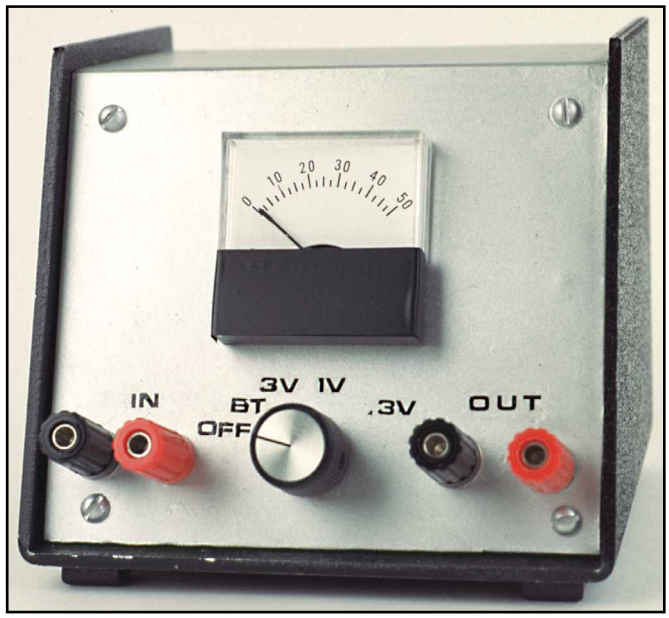



**Fig 20—Schematic of the RF voltmeter. The DVM adds resolution to the panel-meter reading. When the panel meter reads full scale with S1 in the B+ position, the battery is at 9 V.**

**S1—2P5T non-shorting rotary switch, one-inch diameter.**
**A one-lug tie strip is mounted on the negative-terminal screw of the meter to provide a mounting point for the meter resistors.**
**The following resistors are all carbon-composition or metal-film components (1/4 W, ±1%): 1.6 k, 3.6 k, 16 k, 56 k.**
**The three pots are cermet 12-turn components adjustable from the top.**
**The input connector for the RF probe and the output connector for the optional DVM are double banana connectors.**
**The resistors, capacitors, ICs diodes and pots are mounted on RadioShack multipurpose PC board #276-150. The meter, two banana plugs and switch are mounted on the front of the case, the phone jack on the back.**



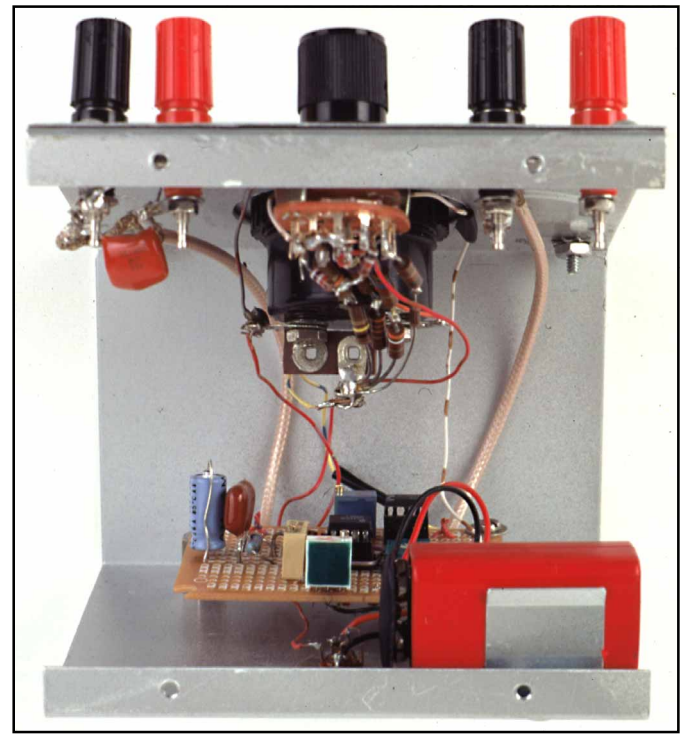



**Fig 21(above)—The assembled meter unit.**
**Fig 22(right)—Inside view of the meter unit. RG-174 connects the INPUT and OUTPUT banana binding posts to the circuit board. The rotary switch, one-lug tie strip and battery are also visible.**

cally used to null the offset of the CA3160, but is used here to initially set the offset to about 0.5 mV to 1.0 mV with no input. Then the 1-MΩ pot sets the output to 100 mV with no input, and the 10 kΩ pot sets the output to 3.0 V when the input is 3.0 V. Finally, the three pots are alternately adjusted until the 100 mV and 3.0 V set points occur together. The 100-kΩ pot is helpful in fine tuning the 100-mV point. A DVM was used at the optional **OUTPUT** sockets during calibration.

A CA3160 was selected for the input op amp because of its high input impedance of 1.5 TΩ and high gain of 320,000. Although it has diodes that provide protection, I think it could be sensitive to electrostatic discharge, so handle it carefully. Use IC sockets to permit easy replacement of the ICs in case of damage. An LM358N IC follows the CA3160 (see Fig 20), primarily to drive the panel meter. If you use a DVM as the display, you can omit the LM358N circuit, meter and multi-position switch. The only functions lost are the battery-voltage check and power on/off switch. Add a on/off switch to the circuit when the multi-position switch is omitted. The CA3160 op amp circuit easily spans the range from 100 mV to 3.0 V without the need for a range switch.



**Fig 23—The meter unit with RF probes and a DVM connected to the OUTPUT jacks.**

## Construction

A 5×5×4½-inch sloping-front instrument case was used to house the components of the meter unit (see Fig 21). The meter on the front panel has a 2×2-inch face and requires a 1½-inch hole in the panel. Although a 50-μA meter from the junk box is used here, a 1-mA movement will work as well, provided the series resistors are changed accordingly.

The rotary switch has two poles and five positions for changing the meter range, testing the battery condition and switching the power off. Two sets of double banana binding posts are used, the **INPUT** pair accepts the dc signal from the RF probe. The **OUTPUT** pair provides a voltage for a DVM display, whether the panel meter is used or not. On the rear of the case, a miniature phone jack accepts 9-V power from either a battery or a 9-V dc supply. The ICs and other parts are mounted on a RadioShack multipurpose PC board that has very convenient holes and traces. The board is bolted to the back of the case via standoff insulators and wired to the front panel components (see Fig 22). The assembly of the meter unit, RF probe and DVM are shown in Fig 23.

## Acknowledgement

The original goal of the design was a linear scale with an accuracy of 5%,

from 100 mV to 3 V (RMS) for frequencies from audio through HF. When I discussed it with *QEX* Managing Editor Robert Schetgen, KU7G, he thought the utility of the meter could be much improved if the voltage range were increased. That set off a chain reaction of improvements, beginning with the compensated multipliers for the RF probe input. The early probe design had a pin at its input, however, which does not easily connect to a divider with an SO-239 connector at its output. Part of the answer is a SO-239 panel connector at the probe input for measurements in coaxial terminations. This was a new voltmeter application, but it seemed to defeat the meter's original purpose: troubleshooting on a chassis.

Rummaging through an ancient parts box provided the link: a probe pin tip with a shank that fits perfectly into the female center pin of the SO-239 connectors on the probe and multipliers. This pin tip is still available in catalogs, surprisingly, but as a standard pin tip it also fits into available alligator clips. This single pin is used for probing or for adding alligator clips or probe adapters to either the RF probe or multipliers. The tip can be removed anytime and the SO-239 used for cable measurements. Bob's outstanding suggestion was taken seriously. The result is an inexpensive, wide-range, RF voltmeter that should not require calibration every six months.

**Notes**

[1]J. A. Kuzdrall, "Linearized RF Detector Spans 50-to-1 Range," Analog Applications Issue *Electronic Design*, June 27, 1994.

[2]Mouser Electronics, 2401 Hwy 287 N, Mansfield, TX 76063; tel 800-346-6873, fax 817-483-0931; E-mail **sales@mouser.com**; **www.mouser.com**.

[3]United States Plastics Corp, 1390 Neubrecht Road, Lima, OH 45801-3196, tel 419-228-2242, fax 419-228-5034.

[4]If the power remains constant, load mismatch multiplies the voltage by the SWR. A 2:1 SWR would produce 600 V, 3:1 900 V and so on.—*Ed*

[5]*ABCs of Probes*, Tektronix Inc, Literature number 60W-6053-7, July 1998. Tektronix, Inc. Export Sales, PO Box 500 M/S 50-255 Beaverton, OR 97707-0001; 503-627-6877. Johnny Parham, "How to Select the Proper Probe," *Electronic Products*, July 1997.

[6]Pomona Test and Measurement Accessories catalog. ITT Pomona Electronics, 1500 E Ninth St, Pomona, CA 91766-3835.

[7]A. Frost, "Are You Measuring Your Circuit or Your Scope Probe?" *EDN*, July 22, 1999. E. Feign, "High-Frequency Probes Drive 50-Ω Measurements," *RF Design*, Oct 1998.

[8]J. Grebenkemper, KI6WX, "The Tandem Match—An Accurate Directional Wattmeter," *QST*, Jan 1987, pp 18-26; and "Tandem Match Corrections," *QST*, Jan 1988, p. 49.

[9]R. Lewallen, W7EL, "A Simplified and Accurate QRP Directional Wattmeter," *QST*, Feb 1990, pp 19-23, 36.

[10]Ocean State Electronics, 6 Industrial Dr, PO Box 1458, Westerly, RI 02891; tel 800-866-6626, fax 401-596-3590.

*Sid Cooper, K2QHE, received his license 45 years ago, although he claims he was genetically programmed by his brother, Bert, WA0BHE (SK), for electronics long before that. He worked on radar and computer design programs as an electronics engineer and manager, then retired from RCA after 20 years. He is fairly active at the IRAC ham club in Roseland, New Jersey, and is a fairly regular participant on the Northeast Corridor Net on Monday nights.* □□


**EZNEC 3.0**

**All New *Windows* Antenna Software by W7EL**

***EZNEC 3.0*** is an all-new antenna analysis program for Windows 95/98/NT/2000. It incorporates all the features that have made ***EZNEC*** the standard program for antenna modeling, plus the power and convenience of a full Windows interface.

***EZNEC 3.0*** can analyze most types of antennas in a realistic operating environment. You describe the antenna to the program, and with the click of a mouse, ***EZNEC 3.0*** shows you the antenna pattern, front/back ratio, input impedance, SWR, and much more. Use ***EZNEC 3.0*** to analyze antenna interactions as well as any changes you want to try. ***EZNEC 3.0*** also includes near field analysis for FCC RF exposure analysis.

***See for yourself***

The ***EZNEC 3.0*** demo is the complete program, with on-line manual and all features, just limited in antenna complexity. It's free, and there's no time limit. Download it from the web site below.

**Prices** - Web site download only: $89. CD-ROM $99 (+ $3 outside U.S./Canada). VISA, MasterCard, and American Express accepted.

***Roy Lewallen, W7EL*** *phone* **503-646-2885**
***P.O. Box 6658*** *fax* **503-671-9046**
***Beaverton, OR 97007*** *email* **w7el@eznec.com**

***http://eznec.com***

**800-522-2253**

**This Number May Not Save Your Life...**

**But it could make it a lot easier! Especially when it comes to ordering non-standard connectors.**

**RF/MICROWAVE CONNECTORS**

- Specials our specialty virtually any SMA, N, TNC, BNC, SMB, or SMC delivered in 2-4 weeks
- Cross reference library to all major manufacturers.
- Large inventory of piece parts for all types of coaxial connectors.
- Experts in supplying "hard to get" RF connectors.
- Connectors supplied to your drawings and specs.
- Our 56 Standard adapters can satisfy virtually any combination of requirements, between SMA, TNC, N, 7mm, BNC and others.
- Extensive inventory of passive RF/Microwave components including attenuators, terminations and dividers.

NEMAL
Cable & Connectors for the Electronics Industry

**NEMAL ELECTRONICS INTERNATIONAL, INC.**

12240 N.E. 14TH AVENUE
NORTH MIAMI, FL 33161
TEL: 305-899-0900 • FAX: 305-895-8178
E-MAIL: INFO@NEMAL.COM
URL: WWW.NEMAL.COM

ICOM IC-756 PRO



The impressive **IC-756 Pro** covers HF plus 6 meters. The high resolution 5 inch TFT color display provides more operating information than ever, including a spectrum scope. The 32 bit floating point DSP provides crisp, clear reception with 41 built-in filters. The "Pro" is *the* choice for serious DXers and contesters.

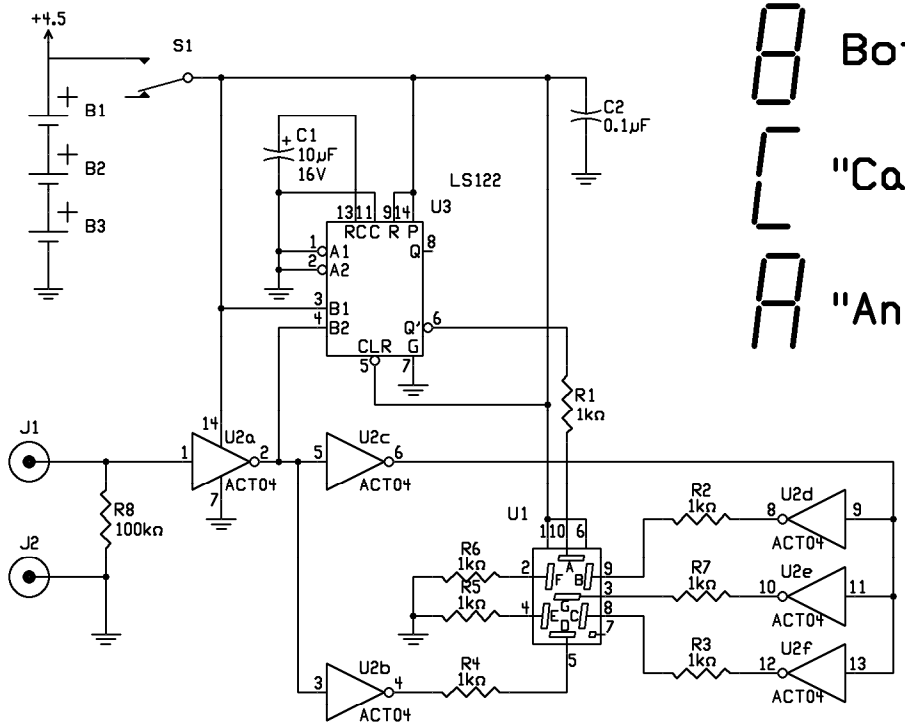
IC-746 ✔ 160 to 2 Meters!



The **IC-746** covers 160-10 meters *plus* 6 and 2 meters with 100 watts on all bands. Call or visit our website for further details and pricing on this and other ICOM radios.

universal radio inc.

**Universal Radio**
**6830 Americana Pkwy.**
**Reynoldsburg, OH 43068**
◆Orders: 800 431-3939
◆Info: 614 866-4267
**www.universal-radio.com**




L Low always

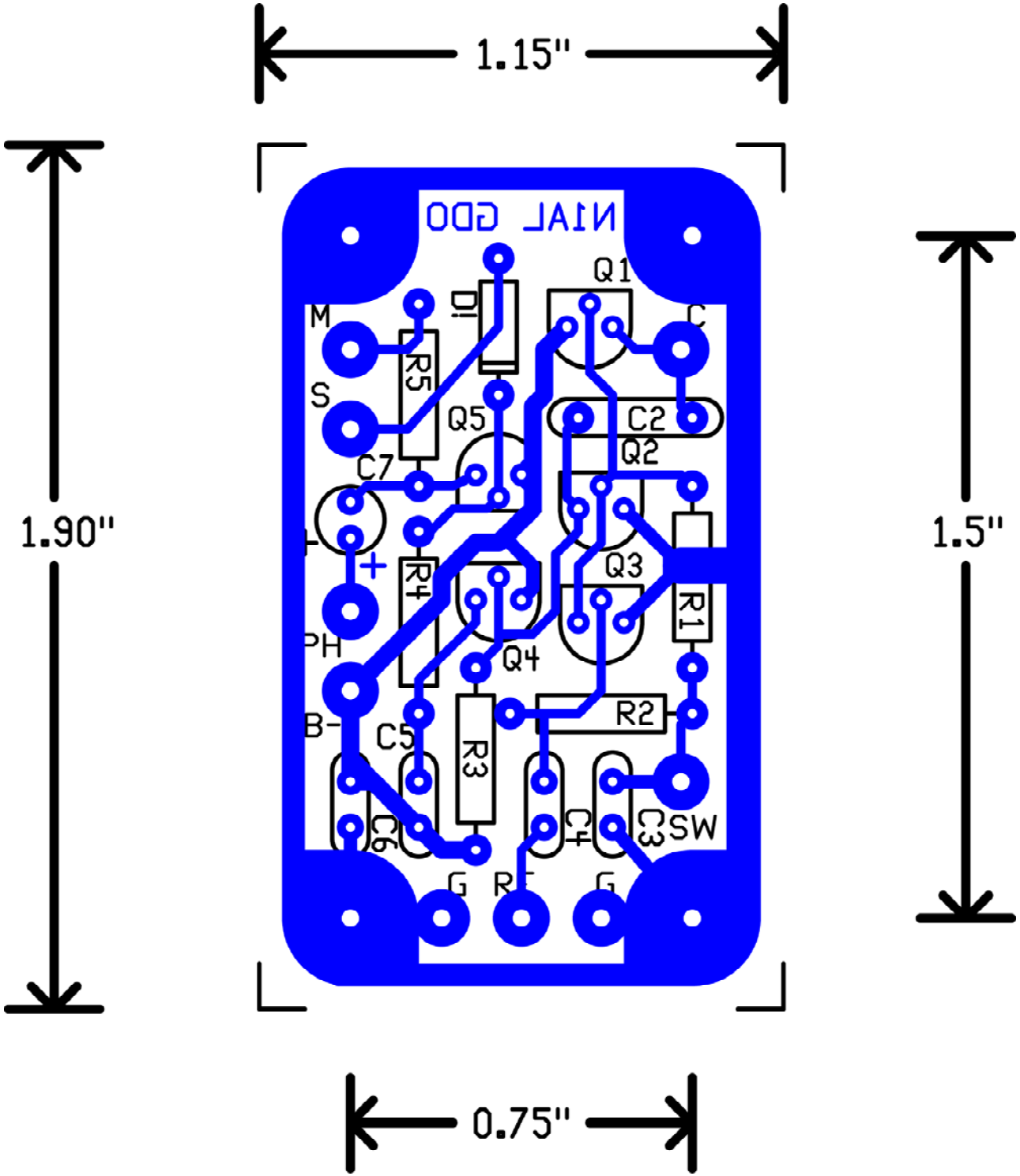
H High always

B Both, ~50% duty factor

C "Cathode" (mostly low, pulses high)

A "Anode" (mostly high, pulses low)

| | | | |
|---|---|---|---|
| Co: | Handbook projects | | |
| Title: | logic_probe.sch | | |
| Board: | Logic Probe | Revision: | A |
| Author: | N1AL | Size: | A |
| Date: | 5/25/2010 | Sheet | 1 of 1 |


1.15"
1.90"
1.5"
0.75"
Q1
D1
M
C
R5
S
Q5
C2
Q2
C7
Q3
R4
R1
PH
Q4
R2
B-
C5
R3
C6
C4
C3
SW
G
G

**Jacques Audet, VE2AZX**

7525 Madrid St, Brossard, QC, Canada J4Y 1G3: **jacaudet@videotron.ca**

# A Low Frequency Adapter for your Vector Network Analyzer (VNA)

*This compact and versatile unit extends low frequency capability down to 20 Hz for your VNA using down/up conversion. It also generates clean signals from audio up to 5 MHz, and provides direct conversion receive capability plus a high impedance input compatible with scope probes to drive 50 Ω loads over a 60 MHz bandwidth.*

I always wished my Hewlett-Packard Vector Network Analyzer (VNA) would be capable of going down to frequencies below 300 kHz. I was working on a project that necessitated frequency response tests in the audio range, but it could not be done with my HP machine. So I started designing a low frequency adapter that would retain the accuracy and linearity of my RF VNA. Figure 1 shows the basic diagram of my Low Frequency Adapter (LFA).

## System Overview

The VNA generates frequencies from 10 MHz to 15 MHz on its port 1 when set in $S_{21}$ mode. This mode allows measuring attenuation or gain as well as phase shift between ports 1 and 2. The above signal is mixed in a double balanced mixer (DBM), which has its local oscillator (LO) at 10 MHz. The difference signals from 20 Hz to 5 MHz go through a low-pass filter (LPF) and are available at the transmit (TX) output port for frequency response testing. The output of the device under test (DUT) is fed to a high impedance buffer and to a 5 MHz low pass filter before being re-multiplexed in the 10 to 15 MHz range by a second DBM.

The signal at the RF output of this mixer has double sidebands, above and below the 10 MHz LO frequency. The VNA synchronously demodulates the upper sideband and uses this signal to compute the attenuation or gain of the device under test in the $S_{21}$ mode. Both the first and second 5 MHz filters (LPF1 and LPF2) provide attenuation of the 10 MHz LO signal, so it does not go thru the device under test path. These filters also greatly attenuate the sum frequencies in the 20 to 25 MHz range, which could decrease the accuracy if these were present at the second DBM IF input.

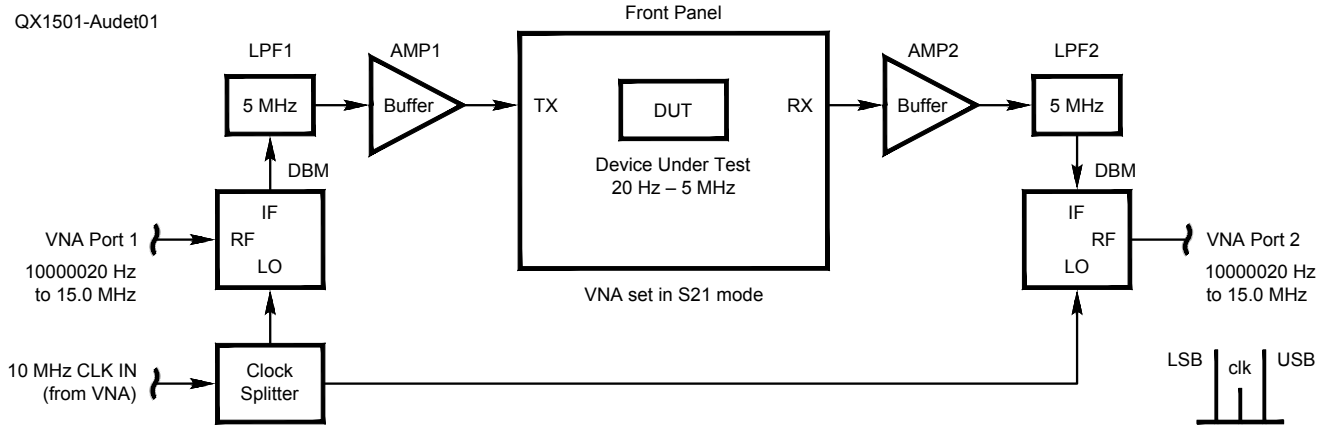



**Figure 1 — Basic block diagram of the low frequency adapter (LFA).**

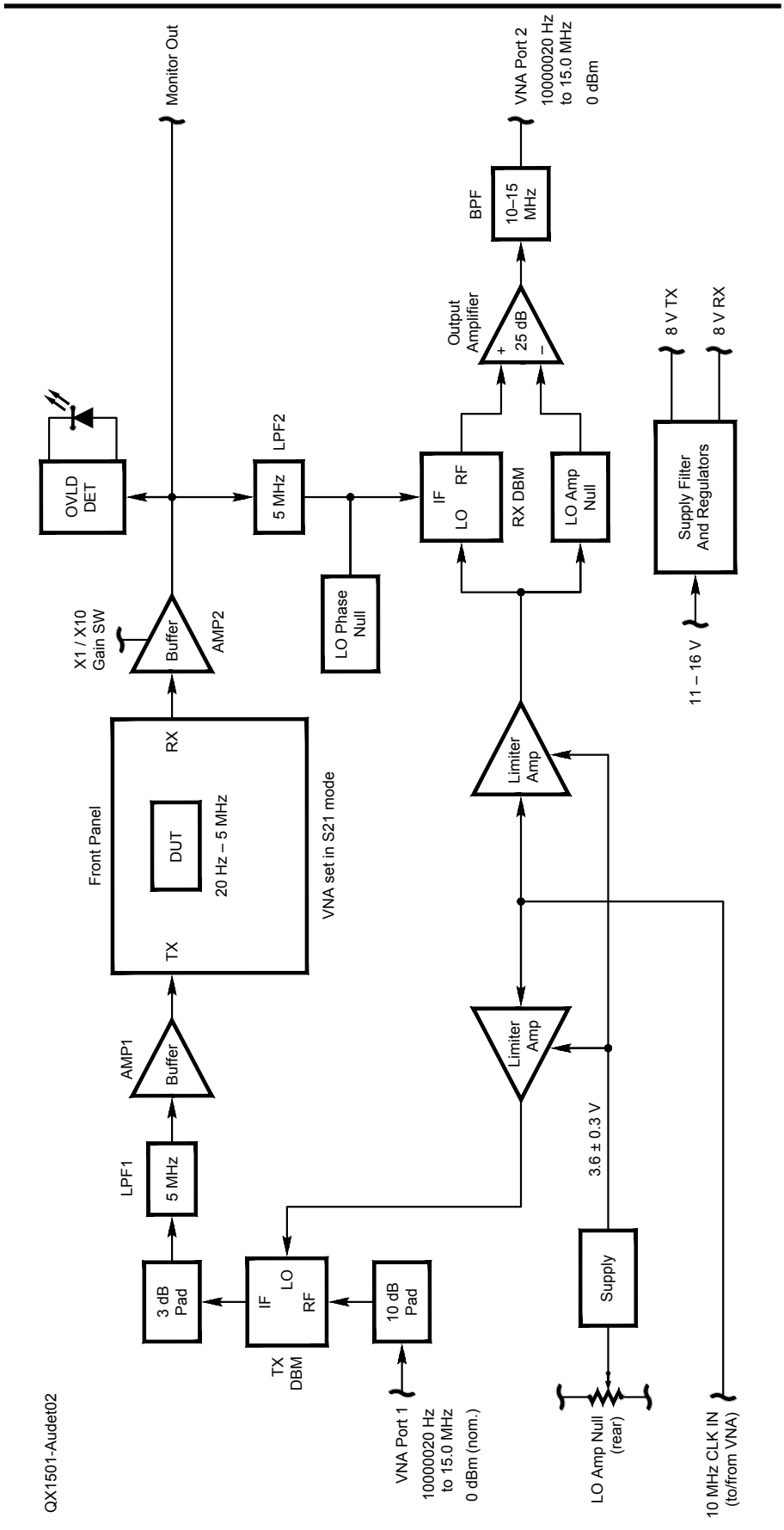



**Figure 2 — Detailed block diagram of the low frequency adapter**

Since the VNA does coherent detection of the signal (in order to measure the phase), it is necessary to have its internal clock synchronized with the low frequency adapter LO signal. This is normally done by using a common external 10 MHz clock feeding the VNA and the low frequency adapter. This also enables the low frequency adapter to do $S_{21}$ phase measurements from 20 Hz to 5 MHz.

I wanted the low frequency adapter to be as transparent as possible to the VNA. This meant that the low frequency adapter should have unity gain from its input and output. Also the TX output port should have a 50 Ω impedance to be able to drive low impedance loads. On the receive side, the input impedance consists of 1 MΩ in parallel with 8 pF so that it is compatible with oscilloscope probes. An additional capacitor may be added to provide the same input capacitance as your scope input for instance, thus providing a flat frequency response with an external ×10 probe. This high impedance provides much flexibility for the user to terminate the device under test by shunting a parallel termination across the receive (RX) input.

Figure 2 shows a more detailed block diagram of the low frequency adapter.

## Detailed Description

The VNA outputs its default-level 0 dBm signal, which is attenuated by 10 dB before reaching the RF port of the TX double balanced mixer. This is done to preserve the linearity at its IF output. The TX output level must track the VNA port 1 level in order to minimize generated distortion on the device under test signals and preserve the VNA capability to perform compression tests. Both double balanced mixers used are of level 7 type, requiring a nominal +7 dBm at the LO port. The 10 MHz clock is sent to a limiter amplifier that drives the two LO ports with a fast square wave, reducing the distortion at high levels and providing a constant LO drive with varying clock input levels. A 3 dB pad at the IF port provides a minimum termination for the TX DBM at both the upper and lower sidebands, helping to minimize distortion. A buffer amplifier (AMP1) provides about 17 dB of gain to compensate for the losses in the pads and the TX DBM, while providing an output level of 225 mV into a high impedance or 112 mV into 50 Ω (–6 dBm).

The RX side provides unity gain from its input to the monitor and VNA outputs. The gain may be increased by 20 dB (×10) using a front switch to improve the system dynamic range. The monitor output has approximately 80 MHz bandwidth in ×1 mode and 60 MHz in ×10 mode. It can be used to monitor the

device under test output with an oscilloscope or to provide a high impedance buffer for the VNA, compatible with oscilloscope probes. An overload detector turns on a red LED when the input exceeds +5 dBm. The same LED is normally green and indicates that the low frequency adapter is powered on. A buffer amplifier (AMP2) also drives a 5 MHz low pass filter, which feeds the IF of the second double balanced mixer. The filter is designed for a 30 Ω load, since this is the impedance that I measured at the IF port when the LO is present.

The signal fed at VNA port 2 consists of the two sidebands and some 10 MHz LO carrier leak.

The second double balanced mixer (RX DBM) must provide a high degree of rejection of the LO signal. This is important at the lowest frequencies, from 20 Hz to 1000 Hz. The LO appears as a spurious signal at the VNA port 2, even more than the undesired LSB signal. For this reason, it is necessary to use the lowest IF bandwidth at the VNA (10 Hz for the HP 8753x). This will require slowing down the sweep speed.

The selected RX DBM provides excellent LO rejection from the LO port to the RF port (typically 70 dB). This was not enough at the lowest frequencies, and limited the dynamic range to 60 to 70 dB, however. The expected dynamic range is 90 dB, which occurs above 2 kHz. With an LO at +7 dBm, and 70 dB of LO-RF isolation, –63 dBm will appear at the RF port. This signal is then amplified by the output amplifier, which has 25 dB of gain. This brings the LO to –38 dBm at the VNA port 2. This high level will reduce the dynamic range at the lowest frequencies. We need to reduce the LO feedthrough below –70 dBm at port 2 to restore the 90 dB dynamic range below 100 Hz.

The external LO nulling circuit uses the LO to feed an adjustable voltage divider connected to the inverting input of the output amplifier. Every double balanced mixer has its own LO to RF phase shift characteristics,

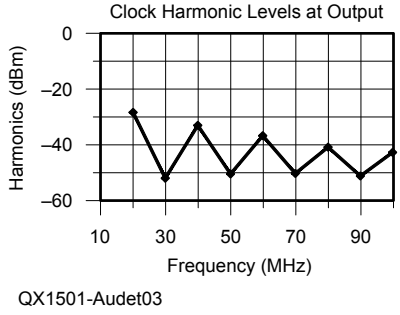



**Figure 3 — Measured harmonic Levels of the 10 MHz LO at the output of the RX DBM mixer. Note the strong levels at all even multiples of the LO frequency.**

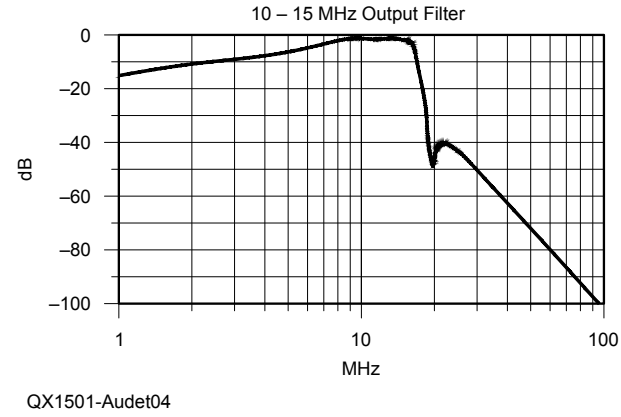



**Figure 4 — Band-pass filter frequency response.**

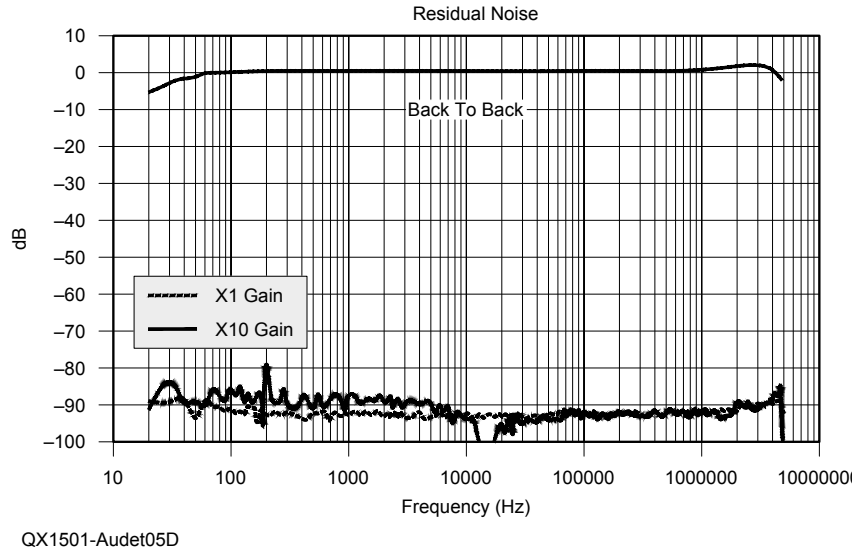



**Figure 5 — System measured back-to-back frequency response and residual noise level.**

however, so a variable phase shifting circuit is required. I found that adding bias current in the order of a few microamps to the double balanced mixer IF port did shift the phase. A few double balanced mixers, out of the 40 units I tested, had their initial phase too far and could not be used on the RX side. They can be used as a TX double balanced mixers, however.

There are two amplitude nulling adjustments, one located on the circuit board and the other at the rear. The phase adjustment is located at the rear. Doing so allows easy tweaking of LO null before low frequency (below 1 kHz) response tests are done.

During the testing phase, I found that I had some intermodulation problems at the VNA. Looking at the spectrum, I discovered that high levels of harmonics were present at the RX double balanced mixer RF output. See Figure 3.

A band-pass filter was added at the output to clean up these spurs and get rid of all intermodulation problems. Figure 4 shows the filter frequency response.

Figure 5 plots the system back-to-back frequency response and residual noise level, showing a 90 dB and 110 dB dynamic range in the ×1 and ×10 modes respectively.

Figure 6 shows a picture of the completed unit. Part A shows the font panel, while Part B shows the rear panel. Figure 7 is a picture of the assembled circuit board. The detailed circuit schematic for the low frequency adapter is not printed in the article, but can be downloaded from the author's website and the ARRL *QEX* files website.[1, 2] The adapter

[1]Notes appear on page 16.




**Figure 6 — The low frequency adapter is built in a standard Hammond extruded aluminum cabinet. The front and rear panels are pictured.**

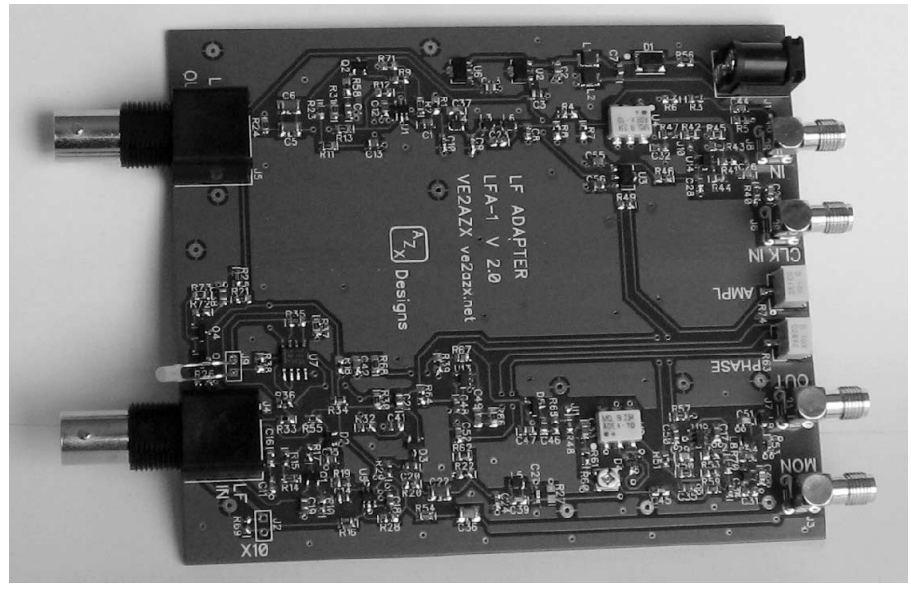



**Figure 7 — The assembled circuit board. The R, L and C components are of 0805-size, surface-mounted. Smaller 0603 components will also fit on the pads.**

is built on a 3.9 × 4.68 inch (9.9 × 11.89 cm) circuit board, using surface-mounted components on the top surface. It uses a power supply Pi filter and separate regulators for the TX and RX sides, to enhance isolation. The amplifiers used are of the current feedback type, with stable gain that is set by resistors. Two low noise +4 V DC references, consisting of emitter followers, provide a constant output impedance and a flat frequency response. The board may be assembled in approximately five hours.

## Low Frequency Adapter Basic Capabilities

All $S_{21}$ measurements are performed within the 10 to 15 MHz frequency range of the VNA. The low frequency measurements include:

- $S_{21}$ magnitude and phase,
- Group Delay,
- Compression point at a single frequency.
- TRU Calibration, with the device under test bypassed with a short circuit, to set a reference amplitude and phase frequency response.

Figure 8 shows the linearity and accuracy in measuring $S_{21}$ magnitude.

## Impedance Measurements

The unit as such does not provide for measuring the reflection coefficient $S_{11}$. The problem of measuring impedances with a return loss bridge is that the accuracy decreases rapidly when the unknown impedance is more than five times above or below the

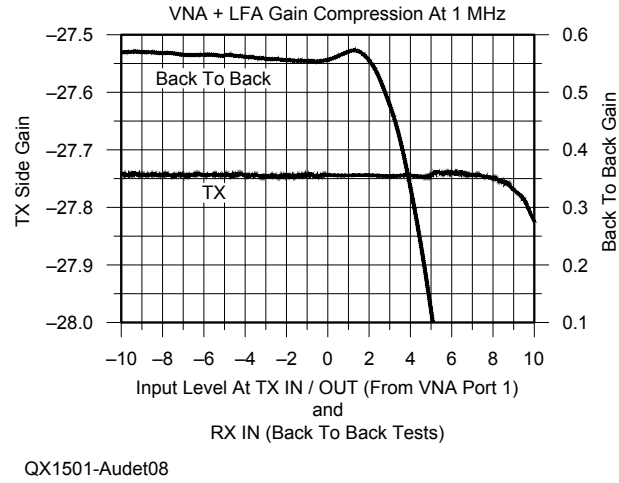



**Figure 8 — Linearity / Gain Compression test results. The TX output gain rolls off less than 0.1 dB at +10 dBm, while the RX side gain compression measures ±0.05 dB up to +3 dBm input.**

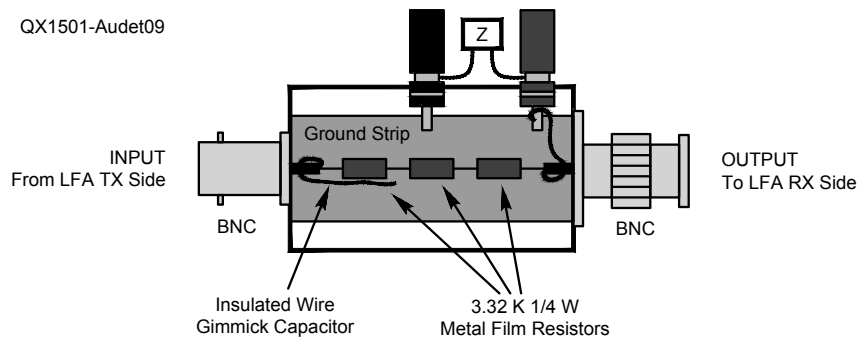



**Figure 9 — Here is the shunt impedance adapter physical layout used for impedance measurements. Note the three resistors connected in series to reduce the overall shunt capacitance.**

reference impedance. At low frequencies, we are generally interested in measuring a wide range of impedances, from say 0.1 Ω to 1 MΩ. This measurement is best handled by a series or shunt transmission circuit, where $S_{21}$ is measured as a complex value and used to compute the impedance value, including the reactance, inductance, capacitance, and resistance in the series or parallel models. I have created an *Excel* spreadsheet that performs these computations.[3]

Note that the low frequency adapter provides 1 MΩ impedance on the RX side, which gives the user more flexibility in doing shunt impedance measurements. When the device under test cannot be floated, then the shunt configuration must be used. With the normal 50 Ω source and load impedances, the maximum value of shunt impedance that can be measured does not exceed about five times 50 Ω, or 250 Ω. Adding resistors in series with the 50 Ω source (TX) and load (RX) will increase the accuracy in measuring higher impedances at the expense of reducing the dynamic range. For example adding 1000 Ω in series at both the TX and RX sides will give about 32 dB of attenuation when the shunt device under test is unconnected.

With the low frequency adapter, the RX impedance is raised from 50 Ω to 1 M Ω, the attenuation will be about 0 dB with the shunt device under test unconnected, and no reduction in the dynamic range will occur. The *Excel* spreadsheet performs these calculations. Figure 9 shows the layout of the shunt impedance adapter that I used. It provides impedance measurements from < 1 Ω to over 20 kΩ, from 20 Hz to 10 MHz. Photo A shows the completed shunt impedance adapter.

**Table 1**
**Using external probe and attenuator allows large signal capability.**

| *Maximum Input Amplitude* | *Setup* |
|---|---|
| 50 mV rms | ×10 Gain on LFA, improves dynamic range |
| 500 mV rms | ×1 Gain on LFA |
| 5 V rms | ×1 Gain + ×10 probe |
| 50V rms | ×1 Gain + ×100 attenuator 1 MΩ in/out |
| 500 V rms | ×1 Gain + ×100 attenuator + ×10 probe |

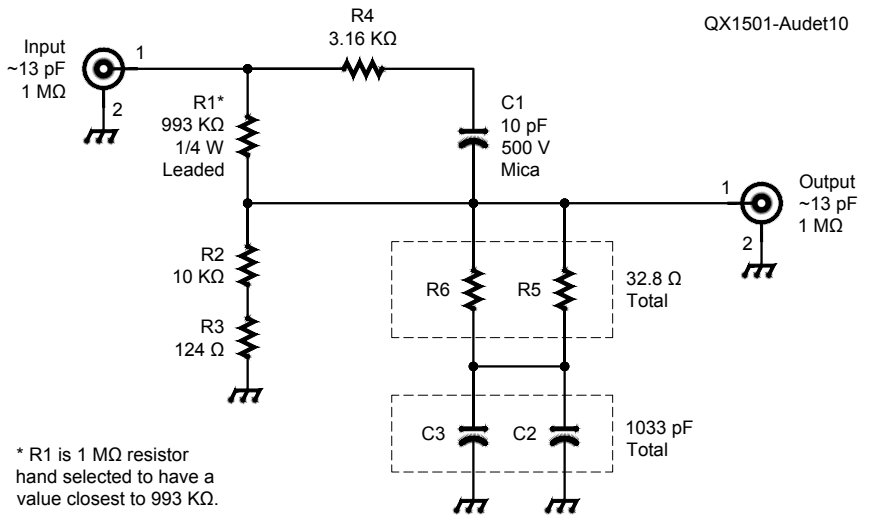



**Figure 10 — The schematic diagram for the ×100 divider/attenuator.**




**Photo A — This photo shows the assembled shunt impedance adapter. The female BNC connector accepts the signal from the low frequency adapter TX connector. The male BNC connector is the output signal, which goes to the low frequency adapter RX connector.**

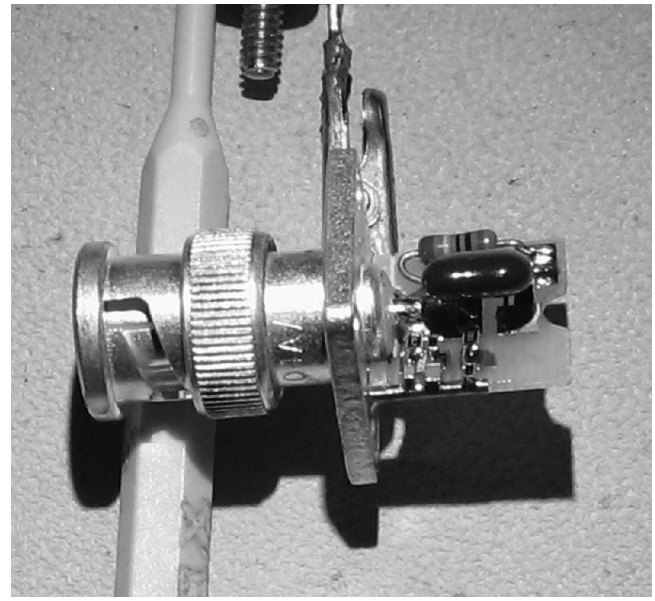


**Photo B — Here is the male BNC connector, with the ×100 attenuator circuit board attached. A short wire from the female BNC connector goes to the circuit board pad at the top right corner before placing that end into the square aluminum extrusion.**

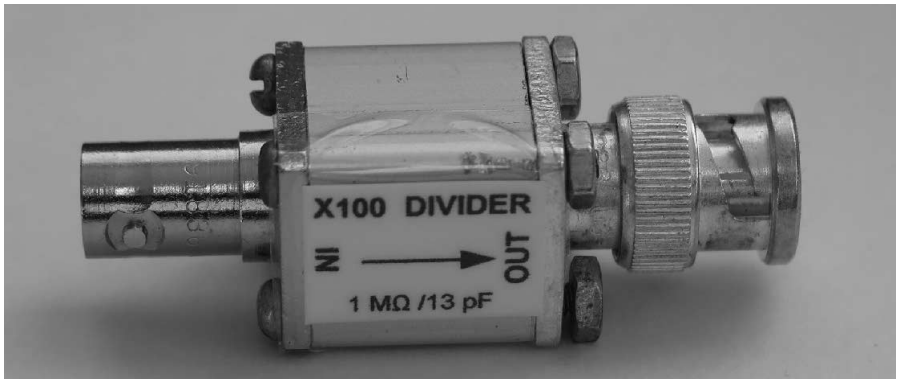



**Photo C — This photo shows the completed ×100 attenuator unit.**

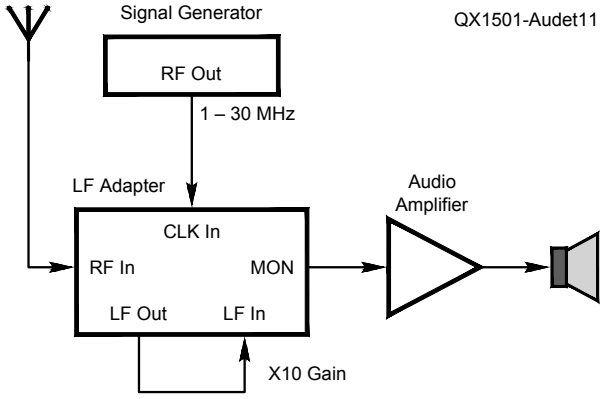



**Figure 11 — Setup for an HF direct conversion receiver. The monitor output uses ×10 gain to improve the signal to noise ratio.**

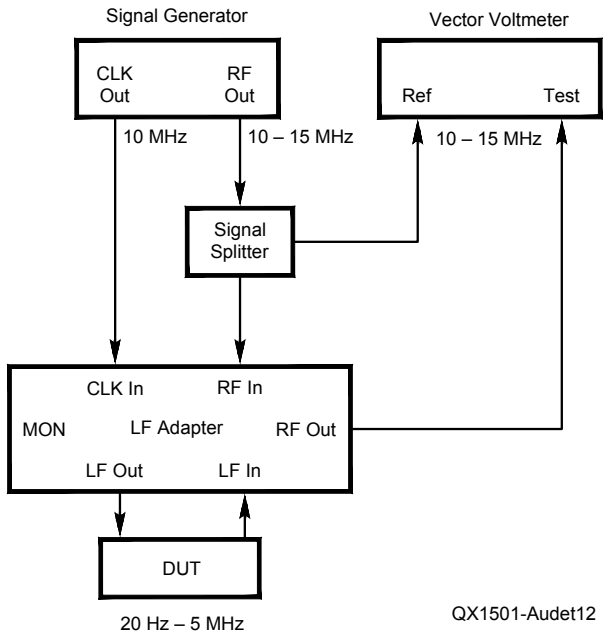



**Figure 12 — Setup for using the vector voltmeter at low frequency, using a synthesized RF signal generator. While I have not tried this setup, a very similar arrangement is described in Note 5.**

## Other Applications of this Low Frequency Adapter

• **Probe Buffer / Amplifier** using the monitor output. It provides an 80 MHz bandwidth in ×1 mode and a 60 MHz bandwidth in ×10 mode.

Using a ×100 attenuator at the low frequency adapter RX input, and/or a ×10 probe provides the ability to handle a very wide range of amplitudes, as shown in Table 1. Figure 10 shows the schematic diagram of the ×100 1 MΩ wideband attenuator that I designed. Photo B is a picture of the circuit board and BNC male connector for the output side of the attenuator. A BNC female connector is used for the input side, and that attaches to the circuit board with a single wire to the circuit board pad between the 933 kΩ and 3.16 kΩ resistors. Photo C is the complete, assembled attenuator. For more details about this ×100 attenuator see the file **Divider-100.pdf** that is part of the **1x15_Audet.zip** file on the ARRL *QEX* files website.[4]

• **Low Frequency Signal Generator**, using your RF signal generator.

The generator clock output provides the clock signal for the low frequency adapter. Its frequency is set from 10.00002 MHz to 15 MHz, to generate an output from 20 Hz to 5 MHz. The low frequency adapter output amplitude tracks the generator output up to +10 dBm.

• **Direct Conversion Receiver**

The low frequency adapter may be used as a simple wideband direct conversion receiver with a well-defined 5 MHz bandwidth. Both sidebands will be detected. Using a low phase noise signal generator allows testing the phase noise of an oscillator using a computer soundcard as a spectrum analyzer. Another possible use is for calibrating the signal generator time base against WWV or CHU, for off the air frequency measurements. Figure 11 illustrates how the low frequency adapter can be connected to an antenna, an RF signal generator and an audio amplifier to create a direct conversion receiver.

• **Vector Voltmeter Operation at Low Frequency**

Figure 12 illustrates the basic operation of the low frequency adapter with a vector voltmeter, for making low frequency measurements. The 10 to 15 MHz signal generator output connects to a signal splitter, so the same signal is going to the RF In port on the low frequency adapter and also to the reference input of the vector voltmeter. Notice that a 10 MHz clock reference signal also connects from the signal generator to the Clock In port on the low frequency adapter. The RF Output, or RX port on the adapter connects to the Test signal input port on the vector voltmeter. The device under test connects to the LF In and LF Out ports on the

front of the low frequency adapter. I have not tried this setup, but a very similar arrangement is described in a Feb 1972 *IEEE Transactions on Instrumentation and Measurements* article.[5]

## Conclusion

The Low Frequency Adapter described here adds low frequency capability to a Vector Network Analyzer, as well as adding audio frequency generation, 1 MΩ probe amplified interface, and direct conversion receiver capability. In the VNA application, an IF bandwidth of 10 Hz must be used to extend the lowest frequency down to 20 Hz. The low frequency adapter has allowed me to measure R, L and C components down to 20 Hz, using my Hewlett Packard 8753D VNA. I have also been able to accurately characterize and document the response of many audio type amplifiers that otherwise would have required tedious measurement methods. I wish to acknowledge the help and encouragement of my good friend Bertrand Zauhar, VE2ZAZ, who did the beta testing on this project. Readers seeking more information (component procurement, circuit schematic, updates, and so on) should visit my website at **http://ve2azx.net/technical/LFA/LowFreqAdapter.htm**.


**We Design And Manufacture To Meet Your Requirements**
***Protoype or Production Quantities**

**800-522-2253**

**This Number May Not Save Your Life...**

**But it could make it a lot easier! Especially when it comes to ordering non-standard connectors.**

**RF/MICROWAVE CONNECTORS, CABLES AND ASSEMBLIES**

- Specials our specialty. Virtually any SMA, N, TNC, HN, LC, RP, BNC, SMB, or SMC delivered in 2-4 weeks.
- Cross reference library to all major manufacturers.
- Experts in supplying "hard to get" RF connectors.
- Our adapters can satisfy virtually any combination of requirements between series.
- Extensive inventory of passive RF/Microwave components including attenuators, terminations and dividers.
- No minimum order.

NEMAL
Cable & Connectors for the Electronics Industry

**NEMAL ELECTRONICS INTERNATIONAL, INC.**
12240 N.E. 14TH AVENUE
NORTH MIAMI, FL 33161
TEL: 305-899-0900 • FAX: 305-895-8178
E-MAIL: INFO@NEMAL.COM
BRASIL: (011) 5535-2368

**URL: WWW.NEMAL.COM**



*Jacques Audet, VE2AZX, became interested in radio at the age of 14, after playing with crystal radio sets and repairing old receivers. At age 17, he obtained his first ham license. In 1967 he obtained his BS degree in electrical engineering from Laval University. He then worked in engineering functions at Nortel Networks, where he retired in 2000. He worked mostly in test engineering on a number of products and components operating from dc to light-wave frequencies.*

*His areas of interest are in RF simulations, filters, duplexers, antennas and using computers to develop new test techniques in measurement and data processing. Jacques is an ARRL Member.*


## Notes

[1]For the complete circuit schematic, as well as any updated information, see the file named **LFA-V2-Schematic.pdf** on my web site: **ve2azx.net/technical/LFA/LowFreqAdapter.htm**.

[2]The complete circuit schematic, as well as other files associated with this article are available for download from the ARRL *QEX* files website. Go to **www.arrl.org/qexfiles** and look for the file **1x15_Audet.zip**.

[3]The *Excel* spreadsheet that I created to perform these calculations is available for download on my web site: **ve2azx.net/technical/Series_Parallell-Impedance_with_VNA-Re_Im.xls**. This spreadsheet is also included in the **1x15_Audet.zip** file available for download on the ARRL QEX files website.

[4]There is more information about the ×100 1 MΩ wideband attenuator included in the file **Divider-100.pdf**, which is part of the **1x15_Audet.zip** file on *QEX* files website.

[5]R. Lane and J. Butler, "LF Adaptor Extends Vector Voltmeter Magnitude and Phase Measurements down to 10 Hz," *IEEE Transactions on Instrumentation and Measurements*, Feb 1972. A copy of this

## Appendix
## Low Frequency Adapter Specifications

**Supported VNA Mode**: $S_{21}$.

**LFA Frequency Range**: 20 Hz to 5 MHz.

In the VNA application, an IF bandwidth of 10 Hz must be used to extend the lowest frequency down to 20 Hz.

**Clock Input**: 10 MHz, 0 to +12 dBm, 50 Ω. Clock must be synchronized to the VNA reference clock.

**Input to LFA from VNA Port 1**: 10.000 020 MHz to 15.000 MHz, 0 dBm nominal, Max +6 dBm, 50 Ω (at rear).

**Output from LFA to VNA Port 2**: 10.000 020 MHz to 15.000 MHz, 0 dBm nominal, Max +6 dBm, 50 Ω (at rear).

**Low Frequency Output**: 20 Hz to 5 MHz, 50 Ω, AC coupled, –6 dBm (112 mV RMS into 50 Ω) or 225 mV nominal into an open circuit.

Return Loss: > 40 dB below 5 MHz.

Harmonics at 1 MHz Out / 0 dBm: –65, –70, –50, –70 dBc for the 2nd, 3rd, 4th,5th harmonics.

Spurs Measured Under No Input From VNA Port 1: – 85, –97, –101, –102, –104 dBm at 10, 30, 50, 70, 90 MHz.

Source Linearity (Up to +9 dBm Out): < 0.05 dB

**Low Frequency Input**: 20 Hz to 5 MHz, 1 MΩ in parallel with 8 pF, AC coupled, 0 dBm nominal input level (front panel).

Overload Level: ~ +5 dBm, as indicated by a red LED on the front panel.

Damage Level: 20 V RMS.

Loopback of Low Frequency Output to Low Frequency Input: Yields ~ 0 dBm output to VNA port 2 at ×1 Gain.

Frequency Response Uncorrected: –3 db at 35 Hz and 5 MHz, ±1.5 dB from 100 Hz to 4 MHz.

A front panel switch provides ×10 gain to compensate for ×10 scope probe attenuation.

Back-to-back linearity, including HP 8753D VNA: Up to +3 dBm in: < ±0.05 dB.

Two rear adjustments potentiometers allow for tweaking the LO carrier balance to improve dynamic range below 1000 Hz.

**Dynamic range:** better than 90 dB from 100 Hz to 5 MHz, when used with a VNA having more than 100 dB of dynamic range. Improves by 20 dB at ×10 gain.

**Sweep time:** For a 170 point log sweep, from 20 Hz to 5 MHz: 50 seconds on the HP 8753D, using 10 Hz IF Bandwidth and LIST frequency mode.

**Monitor Output**: 50 Ω, AC coupled, open circuit level equals input level ±1 dB.

Frequency response –3 dB into 50 Ω: 200 Hz to > 80 MHz at ×1 and > 60 MHz at ×10 gain.

Maximum Output Level: + 6 dBm into 50 Ω

**Power input:** +11 V to +16 V DC at < 80 mA (at rear). A balanced Pi filter at the input cleans up the DC source from any switching power supply noise.

**Size:** 10.5 cm (W) × 15 cm (L) × 3.5 cm (H).

# ARRL Periodicals Archive – Search Results

## A membership benefit of ARRL and the ARRL Technical Information Service

**ARRL Members:** You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

**Need a higher quality reprint or scan?** Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

**QST on CD-ROM**: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

**Non-Members:** Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.

### Copyright/Reprint Notice

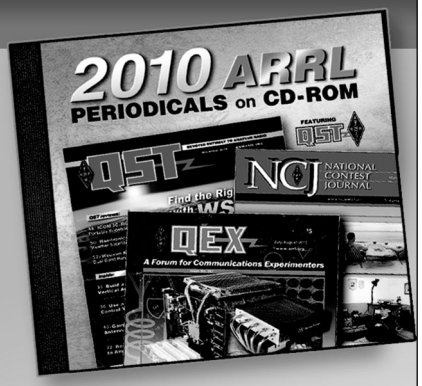
In general, all ARRL content is copyrighted. ARRL articles, pages, or documents--printed and online--are not in the public domain. Therefore, they may not be freely distributed or copied. Additionally, no part of this document may be copied, sold to third parties, or otherwise commercially exploited without the explicit prior written consent of ARRL. You cannot post this document to a Web site or otherwise distribute it to others through any electronic medium.

For permission to quote or reprint material from ARRL, send a request including the issue date, a description of the material requested, and a description of where you intend to use the reprinted material to the ARRL Editorial & Production Department: permission@arrl.org.

**QST Issue:** May 2003
**Title:** A Modern GDO--The "Gate" Dip Oscillator
**Author:** Alan Bloom, N1AL

Click Here to Report a Problem with this File

ADVERTISEMENT




**2010 ARRL Periodicals**
on CD-ROM

**ARRL's popular journals are available on a compact, fully-searchable CD-ROM. Every word and photo published throughout 2010 is included!**

- ***QST*** **The official membership journal of ARRL**
- ***NCJ*** **National Contest Journal**
- ***QEX*** **Forum for Communications Experimenters**

**SEARCH** the full text of every article by entering titles, call signs, names—almost any word. **SEE** every word, photo (including color images), drawing and table in technical and general-interest features, columns and product reviews, plus all advertisements. **PRINT** what you see, or copy it into other applications.

**System Requirements**: Microsoft Windows™ and Macintosh systems, using the industry standard Adobe® Acrobat® Reader® software. The Acrobat Reader is a free download at **www.adobe.com**.

**2010 ARRL Periodicals on CD-ROM**

ARRL Order No. 2001
**Only $24.95***
*plus shipping and handling

Additional sets available:
2009 Ed., ARRL Order No. 1486, $24.95
2008 Ed., ARRL Order No. 9406, $24.95
2007 Ed., ARRL Order No. 1204, $19.95
2006 Ed., ARRL Order No. 9841, $19.95
2005 Ed., ARRL Order No. 9574, $19.95
2004 Ed., ARRL Order No. 9396, $19.95
2003 Ed., ARRL Order No. 9124, $19.95
2002 Ed., ARRL Order No. 8802, $19.95
2001 Ed., ARRL Order No. 8632, $19.95

ARRL The national association for AMATEUR RADIO™
SHOP DIRECT *or call for a dealer near you.*
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)

May 2003 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved



By Alan Bloom, N1AL

# A Modern GDO—The "Gate" Dip Oscillator

## Building an RF circuit? Use an updated version of an indispensable and legendary piece of equipment to test it—the "dipper."

I still have my first grid-dip oscillator, purchased at a flea market about 30 years ago. It's an Eico model 710 that covers 0.4 to 250 MHz with 8 plug-in coils. It weighs over 2 pounds and must be plugged into the wall to obtain power for its 6AF4 vacuum tube. It still works, although the coil socket has become intermittent over the years, which makes it a bit frustrating to use.

Amateurs have been using grid-dip oscillators (GDOs) at least since 1947.[1] While these days they no longer have a vacuum-tube "grid" to dip, the name seems to have stuck. A GDO is a tunable oscillator with the coil mounted outside of the chassis. The external coil allows you to measure the frequency of a tuned circuit without any electrical connection to it. Just bring the coil close to the tuned circuit's coil and tune the GDO while watching for a dip in its meter reading.

This "no connection" measurement capability is handy for other purposes as well. An example is measuring the parasitic elements of a beam antenna. Since the parasitic elements have no feedpoint, you can't measure the resonant frequency the normal way, with an SWR meter. A GDO is also perfect for tuning the traps to resonance on a multi-band Yagi.

A GDO can "sniff out" spurious resonances in the tank circuit of a high-power vacuum-tube RF amplifier. Be sure to turn off the high voltage first!

A GDO has other uses you might not think of. For example, it can measure inductance. Just temporarily connect a capacitor with a known value in parallel with the coil to be tested, measure the resonant frequency and use the formula $L = 1/(2\pi f)^2C$ or use the graph in *The ARRL Handbook*.[2] (L is inductance in µH, C is capacitance in µF and f is frequency in MHz.) It is just as useful for measuring unknown capacitors. Variable capacitors normally are not marked with their value. To identify that "bread slicer" you bought for a song at the local hamfest, just connect it in parallel with a known inductance, measure the frequency, and use the formula $C = 1/(2\pi f)^2L$.



**Figure 1—The "gate dipper" fits comfortably in the hand.**

When you are troubleshooting a receiver, a GDO can help isolate the fault. Tune the GDO to each of the IF frequencies, starting with the last and hold the coil close to that part of the circuitry. If you can hear a signal, then that IF and all the circuitry after it are working. This is especially handy on densely packed, surface-mount boards that are difficult to probe.

If the GDO has an RF output connector, it can be used as a "poor ham's signal generator" in applications where good frequency stability and calibrated level accuracy are not needed. An example is functional "go/no-go" testing of devices such as amplifiers, mixers and wide-band filters.

With the oscillator turned off, the instrument functions as a type of tuned RF detector known as an absorption frequency meter. The obvious use is to check if there is RF energy of the proper frequency in a tuned circuit. It also makes a handy frequency-selective field-strength meter using the capacity probe as an antenna. Another use is to check for feed line radiation. If you run the coil along the coax cable while transmitting, the meter will show the peaks and valleys of the standing waves of the unwanted RF current flowing on the outside of the coax shield. A GDO also makes an excellent "sniffer" to check for RF leakage from a shielded transmitter chassis, both at the

[1]Notes appear on page 58.

May 2003 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

fundamental and harmonic frequencies.

Headphone output was indispensable in the "old days" to check the modulation quality of AM transmitters. Hams don't use AM very much any more, but headphones are still useful to listen for key clicks, hum and low-frequency parasitic oscillations. And, if you couple the coil to an antenna, you have a wide-range "crystal" set!

My favorite, literally "off-the-wall," application was mentioned in a comprehensive tutorial on grid-dip oscillators in January 1974 *QST*.[3] A GDO makes a sensitive detector of pipes and other metallic objects inside a wall. If you listen to the signal on a CW or SSB receiver, you can easily hear a change in frequency whenever the coil passes near a metal object.

## The Design

I got the idea for this design while reading an article in *Amateur Radio*, the journal of The Wireless Institute of Australia, by Lloyd Butler, VK5BR.[4] The nice feature of his circuit is that it uses two-terminal plug-in coils. One terminal is grounded so you can use a rugged, reliable coax connector for the coil socket. Most GDO designs require three connections to the coil or two connections that must both be insulated from ground.[5] Many of the latter also require a two-section tuning capacitor, which can be hard to find.

The old Heathkit "Tunnel Dipper" had two-terminal coils using a tunnel diode in a negative-resistance oscillator. Tunnel diodes are very hard to find nowadays, so VK5BR made a negative resistance element by connecting N-channel and P-channel JFETs back-to-back (gate to drain). I tried it and it oscillates beautifully. In fact, that is just the problem; it oscillates so strongly that it is difficult to get a good dip. VK5BR's solution was to connect a selected resistor in parallel with each coil to reduce the oscillation amplitude.

I decided to go a different route. The resistor kills the coil Q, which reduces the sensitivity and widens the bandwidth in wavemeter mode. Also, P-channel JFETs are becoming hard to find. Finally, I wanted to eliminate the internal calibration potentiometer that sets the operating point of the oscillator. Figure 1 shows the completed dipper, Figure 2 is a view inside the case and Figure 3 pictures the complete coil set. The schematic and parts list are shown in Figure 4. A pair of source-coupled N-channel JFETs (Q1 and Q2) form the oscillator portion of the circuit. Unlike some GDO designs, no RF chokes are required. Because of self-resonances, chokes don't maintain a high impedance over the wide range of frequencies required for a GDO.[6] This unit has no false dips anywhere in the frequency range.

Q4, a bipolar 2N3904 transistor, serves a dual purpose. Its base-emitter junction acts as the RF detector. Further, it amplifies the rectified current flowing in the base and sends it to emitter-follower Q5, a 2N2907 transistor. I used a 200 µA meter because that was what I found in my junk box. If you use another value, change the value of R5 using the formula $R5 = (1.5 / I_m) - R_m$, where $I_m$ is the full-scale meter current and $R_m$ is the meter resistance. Be careful when measuring meter internal resistance. Measure the test current of your ohmmeter (on the scale you will use) with another meter. A sensitive meter can be damaged by excessive measurement current, although most newer digital meters use test currents in the range of 50-200 µA for their ohmmeter scales. Transistor Q3 is a JFET source-follower amplifier for the RF output connector. The RF output may be used as a signal source or to drive a frequency counter for more accurate frequency readout.

In detector mode, the power to the oscillator and RF buffer is turned off. Q4 detects and amplifies any signals picked up by the coil. The meter sensitivity control works in both oscillator and wavemeter mode and also controls the volume to the headphones. Battery current is approximately 3 mA with the oscillator on and zero in wavemeter mode, when no signal is being received.

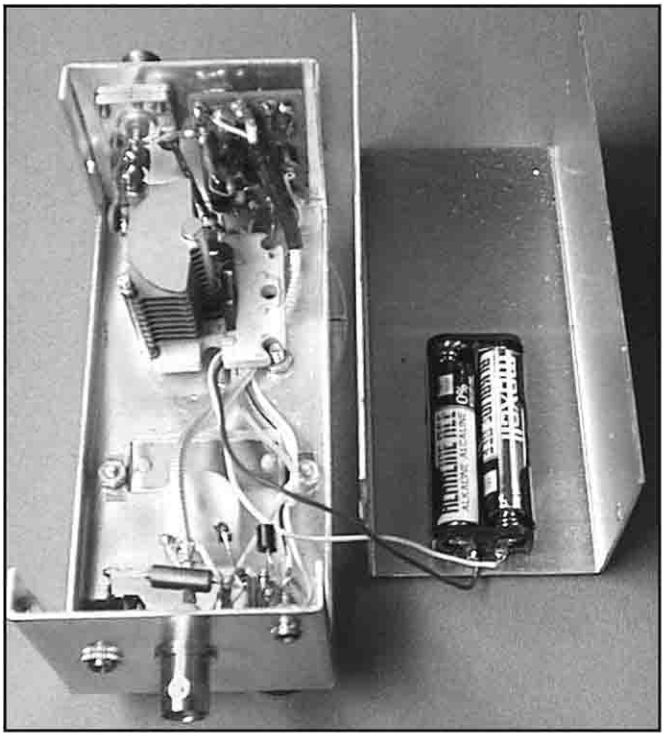


**Figure 2—Photograph of the GDO with the cover off. The tuning capacitor was oriented for the shortest possible connection to the coil connector.**

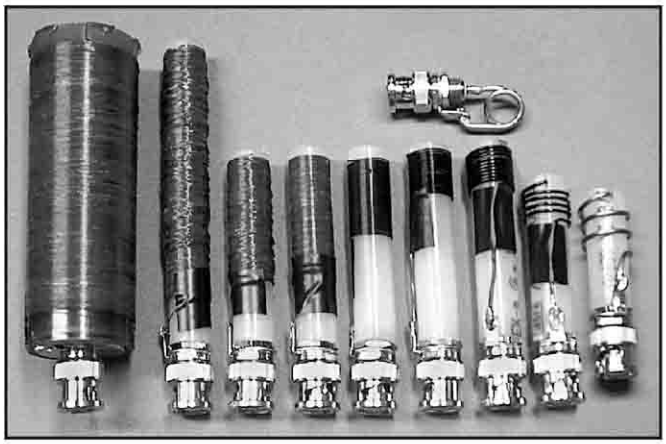


**Figure 3—This is how the coils looked before the heat-shrink tubing was applied. Small pieces of electrical tape hold the turns in place during construction.**

## Construction

This project was built almost entirely from junkbox parts. The parts list in Figure 4 consists of parts selected to be similar to those used in the prototype. Feel free to substitute what you have on hand. The exceptions are transistors Q1, Q2 and Q4, which should be the types specified or their equivalents.

The 2¼ × 2¼ × 5 inch chassis is a compromise; it's large enough to allow all parts to fit easily and small enough to fit comfortably in the hand. The coil was mounted off-center, both to allow the shortest connection to the tuning capacitor and to

May 2003 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

afford easier coil coupling to an external circuit.

The prototype used a perforated test board for the RF portion of the circuitry and a solder tie strip for the meter circuitry. A printed circuit board pattern and parts layout are available at **www.arrl.org/files/qst-binaries/gdo.zip**, should you choose this construction route (the GDO can be hand-wired). Note that the battery is connected with the positive terminal to ground, backwards from the normal arrangement.

The coil forms were made from "3/8 inch" water tubing purchased at the local hardware store. The inside diameter is actually slightly less than 3/8 inch, which makes a nice force-fit onto the 3/8 inch mounting threads of a male chassis-mount BNC connector. The lowest-frequency coil is wound on a large pill bottle with two 1-inch diameter aluminum washers at the connector mount for added strength.

First run the wire through the center of each form and solder it to the BNC connector's center conductor. Then press the form onto the connector threads and cut a small notch in

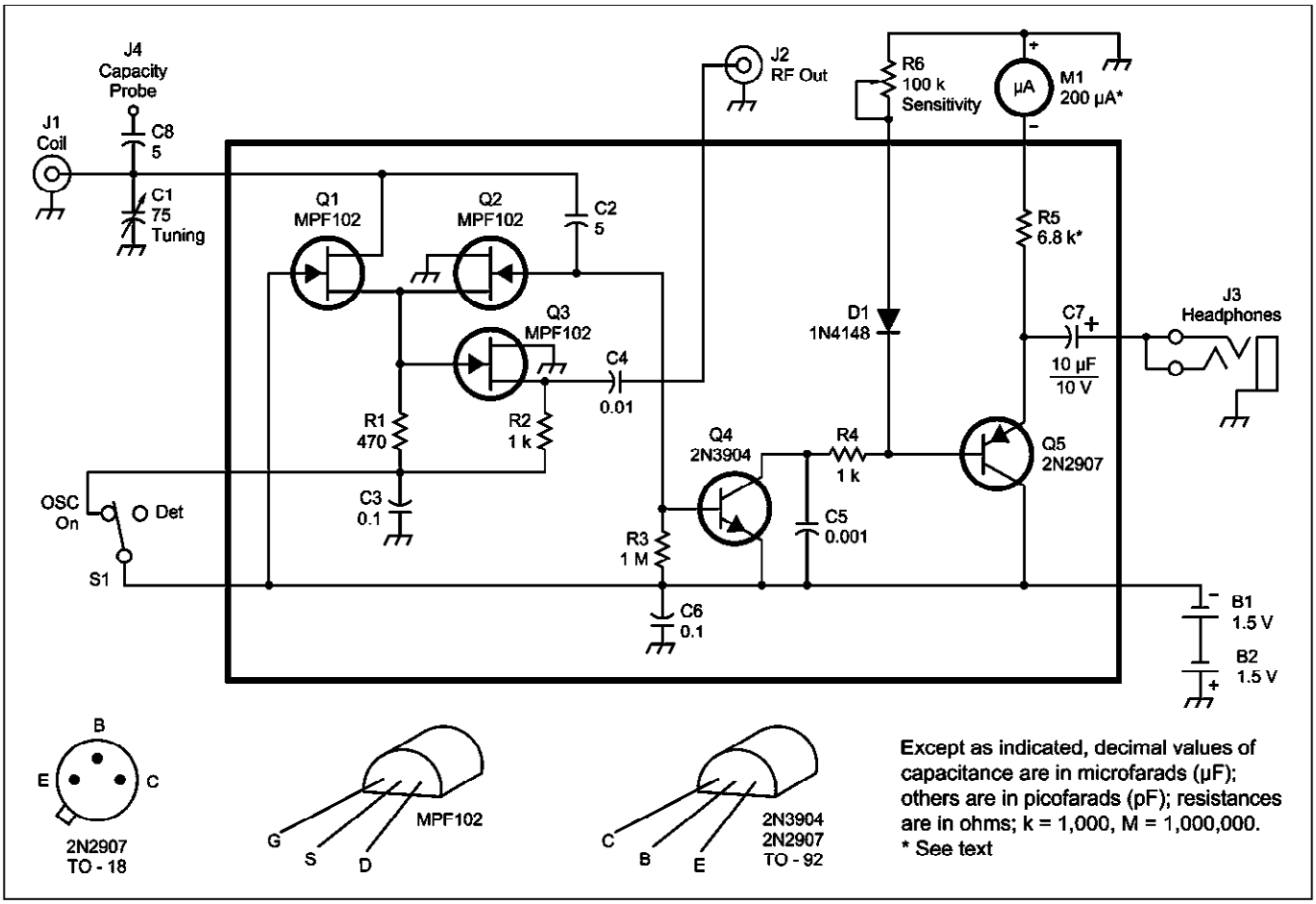



**Figure 4—Schematic diagram of the GDO. Part numbers listed are from Ocean State Electronics (OSE) (800-866-6626, www.oselectronics.com), Alltronics (408-847-0033, www.alltronics.com), Digi-Key (800-344-4539, www.digikey.com) and Mouser Electronics (800-346-6873, www.mouser.com). Resistors are available from all of the above sources.**

**B1, B2—1.5 V AA or AAA cell.**
**C1—75 pF variable, OSE AVC75 or 365 pF variable BC14400.**
**C2, C8—5 pF disc capacitor, Digi-Key P11408CT-NF, Mouser 75-10TCCV50, OSE CD5-5.**
**C3, C6—0.1 µF capacitor, Digi-Key BC1084CT-ND, Mouser 21RX310.**
**C4—0.01 µF capacitor, Digi-Key BC1078CT-ND, Mouser 21RX410.**
**C5—0.001 µF capacitor, Digi-Key BC1072CT-ND, Mouser 21RX510.**
**C7—10 µF, 10 V capacitor, Digi-Key P5134-ND, Mouser 140-XRL10V10, OSE CER10-25.**
**D1—1N4148 diode, Digi-Key 1N4148DICT-ND, Mouser 512-1N4148, OSE 1N4148.**
**J1, J2—BNC female chassis-mount connector, Alltronics CB122, Digi-Key ARFX1062-ND.**
**J3—3.5 mm stereo phone jack, Digi-Key SC1094-ND, Mouser 161-3402.**
**J4—Phone tip jack, Mouser 530-105-0802-1; phone tip plug, Mouser 530-105-0302-2.**
**M1—0-1 mA, 1 5/8 inch square panel meter, OSE 60-158 or 0-200 µA, 1 1/2 × 7/8 inch edge-mount panel meter MS200 (see text).**
**Q1, Q2, Q3—MPF102 transistor, JFET, Digi-Key MPF102-ND, Mouser 512-MPF102.**
**Q4—2N3904 transistor, Digi-Key 2N3904-ND, Mouser 610-2N3904.**
**Q5—2N2907A transistor, Digi-Key PN2N2907-ND, Mouser 610-2N2907A.**
**R1—470 Ω, ¼ W resistor.**
**R2, R4—1 kΩ, ¼ W resistor.**
**R3—1 MΩ, ¼ W resistor.**
**R5—See text.**
**R6—100 kΩ potentiometer, linear taper, Mouser 313-1000-100K.**
**S1—SPST toggle switch, Digi-Key EG2350-ND, Mouser 1055-TA1120.**

**Misc**

**#18, #22 and #26 gauge enameled wire, Mouser 501-MW18H-1LB, 501-MW22H-1LB, 501-MW26H-1LB, respectively.**
**10 BNC male chassis-mount connector (for coils), Alltronics CB111.**
**Dual AA or AAA battery holder, Digi-Key BC2AAAW-ND, BC2AAW-ND, Mouser 122-0421.**
**2¼″ × 2¼″ × 5″ chassis, Digi-Key HM611-ND, Mouser 563-CU-3004A, OSE 1411KU.**
**3 ft 3/8″ PVC water tubing for coils.**

May 2003 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

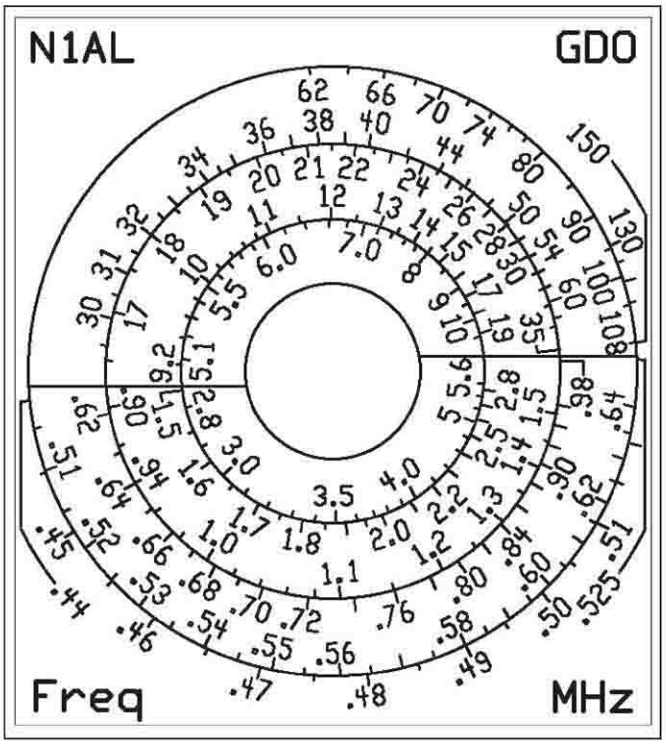



**Figure 5—The dial scale for the prototype GDO. If you make your own GDO you will almost certainly need to lay out your own scale.**

the other end for the wire to go through. After winding the coil, cut and tin the wire end and solder it to the ground lug mounted on the connector. I covered each coil with a layer of heat-shrink tubing or electrical tape to hold the turns in place and to protect the wire. Figure 3 shows the coils before the heat-shrink tubing was added.

Winding data is listed in Table 1. Unless you happen to duplicate my unit exactly, the frequency range of each coil is likely to be different. However, this data can be used as a starting point for figuring out your own coil designs.

The wire for the two highest-frequency coils is bare copper, salvaged from scraps of Romex house wiring. The other coils are wound with enamel-insulated magnet wire. The exact wire gauge is not critical (although it may affect the number of turns required). A good source for fine-gauge wire for the low-frequency coils is a deflection yoke removed from the cathode-ray tube of a defunct television set.

The three smallest coils (29.5-150 MHz) were space-wound to the lengths listed. The middle 4 coils (2.8-35 MHz) and the largest were close-wound. On the 0.9-1.5 and 1.5-2.8 MHz coils, there were too many turns to fit in a single layer so I "scramble" wound them by overlapping turns, a few at a time, as I wound the coil. It would be neater to lay the wire in two overlapping close-wound layers, but that increases the interwinding capacitance which can cause spurious resonances and reduced tuning range.

I extended the range of the lowest-frequency coil by connecting it to the GDO through a BNC "T" connector with a pair of 53 pF capacitors attached with clip leads. (Of course it also would have been possible to wind another coil, but I only had one large pill bottle.) Even for the other coils, adding extra capacitance is a useful trick because it slows down the tuning rate, handy for measuring narrow-band devices like crystals.[7]

With the smallest coil, the GDO oscillates over only a small portion of the capacitor tuning range. Fortunately, it happens to cover the two-meter ham band. With the second-smallest coil, the oscillation amplitude drops off at the low end, but it is usable down to about 62 MHz. The circuit would probably work better at VHF if it were constructed with a lower-inductance tuning capacitor.

The small 75 pF tuning capacitor gives only about a 2:1 tuning range. The advantage is that tuning is not so critical and the frequency dial is easier to read. The disadvantage is that it takes more coils to cover the desired frequency range. If you substitute a 365 pF AM broadcast radio tuning capacitor, you should get better than 3:1 tuning range from each coil.

The tuning dial was made from a circular piece of ¼ inch clear plastic glued to a knob. Making the diameter slightly larger than the chassis width allows one-handed operation using your thumb to adjust the tuning. The scale, cemented to the chassis under the clear plastic dial, was drawn using a computer graphics program. Measuring the calibration data and designing the scale were very time-consuming. A better solution would be to include a built-in frequency counter. A 4 or 5-digit readout would be accurate enough since a GDO's frequency stability is not good enough to make a high-accuracy readout useful. Figure 5 illustrates the homebrew dial.

## Operation

To measure the resonant frequency of a tuned circuit, switch to oscillator mode and adjust the meter sensitivity to about ¾ scale. Then orient the GDO coil close to, and approximately parallel to, the coil under test and tune the dial until you get a strong dip on the meter. Tune slowly or you may miss the dip. Overly close coupling to the circuit under test causes such a strong dip that the oscillator is pulled far off frequency. For the most accurate measurement, use close coupling to find the approximate frequency, but then move the GDO coil farther away until the dip is barely visible and retune. Figure 6 shows

**Table 1**
**Coil Winding Data for the GDO**

| *Wire Gauge* | *Form Diameter* | *Form Length* | *Coil Length* | *Number of Turns* | *Frequency Range* |
|---|---|---|---|---|---|
| #12 | 0.375" | — | 0.125" | 2 | 130-150 MHz |
| #14 | 0.5" | 1.5" | 0.5" | 3 | 62-108 MHz |
| #18 | 0.5" | 2.0" | 0.4" | 5 | 29.5-62 MHz |
| #18 | 0.5" | 2.5" | 0.5" | 10½ | 16.5-35 MHz |
| #22 | 0.5" | 2.5" | 0.6" | 21 | 9.2-19 MHz |
| #26 | 0.5" | 2.5" | 1.0" | 46 | 5.1-10.5 MHz |
| #30 | 0.5" | 2.5" | 1.5" | 100 | 2.8-5.6 MHz |
| #30 | 0.5" | 2.5" | 1.6" | 180 | 1.5-2.8 MHz |
| #30 | 0.5" | 3.9" | 3.3" | 390 | 0.9-1.5 MHz |
| #30 | 1.25" | 3.9" | 3.4" | 230 | 620-980 kHz |
| | | | | (+53 pF) | 505-640 kHz |
| | | | | (+106 pF) | 440-525 kHz |

May 2003 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

the dip meter in use, measuring the frequency of an amplifier tank circuit.

Coupling to toroids or shielded coils can be difficult. One solution is to connect a wire from the capacitance probe to the "hot" end of the circuit to be tested. For looser coupling, just place the wire close to the circuit under test. When measuring the inductance of a toroid with a test capacitor, the capacitor leads often form enough of a loop to allow coupling to the GDO coil. Some authors recommend coupling the GDO to a one-turn loop through the toroid, but that makes the toroid act like a transformer with a shorted secondary and that changes its inductance.

Antenna measurements are best made at the high-current point of the antenna conductor. For a half-wave dipole, this is near the center. Orient the coil perpendicular to the conductor for maximum coupling. Be sure to short the feed point of the antenna before making the measurement. If you can't find the dip, make the shorting wire into a 1-turn loop for better coupling to the GDO coil. You should also see resonance at the odd harmonic frequencies as well as the fundamental.

To measure the electrical length of a transmission line, couple the GDO to a small wire loop; connect it to one end of the line and leave the other end disconnected. (For best accuracy use the smallest loop that gives sufficient coupling.) The line is ¼ wavelength long at the lowest resonant frequency, so the electrical length in meters is 75/f, where f is the resonant frequency in MHz. Again, you will also see resonance at the odd harmonics.

The voltage level at the RF output connector varies from coil to coil but typically runs about 250 mV RMS into an open circuit and 50 mV RMS into a 50 Ω load. That is sufficient to drive a typical frequency counter or serve as a test signal when troubleshooting.

A GDO is like a TV remote control—you don't know you need it until you have it, and then you wonder how you ever got along without it. One simple instrument can do many of the measurements of a bench full of expensive RF test equipment. I've only touched on a few of the many and varied applications for this versatile instrument.[3,7,8]



**Figure 6—N3SU grid-dipping her linear amplifier's tank coil. Note that the high-voltage power supply is not only off but is disconnected from the amplifier.**

**Notes**

[1]C. F. (Bud) Bane, W6WB, "...About Grid-Dip Oscillators," *CQ*, Mar 1947, p 13. This seems to be the first grid dip oscillator design in the amateur literature. It was a portable battery-operated unit using a type 3A5 vacuum tube.

[2]*The 2003 ARRL Handbook,* Fig 6.46—Inductive and capacitive reactance vs frequency, p 6.27.

[3]B. Clark, WB4OBZ, "The Art of Dipping," *QST*, Jan 1974, p 16.

[4]L. Butler, VK5BR, "A Dip Meter using the Lambda Negative Resistance Circuit," *Amateur Radio*, Jan 1997, p 15. *Amateur Radio* is the journal of The Wireless Institute of Australia. For North American readers who may not have ready access to this magazine, the article is posted at **users.tpg.com.au/users/ldbutler/NegResDipMeter.htm**.

[5]H. Olson, W6GXN, "A New Look at Dip Meters," *Ham Radio*, Aug 1981, pp 25-28. This is a good survey of different dip meter circuit designs.

[6]The difficulty of designing a wide-band choke is discussed in F. Lewis, W1LKV, "Anatomy of a Solid-State Dipper," *QST*, Dec 1972, p 23.

[7]This trick is one of several described in R. Johns, W3JIP, "Add-Ons For Greater Dipper Versatility," *QST*, Feb 1981, p 37.

[8]M. Bradley, K6TAF, "What Can You Do with a Dip Meter?" *QST*, May 2002, p 65. A good overview on how to use a dip meter.

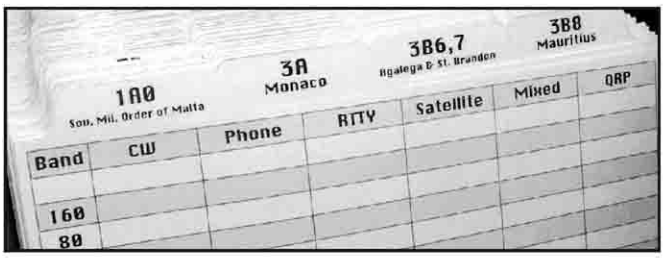
*Al Bloom, N1AL, was first licensed while in high school, in 1968. He received a Bachelor's degree in physics, then a Master's degree in electrical engineering, the latter while working as a W1AW operator. While at the ARRL he co-authored both the* License Manual *and the* Operating Manual. *He designed amateur and commercial equipment for the R. L. Drake Company and has spent the last 23 years at Hewlett-Packard, designing RF and microwave test equipment. Al has written several articles for* QST, *including two cover plaque award winners. Mainly a CW operator, Al is on nearly all of the amateur frequencies from 1.8 to 2450 MHz, including satellite operation. He is a life member of the ARRL and an ARRL Technical Advisor. You can contact the author at* **n1al@arrl.net**. QST

# NEW PRODUCTS

## QSL INDEX DIVIDERS FROM RADIO WAREHOUSE

◊ Attention award chasers: Radio Warehouse has introduced a set of QSL index dividers to sort your cards in a file. Radio Warehouse has also introduced a WAS set of dividers, produced in the same familiar format as the original DXCC edition (see photo). Both sets include Mylar reinforced tabs with a record keeping grid on each card. The dividers are sized to fit in any standard 5 × 8 index box, available at office supply stores.

Price: $19.95 + shipping. For more information, contact Radio Warehouse, PO Box 77001, Charlotte, NC 28271-7000; tel 704-321-2300; **www.radio-warehouse.com**. QST



# NOISE: Measurement and Generation

Paul Wade W1GHZ © 1996, 2010

## Introduction

As anyone who has listened to a receiver suspects, everything in the universe generates noise. In communications, the goal is to maximize the desired signal in relation to the undesired noise we hear. In order to accomplish this goal, it would be helpful to understand where noise originates, how much our own receiver adds to the noise we hear, and how to minimize it.

It is difficult to improve something unless we are able to measure it. Measurement of noise in receivers does not seem to be clearly understood by many amateurs, so I will attempt to explain the concepts and clarify the techniques, and to describe the standard "measure of merit" for receiver noise performance: "noise figure." Most important, I will describe how to build your own noise generator for noise figure measurements.

A number of equations are included, but only a few need be used to perform noise figure measurements. The rest are included to as an aid to understanding, with, I hope, enough explanatory text for everyone.

## Noise

The most pervasive source of noise is thermal noise, due to the motion of thermally agitated free electrons in a conductor. Since everything in the universe is at some temperature above absolute zero, every conductor must generate noise.

Every resistor (and all conductors have resistance) generates an *rms* noise voltage:

$$e = \sqrt{4kTRB}$$

where $R$ is the resistance, $T$ is the absolute temperature in degrees K, $B$ is the bandwidth in Hertz, and $k$ is Boltzmann's constant, $1.38 \times 10^{-23}$ joules / K.

Converting to power, $e^2/R$ , and adjusting for the Gaussian distribution of noise voltage, the noise power generated by the resistor is:

$Pn = kTB$ (watts)

which is independent of the resistance. Thus, all resistors at the same temperature generate the same noise power. The noise is white noise, meaning that the power density does not vary with frequency, but always has a power density of kT watts/Hz. More important is that the noise power is directly proportional to absolute temperature T, since k is a constant. At the nominal ambient temperature of 290 K, we can calculate this power; converted to dBm, we get the familiar -174 dBm/Hz. Just multiply by the bandwidth in Hertz to get the available noise power at ambient temperature. The choice of 290 K for ambient might seem a bit cool, since the equivalent 17° C or 62° F would be a rather cool room temperature, but 290 makes all the calculations come out to even numbers.

The *instantaneous noise voltage* has a Gaussian distribution around the *rms* value. The Gaussian distribution has no limit on the peak amplitude, so at any instant the noise voltage may have any value from -infinity to +infinity. For design purposes, we can use a value that will not be exceeded more than

0.01% of the time. This voltage is 4 times the *rms* value, or 12 dB higher, so our system must be able to handle peak powers 12 dB higher than the average noise power if we are to measure noise without errors.[1]

# Signal to Noise Ratio (S/N)

Now that we know the noise power in a given bandwidth, we can easily calculate how much signal is required to achieve a desired signal to noise ratio, S/N. For SSB, perhaps 10 dB S/N is required for good communications; since ambient thermal noise in a 2.5 kHz bandwidth is -140 dBm, calculated as follows:

$$Pn = kTB = 1.38 \times 10^{-23} \times 290 \times 2500 = 1.0 \times 10^{-17} \text{ watts}$$

$$dBm = 10log\ (\ Pn \times 1000\ )\ \text{[multiplying the power by 1000 to get milliwatts]}$$

The signal power must be 10 dB larger, so minimum signal level of -130 dBm is required for a 10 dB S/N. This represents the noise and signal power levels at the antenna. We are then faced with the task of amplifying the signal without degrading the signal to noise ratio.

# Noise Temperature

Any amplifier will add additional noise. The input noise $N_i$ per unit bandwidth is $kT_g$ is amplified by gain $G$ to produce an output noise of $kT_gG$. The additional noise, $kT_n$ is added to produce a total noise output power $N_o$:

$$N_o = kT_gG + kT_n$$

To simplify future calculations, we pretend that the amplifier is noise-free but has an additional noise generating resistor of temperature $T_e$ at the input, so that all sources of noise are inputs to the amplifier. Then the output noise is:

$$N_o = kG\ (\ T_g + T_e\ )$$

where $T_e$ is the *Noise Temperature* of the excess noise contributed by the amplifier. The noise added by an amplifier is then $kGT_e$, which is the fictitious noise source at the input amplified by the amplifier gain.

# Cascaded Amplifiers

If several amplifiers are cascaded, the output noise $N_o$ of each becomes the input noise $T_g$ to the next stage and we can create a large equation for the total. After removing the original input noise term, we are left with the added noise:

$$N_{added} = (k\ T_{e1}G_1G_2 \ldots\ G_N) + (kT_{e2}G_2 \ldots\ G_N) + \ldots + (kT_{eN}G_N)$$

Substituting in the total gain $G_T = (G_1G_2 \ldots\ G_N)$ results in the total excess noise:

$$T_{eT} = T_{e1} + T_{e2}/G_1 + T_{e2}/G_1G_2 + \ldots + T_{eN}/(G_1G_2 \ldots\ G_{N-1})$$

showing that the noise of each succeeding stage reduced by the gain of all preceding stages. Clearly, if the gain of the first stage, $G_1$, is large, then the noise contributions of the succeeding stages are not

significant. This is why we concentrate our efforts on improving the first amplifier or preamplifier.

# Noise Figure (NF)

The *noise figure* (NF) of an amplifier is the logarithm of the ratio (so we can express it in dB) of the total noise output of an amplifier with an input $T_g$ of 290 K to the noise output of an equivalent noise-free amplifier. A more useful definition is to calculate it from the excess temperature $T_e$:

$$NF = 10log\ (1 + Te\ /T0)\ (dB)\ @\ T0 = 290\ K$$

If the NF is known, then $T_e$ may be calculated after converting the NF to a ratio, F:

$$Te = (F - 1)\ T0$$

Typically, $T_e$ is specified for very low noise amplifiers, where the NF would be fraction of a dB, and NF is used when it seems a more manageable number than thousands of K.

# Losses

We know that any loss or attenuation in a system reduces the signal level. If attenuation also reduced the noise level, then we could suppress thermal noise by adding attenuation. We know intuitively that this can't be true. The answer is that the attenuator or any lossy element has a physical temperature, $T_x$ , which contributes noise to the system while the input noise is being attenuated. The output noise after a loss L (ratio) is:

$$Tg' = Tg\ /L + [(L-1)/L]\ Tx$$

If the source temperature $T_g$ is higher than the attenuator temperature $T_x$ , then the noise contribution is the familiar result found by simply adding the loss in dB to the NF. However, for low source temperatures the degradation can be much more dramatic. If we do a calculation for the affect of 1 dB of loss (L = 1.26) on a $T_g$ of 25 K:

$$Tg' = 25/1.26 + (0.26/1.26)\ x\ 290 = 80\ K$$

The resultant $T_{g'}$ is 80 K, a 5 dB increase in noise power (or 5 dB degradation of signal to noise ratio). Since noise power = $kT$ and $k$ is a constant, the increase is the ratio of the two temperatures 80/25, or in dB, $10log(\ 80/25\ ) = 5$ dB.

# Antenna Temperature

How can we have a source temperature much lower than ambient? If an antenna, assumed to be lossless, is receiving signals from space, rather than the warm earth, then the background noise is much lower. The background temperature of the universe has been measured as about 3.2 K. An empirical number for a 10 GHz antenna pointing into clear sky is about 6 K, since we must always look through attenuation and temperature of the atmosphere.[2] The figure will vary with frequency, but a good EME antenna might have a $T_g$ of around 20 K at UHF and higher frequencies.

A couple of examples of actual antenna might bring all of this together.[3]

1 A 30 inch conventional dish at 10 GHz, with measured gain of 36.4 dBi and efficiency of 64%. The estimated spillover efficiency is 87% for a 10 dB illumination taper. With the dish pointing at a high elevation as shown in Figure 1, perhaps half of the spillover is illuminating earth at 290 K, which adds an estimated 19 K to the 6 K of sky noise, for a total of 25 K. In a 500 Hz bandwidth, the noise output is -157.6 dBm.

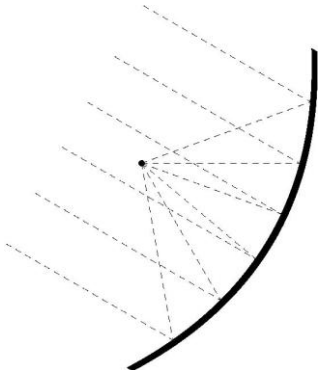


**Figure 1. Parabolic Dish Antenna Aimed at Satellite**

2 An 18 inch DSS offset-fed dish at 10 GHz, with measured gain of 32.0 dB and efficiency of 63%. The spillover efficiency should be comparable, but with the offset dish pointing at a high elevation as shown in Figure 2, far less of the spillover is illuminating warm earth. If we estimate 20%, then 8 K is added to the 6 K of sky noise, for a total of 14 K. In a 500 Hz bandwidth, the noise output is -160 dBm.

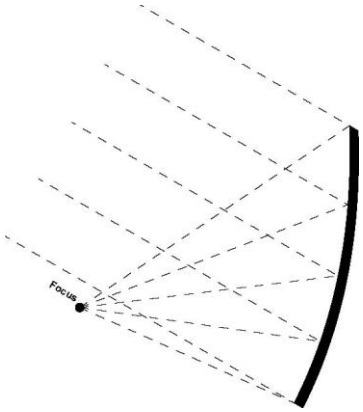



**Figure 2. Offset Parabolic Dish Antenna Aimed at Satellite**

The larger conventional dish has 2.4 dB higher noise output, but 4.4 dB higher gain, so it should have 2.0 dB better signal to noise ratio than the smaller offset dish when both are pointing at high elevations.

However, the while the offset dish is easy to feed with low loss, it is convenient to feed the conventional dish through a cable with 1 dB of loss. Referring back to our loss example above, the noise temperature after this cable loss is 80 K. In a 500 Hz bandwidth, the noise output is now -152.6 dBm, 7.4 dB worse than the offset dish. The convenience of the cable reduces the signal to noise ratio by 5 dB, making the larger conventional dish 3 dB worse than the smaller offset dish. Is it any wonder that the DSS dishes sprouting on rooftops everywhere are offset-fed?

If the dishes are pointed on the horizon for terrestrial operation, then the situation is much different. At least half of each antenna pattern is illuminating warm earth, so we should expect the noise temperature to be at least half of 290 K, or about 150 K. Adding 1 dB of loss increases the noise temperature to 179 K, a

1 dB increase. At the higher noise temperatures, losses do not have a dramatic effect on signal to noise ratio. In practice, the antenna temperature on the horizon may be even higher, since the upper half of the pattern must take a much longer path through the warm atmosphere, which adds noise just like any other loss.

# Image Response

Most receiving systems use at least one frequency converting mixer which has two responses, the desired frequency and an image frequency on the other side of the local oscillator. If the image response is not filtered out, it will add additional noise to the mixer output. Since most preamps are broadband enough to have significant gain (and thus, noise output) at the image frequency, the filter must be placed between the preamp and the mixer. The total NF including image response is calculated:

$$NF = 10log[( 1 + T_e/T_0)( 1 + G_{image}/G_{desired}) ]$$

assuming equal noise bandwidth for desired and image responses. Without any filtering, $G_{image} = G_{desired}$ so $G_{image}/G_{desired} = 1$, doubling the noise figure which is the same as adding 3 dB. Thus, without any image rejection, the overall noise figure is at least 3 dB regardless of the NF of the preamp. For the image to add less than 0.1 dB to the overall NF, a quick calculation shows that the gain at the image frequency must be at least 16 dB lower than at the operating frequency.

# Noise Figure Measurement

So far we have discussed the sources of noise, and a figure of for evaluating a receiving system's response to noise. How can we measure an actual receiver?

The noise figure of a receiver is determined by measuring its output with two different noise levels, $T_{hot}$ and $T_{cold}$, applied at the input. The ratio of the two output levels is referred to as the "Y-factor". Usually, the ratio is determined from the difference in dB between the two output levels, $Y_{db}$:

$$Y_{(ratio)} = log^{-1} (Y_{db} / 10)$$

Then the receiver $T_e$ may be calculated using $Y_{(ratio)}$:

$$T_e = (T_{hot} - YT_{cold}) / (Y-1)$$

and converted to noise figure:

$$NF = 10log( 1 + T_e/T_0) \text{ (dB) where } T_0 = 290 \text{ K}$$

The two different noise levels may be generated separately, for instance by connecting resistors at two different temperatures. Alternatively, we could use a device that can generate a calibrated amount of noise when it is turned on. When such a device is turned off, it still generates noise from its internal resistance at $T_{cold}$, the ambient temperature (290 K); usually this resistance is 50 ohms, to properly terminate the transmission line which connects it to the receiver. When the noise generator is turned on, it produces excess noise equivalent to a resistor at some higher temperature at $T_{hot}$.

The noise produced by a noise source may be specified as the *Excess Noise Ratio* ($ENR_{dB}$), the dB difference between the cold and the equivalent hot temperature, or as the equivalent temperature of the excess noise, $T_{ex}$, which is used in place of $T_{hot}$ in the previous equation. If the ENR is specified, then the calculation is:

$$NF_{dB} = ENR_{dB} - 10log(Y_{(ratio)} - 1)$$

The terms $T_{ex}$ and ENR are used rather loosely; assume that a noise source specified in dB refers to $ENR_{dB}$, while a specification in degrees or K refers to $T_{ex}$.

An automatic noise figure meter, sometimes called a PANFI (for Precision Automatic Noise Figure Meter), turns the noise source on and off at a rate of about 400 Hz and performs the above calculation electronically.[4] A wide bandwidth is required to detect enough noise to operate at this rate; a manual measurement using a narrowband communications receiver would require the switching rate to be less than one Hz, with some kind of electronic integration to properly average the Gaussian noise.

Noise figure meters seem to be fairly common surplus items. The one most recently in production, the HP or Agilent 8970, measures both noise figure and gain, but commands a stiff price. Current instruments that measure noise figure include many other features at a price well beyond amateur budgets.

AIL (later AILTECH or Eaton) made several models; the model 2075 measures both NF and gain, while other models are NF only. The Model 75 (a whole series whose model numbers start with 75) shows up frequently for anywhere from $7 to $400, typically $25 to $50, and performs well. Every VHFer I know has one, with most of them waiting for a noise source to be usable. Earlier tube models, like the AIL 74 and the HP 340 and 342, have problems with drift and heat, but they can also do the job.

Another alternative is to build a noise figure meter.[5]

# Using the Noise Figure Meter

I'll describe the basic procedure using the Model 75; others are similar, but the more complex instruments will require studying the instruction manual.

Input to almost all noise figure meters is at 30 MHz, so a frequency converter is required (some instruments have internal frequency converters; except for the HP 8970, I'd avoid using this feature). Most ham converters with a 28 MHz IF work fine, unless the preamp being measured is so narrowband that a MHz or two changes the NF. The input is fairly broadband, so LO leakage or any other stray signals can upset the measurement – this has been a source of frustration for many users. There are two solutions: a filter (30 MHz low-pass TVI filters are often sufficient) or a tuned amplifier at 30 MHz. Since a fair amount of gain is required in front of the noise figure meter, an amplifier is usually required anyway.

A noise source (which we will discuss in detail later) is connected to the rear of the instrument: a BNC connector marked "DIODE GATE" provides +28 volts for a solid-state noise source, and high voltage leads for a gas tube noise source are also available on many versions. The noise figure meter switches the noise source on and off. The noise output coax connector of the noise source is connected to the receiver input.

The Model 75 has four function pushbuttons: *OFF, ON, AUTO,* and *CAL.* The *OFF* and *ON* positions are for manual measurements: *OFF* displays the detector output with the noise source turned off, and *ON* displays the detector output with the noise source turned on. If all is working, there should be more output in the *ON* position, and a step attenuator in the IF line may be used to determine the change in output, or Y-factor, to sanity-check our results. The knob marked "GAIN" is used to get the meter reading to a desirable part of the scale in the *OFF* and *ON* positions only; it has no effect on automatic measurements.

The *AUTO* position causes the instrument to turn the noise source on and off at about a 400 Hz rate and to calculate the NF from the detected change in noise. The Model 75 has a large green light near the

meter which indicates that the input level is high enough for proper operation – add gain until the light comes on. Then the meter should indicate a noise figure, but not a meaningful one, since we must first set the $ENR_{dB}$ using the *CAL* position. The lower scale on the meter is marked for from 14.5 to 16.5 dB of ENR; adjust the "CAL ADJ" knob until the reading in the *CAL* position matches the ENR of the noise source.

If the ENR of your noise source is outside the marked range, read the section below on homebrew noise sources.

Now that we have calibrated the meter for the ENR of the noise source, we may read the noise figure directly in the *AUTO* position. Before we believe it, a few sanity checks are in order:

1. Manual Y-factor measurement and calculation.
2. Insert a known attenuator between the noise source and preamp. The NF should increase by exactly the attenuation added.
3. Measure something with a known noise figure (known means measured elsewhere; a manufacturers claim is not necessarily enough).

Finally, too much gain in the system may also cause trouble, if the total noise power exceeds the level that an amplifier stage can handle without gain compression. Gain compression will be greater in the on state, so the detected Y-factor will be reduced, resulting in erroneously high indicated NF. The Gaussian distribution of the noise means that an amplifier must be able to handle 12 dB more than the average noise level without compression. One case where this is a problem is with a microwave transverter to a VHF or UHF IF followed by another converter to the 30 MHz noise figure meter, for too much total gain. I always place a step attenuator between the transverter and the converter which adjust until I can both add and subtract attenuation without changing the indicated noise figure.

One final precaution: noise figure meters have a very slow time constant, as long as 10 seconds for some of the older models, to smooth out the random nature of noise. If you are using the noise figure meter to "tweak" a receiver, *tune very slowly!*

# Sky Noise Measurement

Another way to measure noise figure at microwave frequencies is by measurement of sky noise and ground noise.[3,6] Sky noise is very low, around 6 K at 10 GHz, for instance, and ground noise is due to the ground temperature, around 290 K, so the difference is nearly 290 K. At microwave frequencies we can use a manageable antenna that is sharp enough that almost no ground noise is received, even in sidelobes, when the antenna is pointed at a high elevation. A long horn would be a good antenna choice.

The antenna is pointed alternately at clear sky overhead, away from the sun or any obstruction, and at the ground. The difference in noise output is the Y-factor; since we know both noise temperatures, the receiver noise temperature is calculated using the $Y_{(ratio)}$:

$$T_e = (T_{hot} - YT_{cold}) / (Y-1)$$

The latest version of my microwave antenna program, HDL_ANT, will make this calculation.[3] Since the measured Y-factor will be relatively small, this measurement will only be accurate for relatively low noise figures. On the other hand, low noise figures are the most difficult to measure accurately using other techniques.

A system for measuring sun noise was described by G3WDG which also works well for measuring noise figure from sky noise.[7] He built a 144 MHz amplifier with moderate bandwidth using MMICs and helical filters which amplifies the transverter output to drive a surplus RF power meter. The newer solid-state power meters are stable enough to detect and display small changes in noise level, and the response

is slow enough to smooth out flicker. Since my 10 GHz system has an IF output at 432 MHz, duplicating G3WDG's amplifier would not work. In the junk box I found some surplus broadband amplifiers and a couple of interdigital filters, and combined these to provide high gain with a few MHz bandwidth, arranged as shown in Figure 3. I found that roughly 60 dB of gain after the transverter was required to get a reasonable level on the power meter, while the G3WDG system has somewhat narrower bandwidth so more gain is required.

Several precautions are necessary:

1. Peak noise power must not exceed the level that any amplifier stage can handle without gain compression. Amplifiers with broadband noise output suffer gain compression at levels lower than found with signals, so be sure the amplifier compression point is at least 12 dB higher than the indicated average noise power.
2. Make sure no stray signals appear within the filter passband.
3. Foliage and other obstructions add thermal noise, which obscure the cold sky reading.
4. Low noise amplifiers are typically very sensitive to input mismatch, so the antenna must present a low VSWR to the preamp.

A noise figure meter could also be used as the indicator for the sky noise measurement, but a calibrated attenuator would be needed to determine the Y-factor. Using different equipment gives us an independent check of noise figure, so that we may have more confidence in our measurements.

Another choice for the indicator would be one of the software-defined radios. Those from Flex Radio Systems have accurate signal power readout, far better than a typical S-meter, and adjustable bandwidth and averaging time. Averaging for several seconds is needed to make the signal level stable with random noise.

W2IMU suggested that the same technique could be used for a large dish at lower frequencies.[8] With the dish pointing at clear sky, the feed horn is pointing at the reflector which shields it from the ground noise so it only sees the sky noise. If the feed horn is then removed and pointed at the ground, it will then see the ground noise.

Noise figure meters are convenient, but if you don't have one, the equipment for measuring sun and sky noise could also be used indoors with a noise source. The only complication is that the Y-factor could be much larger, pushing the limits of amplifier and power meter dynamic range

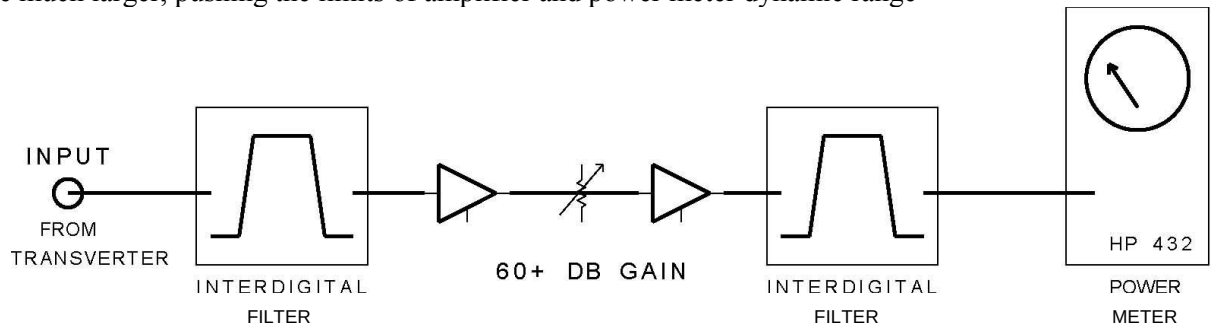



**Figure 3. Indicator for Sun Noise**

# Noise Sources

The simplest noise source is simply a heated resistor – if we know the temperature of the resistor, we can calculate exactly how much noise it is generating. If we then change the temperature, the noise output will change by a known amount. This would work if we could find a resistor with good RF properties whose value does not change with temperature, an unlikely combination. There are commercial units,

called Hot-Cold Noise Sources, with two calibrated resistors at different temperatures with low VSWR.

Typically, one resistor is cooled by liquid nitrogen to 77.3 K (the boiling point of nitrogen), while the other is heated by boiling water to 100°C, or 373.2 K. The preamp is connected to first one resistor, then the other; the difference in noise in noise output is the Y-factor. Using the $Y_{(ratio)}$, the preamp noise temperature is calculated:

$$T_e = (T_{hot} - YT_{cold}) / (Y-1)$$

Since the boiling point of pure liquids is accurately known, this type of noise generator can provide very accurate measurements. However, they are inconvenient to use, since the receiver must be connected directly to alternate resistors (the loss in an RF switch would significantly reduce the noise output and accuracy). Also, few amateurs have a convenient source of liquid nitrogen. The following three types of noise sources are commonly available and convenient to use.

## Temperature-limited vacuum tube diode

The noise output is controlled by the diode current, but is only accurate up to around 300 MHz due to limitations of the vacuum tube. These units generate around 5 dB of excess noise.

## Gas tube sources

The noise is generated by an ionized gas in the tube, similar to a fluorescent light – homebrew units have been built using small fluorescent tubes. The noise tubes use a pure gas, typically argon, to control the noise level. These units typically generate about 15 dB of excess noise.

Coaxial gas tube sources work up to around 2.5 GHz, and waveguide units to much higher frequencies. One problem using these is that a high voltage pulse is used to start the ionization (like the starter in a fluorescent light) which is coupled to the output in the coaxial units and is large enough to damage low-noise transistors. Since a noise figure meter turns the noise source on and off continuously, pulses are generated at the same rate.

Since waveguide acts as a high-pass filter, the starting pulses are not propagated to the output, so waveguide gas-tube noise sources are safe to use, though bulky and inconvenient. However, they could be used to calibrate a solid-state noise source.

Another problem with all gas tubes is that the VSWR of the noise source changes between the on and off states. If the source VSWR changes the noise figure of an amplifier, as is almost always the case, then the accuracy of the measurement is reduced.

## Solid-state noise sources

Reverse breakdown of a silicon diode PN junction in causes an avalanche of current in the junction which would rise to destructively high levels if not limited by an external resistance. Since current is "electrons in motion," a large amount of noise is generated. If the current density of the diode is constant, then the average noise output should also be constant; the instantaneous current is still random with a Gaussian distribution, so the generated noise is identical to thermal noise at a high temperature. Commercial units use special diodes designed for avalanche operation with very small capacitance for high frequency operation, but it is possible to make a very good noise source using the emitter-base junction of a small microwave transistor.

Typical noise output from an avalanche noise diode is 25 dB or more, so the output must be reduced to a usable level, frequently 15 dB of excess noise to be compatible with gas tubes or 5 dB of excess noise for more modern equipment. If the noise level is reduced by a good RF attenuator of 10 dB or more, then the source VSWR (seen by the receiver) is dominated by the attenuator, since the minimum return loss is

twice the attenuation. Thus, the change in VSWR as the noise diode is turned on and off is minuscule. Commercial noise sources consist of a noise diode assembly and a selected coaxial attenuator permanently joined in a metal housing, calibrated as a single unit.

## Homebrew Noise Sources

There are three components of a noise source: a noise generator, an attenuator, and the calibration data of ENR at each frequency. The most critical one is the attenuator; it is very important that the noise source present a very low VSWR to the preamp or whatever is being measured, since low-noise amplifiers are sensitive to input impedance, and even more important that the VSWR does not change when the noise source is turned on and off, since a change causes error in the measurement. Because an attenuator provides twice as many dB of isolation as loss (reflections pass through a second time), 10 dB or more of attenuation will reduce any change in VSWR to a very small value.

Commercial solid-state noise sources occasionally appear in surplus sources, usually at high prices but occasionally very cheap if no one knows what it is. I have found two of the latter, and one of them works! It produces about 25 dB of excess noise, which is too much to be usable. I went through my box of hamfest attenuators and found one which has excellent VSWR up to 10 GHz and 13 dB of attenuation. Mated with the noise source, the combination produces about 12 dB of excess noise — a very usable amount.

Finally, I calibrated it against a calibrated noise source for all ham bands between 50 MHz and 10 GHz; not exactly NTIS traceable, but pretty good for amateur work.

While noise sources are hard to locate, noise figure meters are frequent finds. If we could come up with some noise sources, all the VHFers who have one gathering dust could be measuring and optimizing their noise figure.

Several articles have described construction of homebrew noise sources, which work well at VHF and UHF, but not as well at 10 GHz.[9,10,11] All of them have the diode in a shunt configuration, with one end of the diode grounded. When I disassembled my defective commercial noise source (even the attenuator was bad), I found a bare chip diode in a series configuration – diode current flows into the output attenuator. Obviously I could not repair a chip diode, but I could try the series diode configuration. I found the smallest packaged microwave transistor available, some small chip resistors and capacitors, and soldered them directly on the gold-plated flange of an SMA connector with zero lead length, as shown in the photograph, Figure 4. We've all soldered components directly together in "dead-bug" construction; this is more like "fly-speck" construction.

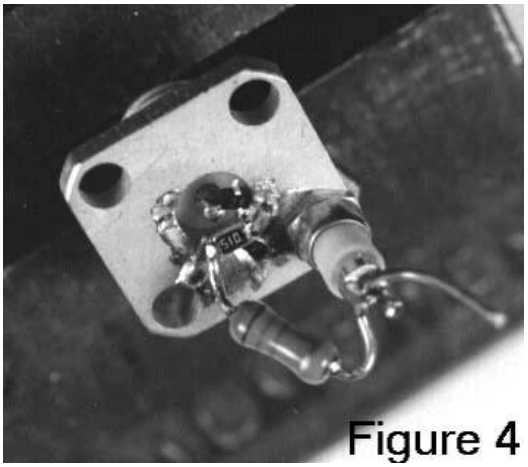


Figure 4

The schematic is shown in Figure 5**,** and it works at 10 GHz! I built several versions to evaluate reproducibility, and measured them at several ham bands from 30 MHz to 10 GHz, with results shown in Figure 6. All units were measured with the same 14 dB attenuator, so the diode noise generator output is 14 dB higher.

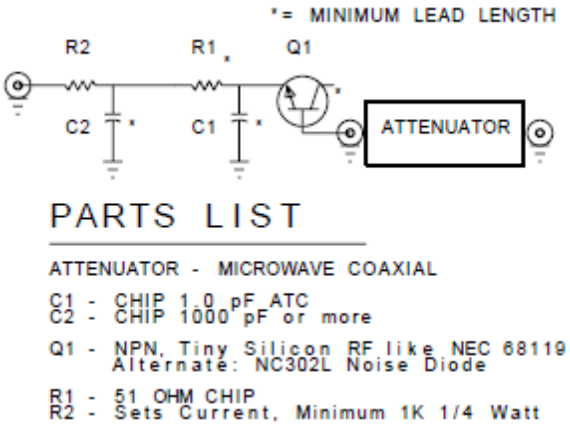



**Figure 5. Noise Source Schematic**

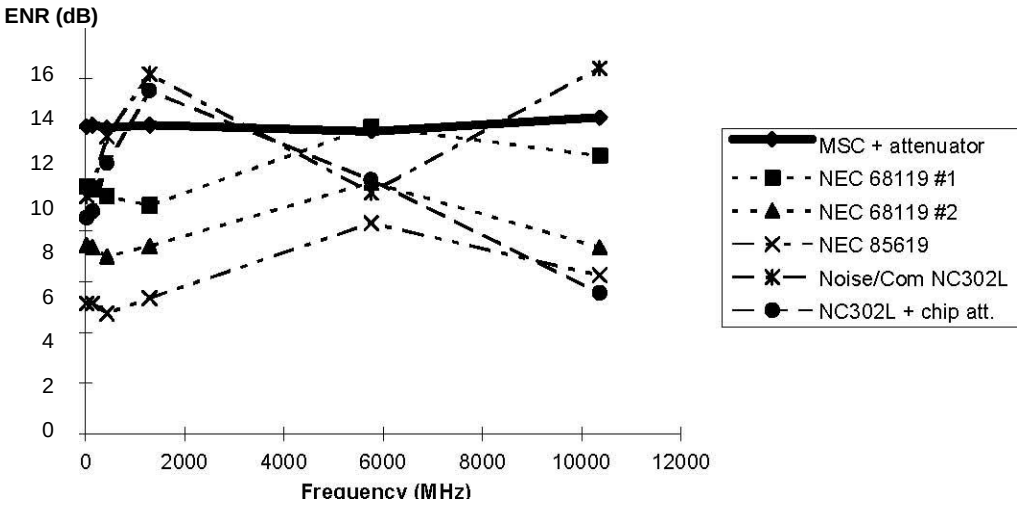



**Figure 6. Noise Source ENR**

(Later I found that the MIT Radiation Laboratory had described a noise source with a series diode 50 years ago, so we aren't giving away anyone's trade secrets.[12])

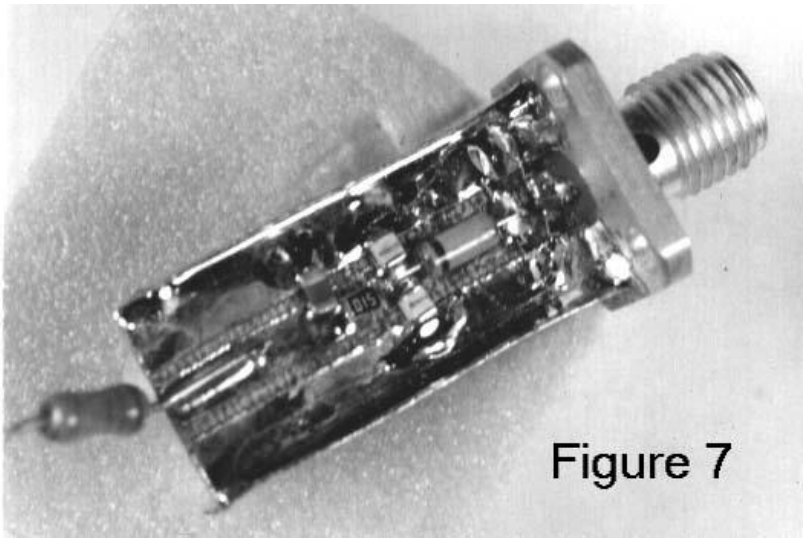
I then remembered that I had a commercial noise diode, a Noise/Com NC302L, which was used in a noise source described in *QST*, with the diode in the shunt configuration.[11] The diode is rated as working to 3 GHz, so, in the amateur tradition, I wanted to see if I could push it higher, using the series configuration. Since I didn't expect to reach 10 GHz, I increased the value of the bypass capacitor, but otherwise, it looks like the units in Figure 5. When I measured this unit, it not only worked at 10 GHz, but had more excess noise output than at lower frequencies, probably due to an unexpected resonance. The performance is shown in Figure 6 along with the other units.

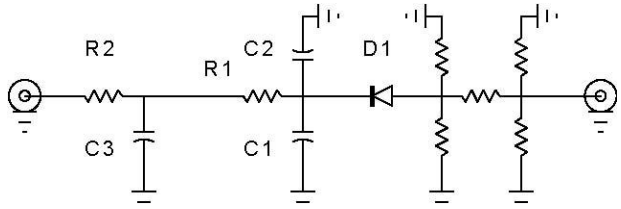
Also shown in Figure 6 is the output of my pseudo-commercial noise source; even with the external

attenuator, the excess noise output is pretty flat with frequency. Commercial units are typically specified at + or - 0.5 dB flatness. In Figure 6, none of the homebrew ones are that flat, but there is no need for it; as long as we know the excess noise output for a particular ham band, it is perfectly usable for that band.

All the above noise sources relied on a coaxial microwave attenuator to control the VSWR of the noise source. Attenuators are fairly frequent hamfest finds, but ones that themselves have good VSWR to 10 GHz are less common, and it's hard to tell how good they are without test equipment. An alternative might be to build an attenuator from small chip resistors. I used my *PAD.EXE* program to review possible resistor values, and found that I could make a 15.3 dB $\pi$ attenuator using only 140 ohm resistors if the shunt legs were formed by two resistors in parallel, a good idea to reduce stray inductance.[13] I ordered some "0402" size (truly tiny) chip resistors from DigiKey, more NC302L diodes from Noise/Com, and built the noise source shown in Figure 7 on a bit of Teflon PC board, cutting out the 50 ohm transmission line with an X-Acto knife. The schematic of the complete noise source is shown in Figure 8.

The chip resistor attenuator works nearly as well as an expensive coaxial one. The measured VSWR of two noise sources, one with the chip attenuator and the other with a coaxial attenuator, is shown in Figure 9. Curves are shown in both the OFF and ON states, showing how little the VSWR changes. The VSWR of the chip attenuator unit is 1.42 at 10 GHz, slightly over the 1.35 maximum specified for commercial noise sources, but still fine for amateur use.




PARTS LIST

C1 -CHIP 1.0 pF ATCC2 -CHIP 8.2 pF ATCC2 -CHIP 1000 pF or more

D1 -NC302L Noise Diode

R1 -51 OHM CHIP R2 -Sets Current, Minimum 1K 1/4 Watt R3-7 -140 OHM CHIP, 0402 SIZE

**Figure 8. Noise Source with Chip Attenuator**

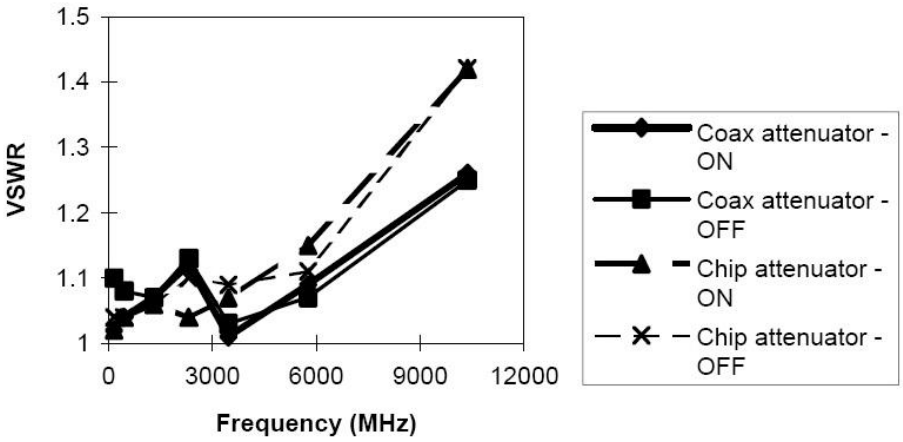



**Figure 9. VSWR of Homebrew Noise Sources**

# Noise Source Alignment

The only alignment requirement for a solid-state noise source is to set the diode current; the current is always set at the highest frequency of interest. A noise figure meter must be set up with converters, etc., for the highest frequency at which the noise source might be used, and set to display the detector output (*OFF* position on a Model 75). Then voltage from a variable DC power supply is applied to the noise diode through the 1k current-limiting resistor. The detector output should increase as the voltage (diode current) increases, reach a peak, then decrease slightly. The optimum current is the one that produces peak output at the highest frequency (I set mine at 10 GHz). Then additional resistance must be added in series with the current-limiting resistor so that the peak output occurs with 28 volts applied, so that the noise source may be driven by the noise figure meter. Once the proper resistor is determined and added, the DC end of the noise source is connected to the diode output of the noise figure meter, and the meter function set to *ON*. This should produce the same detector output as the power supply. Then the meter function is set to *AUTO*, and the meter should produce some noise figure indication, but not a calibrated one yet. However, it is good enough to tune up preamps – a lower noise figure is always better, even if you don't know how low it is.

# Noise Source Calibration

Much of the high price of commercial noise sources pays for the NTIS-traceable calibration. Building a noise source only solves part of the problem – now we need to calibrate it.

The basic calibration technique is to measure something with a known noise figure using the new noise source then calculate what ENR would produce the indicated noise figure. Fortunately, the calculation is a simple one involving only addition and subtraction; no fancy computer program required. Simply subtract the indicated noise figure, $NF_{indicated}$, from the known noise figure, $NF_{actual}$, and add the difference to the ENR for which the meter was calibrated, $ENR_{cal}$ :

$$ENR_{(noise\ source)} = ENR_{cal} + (NF_{actual} - NF_{indicated})$$

This procedure must be repeated at each frequency of interest; at least once for each ham band should be fine for amateur use.

The known noise figure is best found by making the measurement with a calibrated noise source, then substituting the new noise source so there is little opportunity for anything to change. Next best would be a sky noise measurement on a preamp. Least accurate would be to measure a preamp at a VHF conference or other remote location, then bring it home and measure it, hoping that nothing rattled loose on the way. If you can't borrow a calibrated noise source, it would be better to take your noise source elsewhere and calibrate it. Perhaps we could measure noise sources as well as preamps at some of these events.

# Using the noise source

Now that the ENR of the noise source has been calibrated, the noise figure calibration must be adjusted to match. However, the Model 75 in the *CAL* position has only two dB of adjustment range marked on the meter scale. Older instruments have no adjustment at all. However, we can just turn around the equation we used to calculate the ENR and calculate the NF instead:

$$NF_{actual} = NF_{indicated} + ( ENR\ (noise\ source) - ENR_{cal} )$$

There is a short cut. My noise source has an ENR around 12 dB, so I set the "CAL ADJ" in the *CAL* position as if the ENR were exactly 3 dB higher, then subtract 3 dB from the reading. Even easier, the meter has a *+3dB* position on the "ADD TO NOISE FIGURE" switch. Using that position, I can read the meter starting at 0 dB. Any ENR difference from 15 dB that matches one of the meter scales would also work ☺ rather than an involved explanation, I'd urge you to do the noise figure calculations, then try the switch positions and see what works best for quick readout.

Reminder: noise figure meters have a very slow time constant, as long as 10 seconds for some of the older models, to smooth out the random nature of noise. *Tune slowly!*

Don't despair if the ENR of your noise source is much less than 15 dB. The optimum ENR is about 1.5 dB higher than the noise figure being measured.[1] The fact that today's solid state noise sources have an ENR around 5 dB rather than the 15 dB of 20 or 30 years ago shows how much receivers have improved.

# Conclusion

The value of noise figure measurement capability is to help us all to hear better. A good noise source is an essential part of this capability. Accurate calibration is not necessary, but helps us to know whether our receivers are as good as they could be.

**References**


1 R. Pettai, *Noise in Receiving Systems*, Wiley, 1984.

2 M.B. Graves, WRØI, "Computerized Radio Star Calibration Program," *Proceedings of the 27th Conference of the Central States VHF Society*, ARRL, 1993, pp. 19-25.

3 P. Wade, N1BWT, "More on Parabolic Dish Antennas," QEX, December 1995, pp 14-22. *HDL_ANT* program may be downloaded from **www.w1ghz.org/software/hdl_3b4.zip.**

4 W.E. Pastori, "Direct-Reading Measurement of Receiver-Noise Parameters," *Microwave Journal*, April 1973, pp. 17-22.

5 R. Bertelsmeier, DJ9BV, and H. Fischer, DF7VX, "Construction of a Precision Noise Figure Measuring System," *DUBUS Technik 3*, DUBUS, 1992, pp. 106-144.

6 H. Fasching, OE5JFL, "Noise Figure Measurement using Standard Antennas," *DUBUS Technik 4*, DUBUS, 1995, pp. 23-25.

7 C. Suckling, G3WDG, "144MHz wideband noise amplifier," DUBUS, 2/1995, pp.5

8 R. Turrin, W2IMU, "Method for Estimating Receiver Noise Temperature," *Crawford Hill Technical Note #20*, September 1986.

9 K. Britain, WA5VJB, "10 GHz Noise Source," *Microwave Update '87*, ARRL, 1987, p 63.

10 P. Wade, N1BWT, & S. Horsefield, NR1E, "Homebrew Solid State Noise Sources," *Proceedings of the 1992 (18th) Eastern VHF/UHF Conference*, ARRL, 1992.

11 W.E. Sabin, WØIYH, "A Calibrated Noise Source for Amateur Radio," *QST*, May 1994, pp. 37-40.

12 G.E. Valley, Jr., & H. Wallman, *Vacuum Tube Amplifiers*, MIT Radiation Laboratory Series, McGraw-Hill, 1948.

13 P. Wade, N1BWT, "Building VHF Power Attenuators," *QEX*, April 1994, pp. 28-29. *PAD.EXE* program may be downloaded from **www.arrl.org/qexfiles**.

# PC BOARDS

**Bill Kaune W7IEQ**

I originally made a single-sided printed circuit board from scratch for the power meter. I made the artwork using the computer program TurboCad (Version 7). The resulting TurboCad file is named "PC Board, Version 3.tcw" and is included. This artwork was made from the perspective of looking from the top. I normally print the artwork onto a transparency blank (clear plastic sheet, normally used for transparencies, available at office products stores) and use this to mask the raw PC board. I am also enclosing a pdf file of the artwork in case you want to use it. It is named "PC Board, Rev 3, Artwork.pdf". If you do this, be sure the magnification of the printer is properly set. On mine, I had to set Scaling to "None" in the printer dialog of the program used to display the file. Finally, I am including another drawing of the artwork, this time expanded to fill the page, and showing the parts locations in pencil. This file is named "PC Board Layout, Version 3, showing parts.pdf". Note that the dotted lines shown on this view are wires that have to be installed.

Several years after I made this board, I discovered ExpressPCB, an online maker of circuit boards. Doing boards with them is much easier, and you get a nicer product. You can download schematic capture and PC board software from their website (no cost). This software is easier to use than TurboCad, and I now make all my circuit boards using it. When you have completed the design, you simply transmit the resulting file to ExpressPCB and they make boards that get delivered within a few days. The price is modest, and for me worth it. The file named "PC Board, Version 3.pcb" is the ExpressPCB file for the main PC board and the file "RF Board, Version 3.pcb" is the PC Board file for the RF section (U1 and U2). To use these files, you will have to download the appropriate software from the ExpressPCB website. Were I doing this project now, I would get the boards made from ExpressPCB rather than make them myself. Please note that I just designed these two boards and have not tested the design. I tried to be careful, but it is possible that they could contain an error or two.

The ExpressPCB board designs make certain assumptions. The resistors R3, R5, R6, and R7 are assumed to have a lead spacing of 0.25 inches, usually meaning the resistor is rated at 1/10 – 1/8 watt. The 5 kΩ trimmer resistor is assumed to have three leads in a row with spacing between adjacent leads of 0.1 inches. All of the 0.1 μF tantalum capacitors are assumed to have a lead spacing of 0.1 inches. These capacitors are polarized, so you have to be sure to install them with

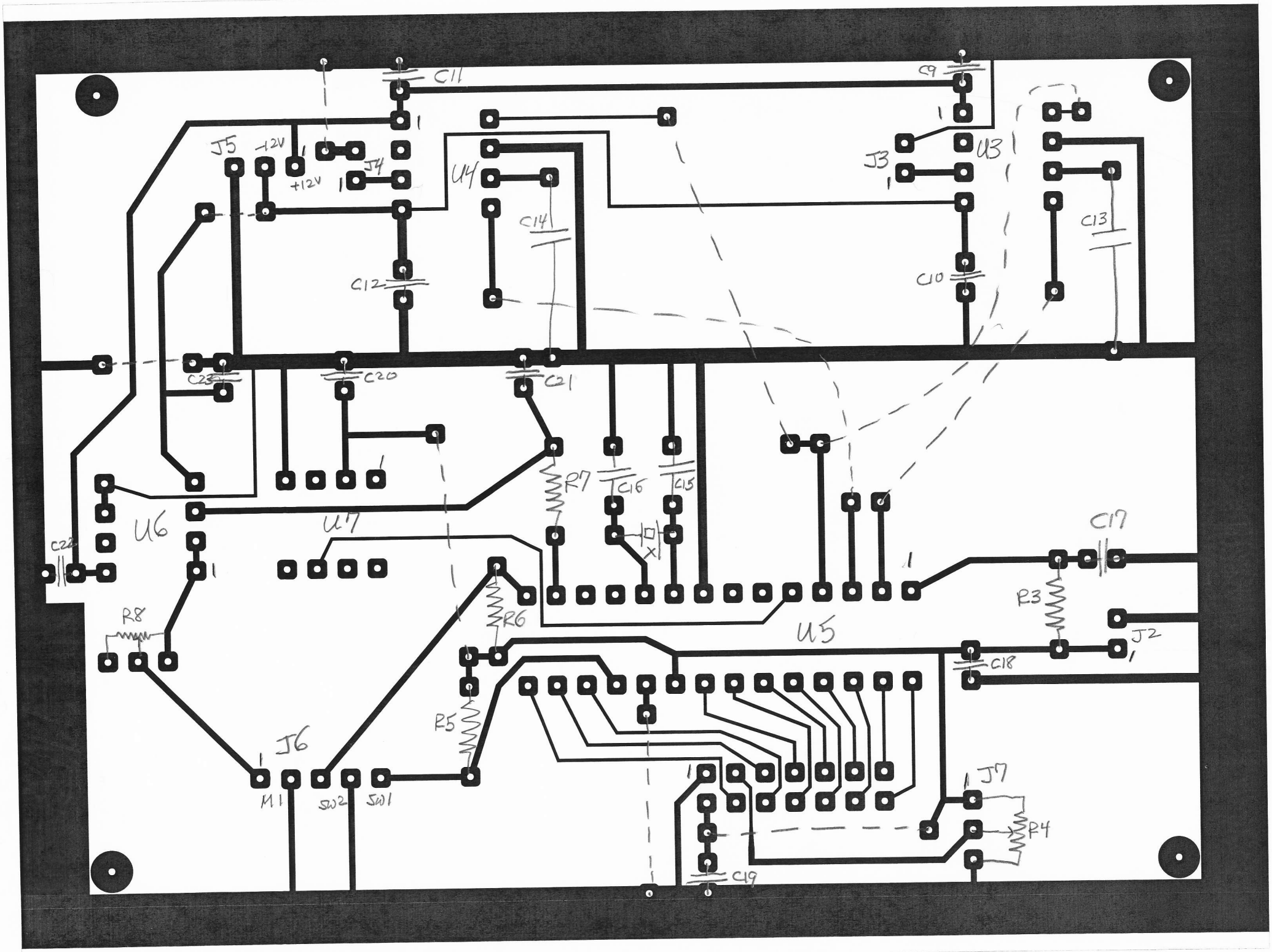
the + lead to the higher voltage. For the –12 volt bus bypass capacitors, the positive side goes to ground. C15 and C16 are assumed to have a lead spacing of 0.2 inches. C17 is assumed to have a 0.1-inch lead spacing.

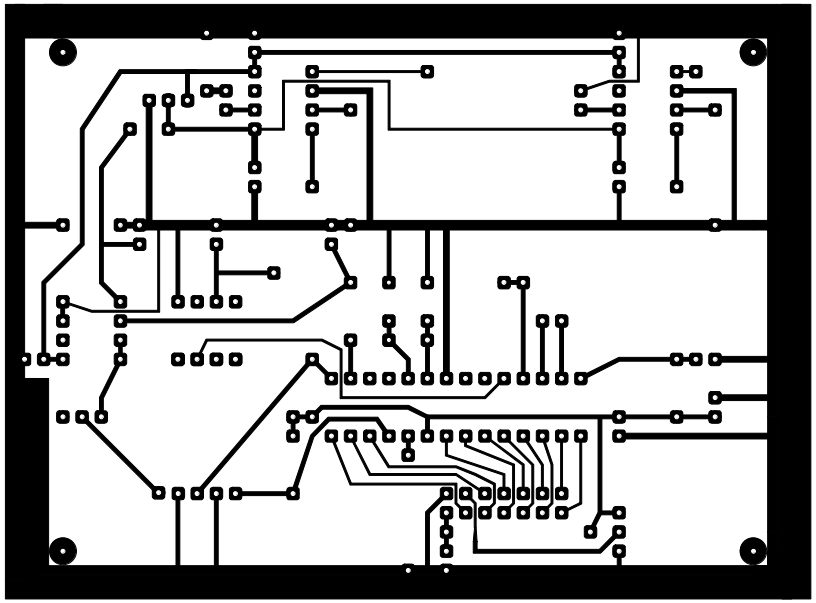
The surface mount parts used in the RF section of the unit have the following sizes: R1 and R2 0603; C1-C4 0603, C5 and C6 3216. U1 and U2 are the surface mount versions of these parts. You could use different size resistors and capacitors, but you would need to change the PC board design appropriately.

The main board includes seven headers (connectors). The pin spacing on all headers is 0.1 inches. J1 is two parallel 7-pin headers and is used to connect to the LCD display. J2 is a 2-pin header. Pin 1 is connected to the +5 volt supply and pin 2 is connected to ground of the power supply. J3 and J4 and both 2-pin headers. J2 is the connection to the forward-power channel on the RF board. Pin 1 carries the signal and pin 2 the ground. J3 serves a similar purpose for the reflected-power channel on the RF board.

J4 is a three pin header. Pins 1 and 2 are connected to the +12 volt and –12 volt supplies, while pin 3 serves as the ground. J6 is a 5-pin header. Pins 1 and 2 connect to the 1-mA analog meter on the front panel; pin 1 is the + lead. Pins 3 and 4 connect to the front panel switch #2 (SW2). This switch is used to select between forward or load power. Pins 4 and 5 connect to the front panel switch #1 (SW1), which selects between the shorter and longer averaging times.

Finally, J7 is a 3-pin header. Leads to this header connect to the front panel potentiometer R4, which is used to set the contrast of the LCD display. Pin 1 connects to the top of R4, pin 2 to its slider, and pin 3 to the bottom (ground) side of R4.


C11
C9
J5
+12V
+12V
J4
U4
J3
U3
C14
C13
C12
C10
C23
C20
C21
R7
C16
C15
U6
U7
C22
C17
R8
R6
U5
R3
J2
C18
R5
J6
J7
M1
SW2
SW1
R4
C19



# A Modern Directional Power/SWR Meter

*Every ham needs an RF power meter.*
*Here's a high performance unit to build at home.*

**Bill Kaune, W7IEQ**

I first got the idea of designing and building a power meter from a construction article in *The 1997 ARRL Handbook* entitled "The Tandem Match — An Accurate Directional Wattmeter" by John Grebenkemper, KI6WX. John described the difficulties of building an accurate power meter using diodes to convert RF to dc because of the diode's inherent nonlinearity. He describes a fairly complicated (at least it looked complicated to me) analog circuit that corrects for this nonlinear behavior and that also calculates and displays SWR.

Shortly after reading this article, I noticed several articles in *QST* that described a new integrated circuit, the Analog Devices AD8307, that converts a low level RF signal into a voltage proportional to the logarithm of the signal's power. I became intrigued with this device because it eliminated the difficulties associated with the use of diodes and would work over a wide range of powers, from milliwatts to the legal limit.

In addition, I was interested in learning more about microprocessors. Thus, I developed the idea of using AD8307s in the front end of a power meter to convert from RF watts to dBm, then a microprocessor with a built-in analog-to-digital converter (ADC) to process the dBm signals and display power and SWR on a liquid crystal display panel and, perhaps, a panel meter.

The primary use I had in mind for this unit was to monitor the output power and tuning of my rig. I live in an area with antenna restrictions so must operate with a wire antenna whose configuration is more determined by my lot size and configuration than by electrical considerations. Thus, my antenna does not present a good match to 50 Ω on any amateur band. As a consequence, I always use a manual antenna tuner. My setup, including the power meter described here, is shown in Figure 1.

RF power generated by my transmitter is routed via RG-8 coaxial cable through a directional coupler to my antenna tuner, which is connected to my antenna with about 60 additional feet of RG-8. The directional coupler contains circuits that sample the RF power flowing from the transmitter to the tuner (the *forward power*) and the RF power reflected back from the tuner to the transmitter (the *reflected power*). These samples are sent via RG-58 cable to the two input channels of the power meter. This article describes the directional coupler and the power meter in enough detail so that an interested ham can duplicate the device or, perhaps, improve on my design.

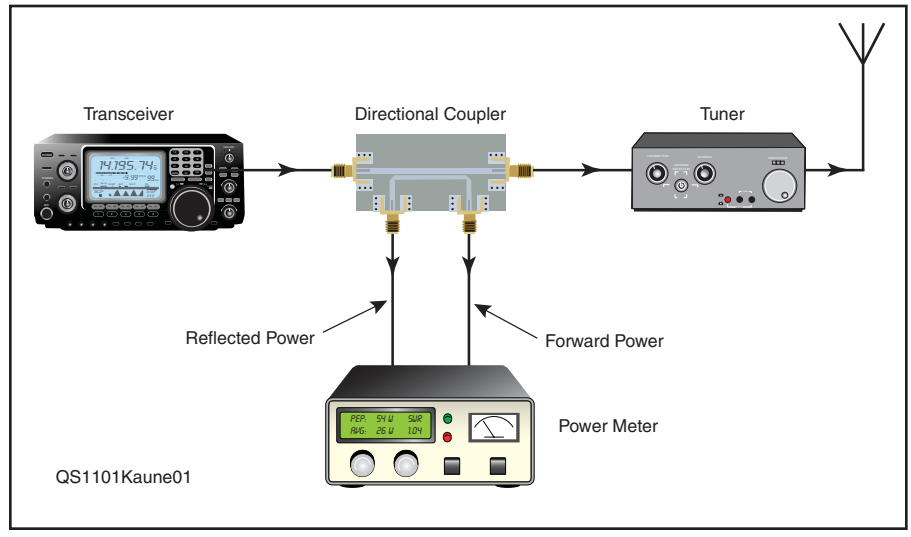



**Figure 1 — W7IEQ station setup, including the power meter being described here.**

## Directional Coupler

The directional coupler I built is based on the unit described in the article by John Grebenkemper mentioned earlier. Coaxial (SO-239) UHF sockets were mounted on opposite faces of an aluminum box (Radio Shack 270-238). A cable from the transceiver was connected to one of these sockets and the other was connected to my antenna tuner. Two BNC sockets were mounted 2 inches away from these coax sockets. Two short lengths of RG-8 coaxial cable were prepared by exposing about ¼ inch of the shield on one end of each. I wound 31 turns of #26 AWG magnet wire on each of two Amidon FT-82-67 ferrite toroids so that the windings occupied about 75% of the circumferences of the cores. These toroids were then slipped over the two sections of RG-8 until they were about ½ inch from the end of the exposed shield. One section of RG-8 was soldered between the two UHF sockets and the other between the two BNC sockets. Holes were drilled in the PC board shields to accommodate these two lengths of cable.

The exposed braid at one end of each section of RG-8 was soldered to a nearby solder lug, as shown in Figure 2. (Do not solder the other end of the braid to ground.) This braid forms a shield that prevents capacitive coupling between the transmission line and the wire wound on the toroids. In addition, one lead from each of the toroids was soldered to these lugs. The other lead from each toroid was passed through the center shield and soldered to the center terminal of the nearby connector. Small rubber grommets were used to insulate the points that these leads passed through the center shields.

The forward and reflected power samples coupled from the main line are reduced by a factor of $1/N^2$, where N = 31 is the number of turns of wire on each toroid. Thus the forward and reflected power samples are reduced by about 30 dB. For example, if a transceiver were delivering a power of 100 W to a pure 50 Ω load, the forward power sample from

From January 2011 QST © ARRL

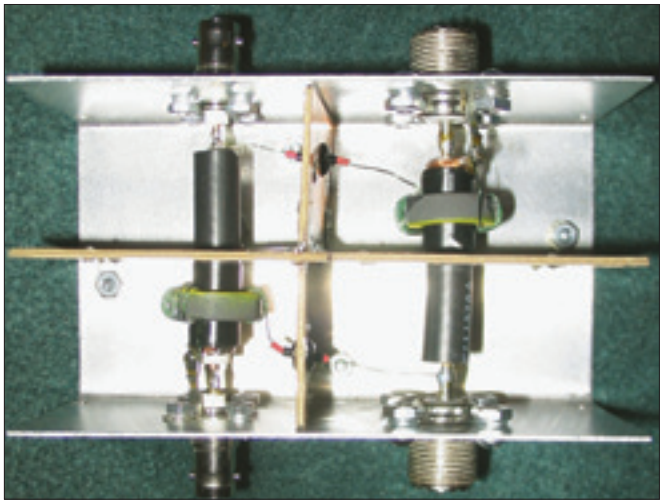


Figure 2 — Compled directional coupler.




**Figure 3 — Front panel of power meter. The LCD shows the peak envelope power (PEP), the average envelope power (AEP) and the SWR. The two knobs control the contrast and back lighting of the LCD. One toggle switch determines whether forward or load powers are displayed. A second switch sets the averaging time for the AEP calculation. The meter shows the SWR and is used for tuning purposes.**

the directional coupler would be about 0.1 W (20 dBm) and the reflected power sample, in theory at least, would be exactly 0 because there would be no reflections from the 50 Ω load. Of course, in a real device, the reflected power sample will never be exactly 0. The *directivity* of a directional coupler is defined as the ratio of the forward power sample divided by the reflected power sample when the coupler is terminated in 50 Ω. In my coupler, the directivity, measured using an inexpensive network analyzer, is at least 35 dB at 3.5 MHz and 28 dB at 30 MHz.

## Power/SWR Meter — Circuit Description

Figure 3 shows a front panel view of the power meter. I used a two line, 20 characters per line, liquid crystal display (LCD) to display the measured peak (PEP) and average (AEP) envelope powers as well as the standing wave ratio (SWR). Depending on the position of a front panel switch, the power meter calculates either the peak and average envelope power traveling from the transceiver to load (the forward power) or the peak and average envelope powers actually delivered to the load (the forward power minus reflected power). The average envelope power (AEP) represents an average of the forward or load powers over an averaging period, depending on the position of a second front-panel switch, of either 1.6 or 4.8 seconds. I generally use the shorter period, but sometimes use the longer when I am operating SSB.

I included a 1 mA-movement analog meter on the front panel to facilitate antenna tuning. This meter continuously displays the quantity 1 – 1/SWR, where SWR is the standing wave ratio on the line. Thus, an SWR of 1.0 corresponds to a meter reading of 0, that is no deflection of the meter. An SWR of 2 results in a 50% deflection of the meter, while an SWR of 5 produces an 80% deflection of the meter. I have found this method of displaying SWR to be effective for antenna tuning.

Figure 4 shows the schematic diagram and parts list of the power meter. The forward and reflected power samples from the directional coupler enter at the left through two identical channels. Since the logarithmic detectors, U1 and U2 in Figure 4, can only handle a maximum input power of about 15 dBm, I placed two external 20 dB attenuators (Mini-Circuits HAT-20) in cables from the directional coupler. As noted earlier, the directional coupler has an internal attenuation of about 30 dB, so the total attenuation in each channel is about 50 dB. Thus, a rig operating at a power level of 1 kW (60 dBm) will result in an input to the forward power channel of about 10 dBm.

The internal input impedances of the logarithmic detector chips, U1 and U2, are about 1100 Ω in parallel with 1.4 pF. These impedances are in parallel with R1 and R2, respectively, both 52.3 Ω, yielding net input impedances of 49.9 Ω. Capacitors C7 and C8 set the time constants of the output networks of U1 and U2 to filter out RF components in the signal while retaining the modulation components with frequencies up to about 20 kHz. In other words, the outputs of U1 and U2 follow the modulation envelope of the RF signal.

The output signals from U1 and U2 enter U3 and U4, sample-and-hold (S/H) integrated circuits. These circuits operate as follows: As long as the voltages on pins 8 are held high (+5.0 V), the outputs on pins 5 are the same as the inputs (pins 3). However, if the voltages on pins 8 are set low (≈ 0 V), the outputs become *frozen* at the values at the inputs at the moment when the voltages on pins 8 were changed. In this way, input voltages from the forward and reflected power logarithmic detectors can be sampled at the exact same time and held for subsequent serial reading into the microprocessor.

I used a PIC microprocessor in the power meter for several reasons. PIC processors seem to be used in a many types of amateur equipment, so it seemed sensible to become familiar with them. They are remarkably inexpensive. Being RISC (reduced instruction set computers), they have a relatively small number of instructions to learn. Also, you can purchase units that contain a variety of on-chip peripherals, such as analog-to-digital converters (ADC). And, finally, there are software and a large number of helpful documents available for no charge from the producer of these devices (**www.microchip.com**).

Because I needed a device with at least two channels of ADC and a way of outputting a voltage that would drive an analog panel meter, I selected the PIC16F876A, which includes five input analog channels and a pulsewidth modulated output. The two analog signals from the S/H chips were input on pins 2 and 3 of U5, the system processor. In this processor, the ADC is a 10 bit device and conversion can be done relative to a reference voltage. While you can use the 5.0 V supply voltage for this purpose, I elected to improve the accuracy and stability of the unit by using a separate voltage reference chip (U7) that delivers a precise 2.50 V from its pin 6. This choice was also appropriate because the maximum output voltage from the logarithmic converters is close to 2.5 V.

I operated the processor with a 20 MHz clock, which corresponds internally to an instruction execution time of 200 ns (four clock cycles per one instruction cycle). As noted earlier I selected a two line, 20 character per line, LCD to display digital values of the peak envelope power (PEP), average envelope power (AEP), and SWR. R4 adjusts the contrast of the display and R8 its

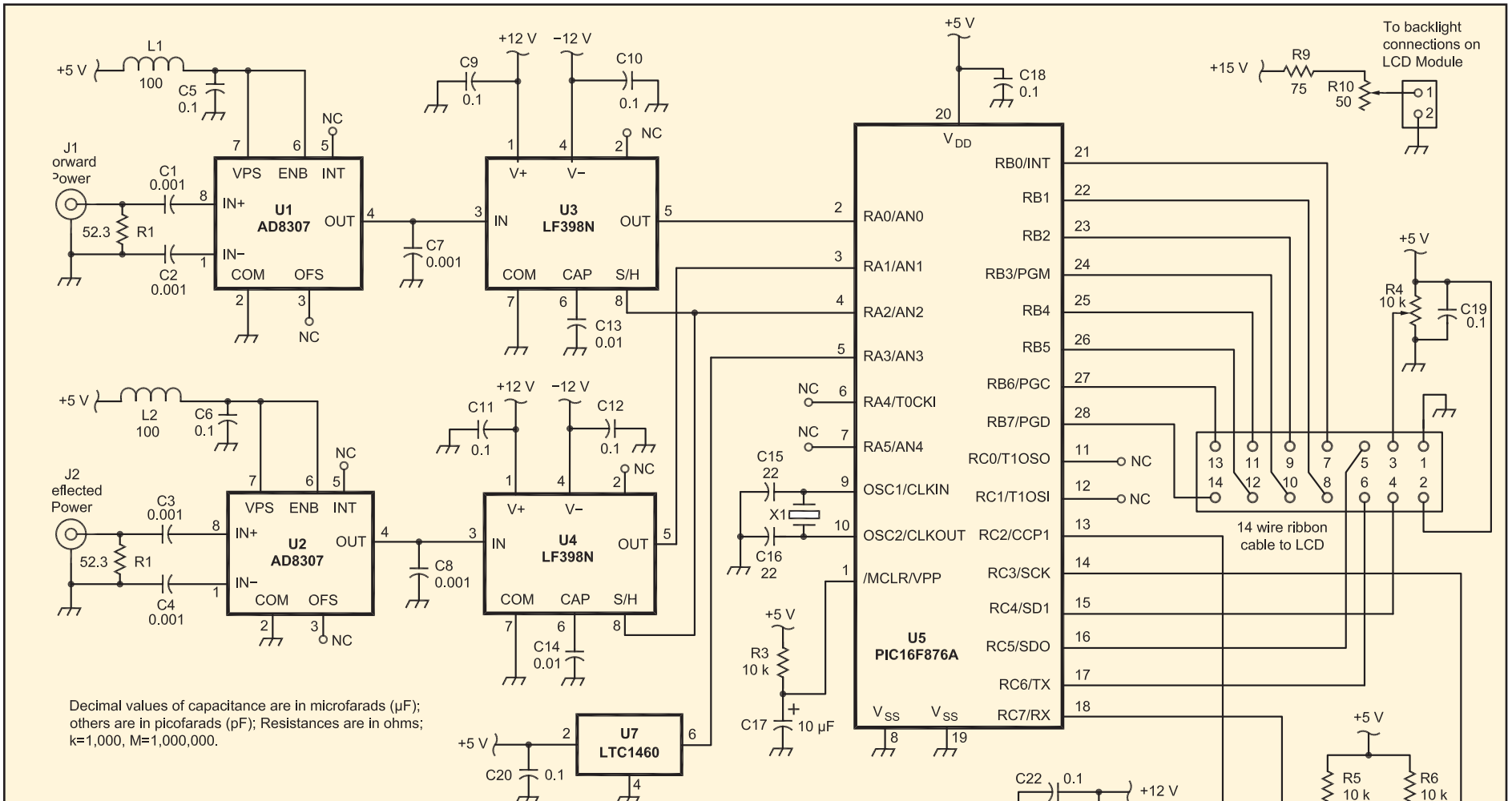



**Figure 4 — Schematic diagram and parts list for power meter.**

**C1-C4, C7, C8 — 0.001 µF ceramic capacitor.**
**C5, C6, C9-C12, C18, C19, C20-C23 — 0.1 µF tantalum capacitor.**
**C13, C14 — 0.01 µF polypropylene capacitor.**
**C15, C16 — 22 pF silver mica capacitor.**
**C17 — 10 µF, 15 V electrolytic capacitor.**
**J1, J2 — Female panel mount BNC connectors.**
**L1, L2 — 100 µH RF choke.**
**LCD — 2 line × 20 character display (Digi-Key 73-1063-ND).**
**M1 — 1 mA meter.**
**R1, R2 — 52.3 Ω, ¼ W resistor.**
**R3, R5, R6 — 10 kΩ, ¼ W resistor.**
**R4 — 10 kΩ potentiometer.**
**R7 — 100 kΩ, ¼ W resistor.**
**R8 — 5 kΩ trimmer.**
**R9 — 75 Ω, 4 W resistor.**
**R10 — 50 Ω, 2 W potentiometer.**
**S1, S2 — SPST miniature switch.**
**U1, U2 — AD8307 integrated circuit.**
**U3, U4 — LF398N integrated circuit.**
**U5 — PIC16F876A programmable integrated circuit.**
**U6 — TL082 operational amplifier integrated circuit.**
**U7 — LTC1460, 2.5 V reference.**
**X1 — 20 MHz microprocessor crystal.**

From January 2011 QST © ARRL

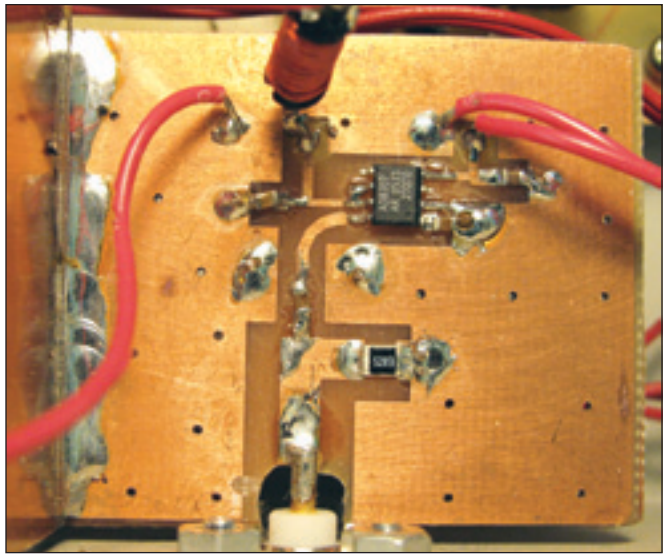


**Figure 5 — One of the two RF logarithmic converters.**

brightness. In practice, I never change these settings, so if I were building this unit again, I would place them on the back panel.

The PIC processor includes a pulsewidth-modulated output that I use to control the analog SWR meter on the front panel. Essentially, this peripheral produces an output train of rectangular pulses with a frequency of about 4.9 kHz. The pulse height is 5.0 V, the operating voltage of the processor. The spacing in time between two consecutive pulses is about 200 µs. In software, the width of the pulses can be adjusted in 256 increments from 0 µs (the output voltage is always 0) to 100 µs (output is 50% high and 50% low) to 200 µs (output is always high). R7 and C21 comprise a low-pass filter that removes the 4.9 kHz component and leaves only the dc component of this pulse train. In other words, this filter essentially "calculates" the average voltage from the pulsewidth modulated stream, which ranges from 0 V when the pulsewidth is 0 µs to 5.0 V when it is 200 µs.

## Construction Notes

The power meter was built in a 8 × 3 × 6 inch deep aluminum cabinet purchased from our local RadioShack; unfortunately this cabinet is no longer available. The RF circuits associated with U1 and U2 were constructed on a small piece of PC board. Figure 5 is a close-up photograph of one of these circuits. I actually built two versions. The first used the *dead bug* construction method and included about 20 dB of on-board attenuation. Unfortunately, the result was somewhat frequency sensitive. The second version used surface mount parts and did not include any internal attenuation.

Eight data lines and three control lines connect between the microprocessor and the LCD. These connections, as well as the connections associated with power, ground and the contrast circuits for the display, were accomplished using a 16 wire ribbon cable. This cable can be seen in Figure 6.

The following components were mounted on the front panel: The brightness and contrast potentiometers (R9 and R4, respectively), the FORWARD/LOAD POWER and AVERAGING TIME switches (S1 and S2, respectively), the LCD display and the SWR analog meter (M1). Connections between these components and the main PC board were made using Molex connectors. I purchased a number of 12 pin male and female headers (Digi-Key WM4010-ND and WM2021-ND) and associated crimp terminals (WM2200-ND). I then cut the headers into smaller pieces (fewer pins) as needed, and soldered the terminals onto the wires in each cable harness. As Figure 6 shows, I used these connectors liberally so that the PC board could be removed without having to unsolder any connections. Finally, I mounted a male DB-9 connector on the rear panel for connection to the remote power supply.

## Power Supply

The power supply needs to supply the following voltages: an unregulated 15 V for the back-lighting of the LCD display; a regulated ±12 V for the sample and hold chips (U3 and U4) and U6, the meter driver; and a regulated +5.0 V for the RF deck, system microprocessor and the voltage reference chip (U7). Since the power requirements are small, I constructed a simple power supply using only a single transformer. The schematic and parts list are given in Figure 7. I built the power supply in a separate aluminum box (RadioShack 170-238) and all the parts for it were obtained from Radio Shack. I used a 9 pin serial cable from my junk box to carry power from this supply to the power meter.

## Software

As in many modern pieces of electronic equipment, a microprocessor plays a key role in my power meter. As noted earlier, I selected the PIC16F876A for my power meter. While I found the PIC relatively easy to program, learning how to use the various peripherals was more of a challenge. The data sheets from the manufacturer were difficult to follow, and I only began to make progress after purchasing a third-party book describing these units.[1]

When power is supplied to the power meter, the microprocessor will not start until the voltage at pin 1 rises to a threshold value. R3 and C17 establish a time constant of 0.1 seconds, enforcing a delay of about ¼ second. This allows the supply voltages time to stabilize before the processor starts. Once running, the processor samples the forward and reflected power signals every 90 µs, that is, at a rate of 11.1 kHz. Between each sample, the processor converts the power signals, in dBm, to watts. Each measured dBm value is a 10 bit number. This number is used as an index into lookup tables held in program memory. After lookup, each forward and reflected power becomes a 16 bit floating-point number in units of watts. Each forward power value is added to an accumulator, later used to calculate the average envelope power (AEP). In addition, the processor keeps track of the maximum forward power and the respective reflected power.

After 1111 samples, taken during a period of 0.10 seconds, the processor pauses to calculate the SWR, using the maximum forward and corresponding reflected powers measured during this period, updates the front panel tuning indicator, and updates the LCD display if the maximum forward power measured during the period exceeds the PEP value being displayed at that moment. Depending on the setting of the front-panel averaging switch, either 16 or 48 cycles of 1111 samples each are taken, at which point a new value of the AEP and SWR are calculated and displayed on the LCD.

The lookup tables for the forward and reflected powers occupy 4098 bytes (50%) of the available program memory. The remainder of the program uses 2753 bytes, of which 885 bytes are consumed by a 24

[1]M. Predko, *Programming and Customizing PIC Microcontrollers*. McGraw-Hill, New York (2001). Available from Amazon (**www.amazon.com**) or Barnes & Noble (**www.barnesandnoble.com**).

**From January 2011 QST © ARRL**

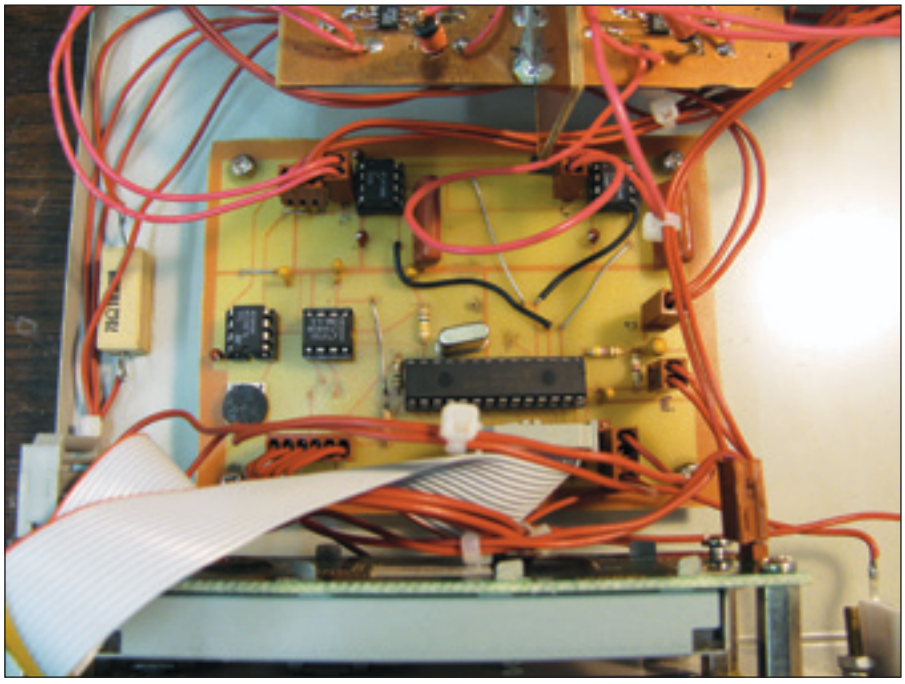


**Figure 6 — Interior of power meter. The front panel of the unit is below the bottom of the photograph. The edge of the RF board shows at the top of the photograph. The larger integrated circuit in the middle of the photograph is the 16F876A processor. The two sample-and-hold integrated circuits are located above the processor. The two chips to the left of the processor are a 2.50 V precision voltage reference and a TL082 operational amplifier.**

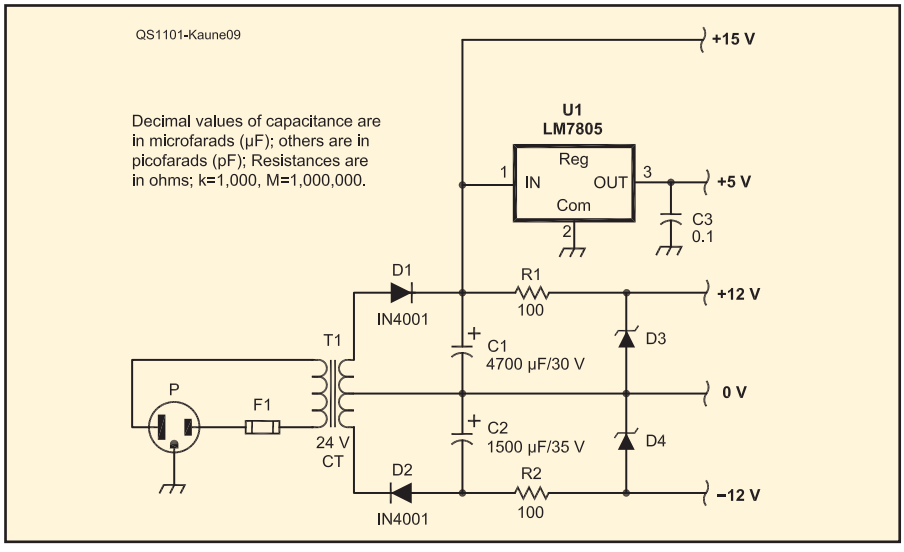



**Figure 7 — Schematic diagram and parts list of the power supply.**
**C1 — 4700 µF, 35 V electrolytic capacitor.**
**C2 — 1000 µF, 35 V electrolytic capacitor.**
**D1, D2 — 1N4001 diode.**
**D3, D4 — 1N4742A 12 V Zener diode.**
**F1 — Fuse, 1A.**
**R1, R2 — 100 Ω, 1 W resistor.**
**T1 — Power transformer, 24 V CT.**
**U1 — LM7805 voltage regulator.**

bit floating point package that I downloaded from the Microchip Web site.

## Calibration

The output of a logarithmic converter is a voltage linearly proportional to the input RF power measured in units of dBm. Furthermore, the digitized value, $D_{ADC}$, of the voltage measured by the ADC is a linear function of the input voltage. Putting these together, dBm = $\alpha D_{ADC} + \beta$, where $\alpha$ and $\beta$ are constant. Calibration consists of determining these two constants. I did this by writing a special calibration program for the PIC processor that constantly displayed the digitized ADC values for both the forward and reflected power channels as I inputted various power levels to each channel. In this way, I determined that $\alpha = 0.10063$ and $\beta = -70.83$ for the forward power channel and $\alpha = 0.09965$ and $\beta = -71.50$ for the reflected power channel. Unfortunately, I have no way to accurately measure absolute power, so these values are somewhat uncertain by perhaps ±10% or so.

Once I had determined the calibrations of the two channels, I constructed the look-up tables used by the processor to quickly determine the power in watts corresponding to a measured ADC number.

## Conclusions

In contrast to a number of my projects, the power meter described here seems to work well and has proved to be a useful instrument. I use it whenever I operate. My transceiver has a maximum output of a little more than 200 W, and I have a linear amplifier that can run an output power of 1000 W. The power meter works well with both levels of power. What I did not anticipate when I built it is how well it works at low power levels. When I transmit SSB, but am not speaking into the microphone, the background noise in my shack is sufficient to produce an output power of about 0.1 W or so. I find that this power level is sufficient for me to adjust my antenna tuner using the tuning meter on the front panel of the power meter. In this way, I can tune up without disturbing other stations operating on nearby frequencies.

A version of this article with additional construction details, a discussion of how the coupler works and related Hamspeak items is provided on the QST-In-Depth Web site (**www.arrl.org/qst-in-depth**). Also there are the required firmware and PC board artwork.

*Bill Kaune, W7IEQ, was first licensed in 1956 as a Novice and was licensed again in 1998. He upgraded to Amateur Extra class in 2000. Bill earned BS and PhD degrees in physics. He spent most of his career collaborating with biologists and epidemiologists researching the biological effects of power-frequency electric and magnetic fields. He is now retired.*

*Bill is married, has two grown daughters, four grandchildren and a standard poodle. In addition to Amateur Radio, he spends his time hiking and backpacking, including two recent trips to Alaska. Bill is currently the treasurer of the Jefferson County Amateur Radio Club, a member of the ARRL and a member of the ARRL's RF Safety Committee. You can reach him at 160 Cedarview Dr, Port Townsend, WA 98368 or at* **w7ieq@arrl.net**.

VOTE Did you enjoy this article? Cast your vote at:
www.arrl.org/cover-plaque-poll

**MORE ON A HIGH POWER RF SAMPLER (MAY 2011)**

*QST* Technical Correspondence, July 2011

ARRL Lab Engineer Zach Lau, W1VT, pointed out an error in the May 2011 *QST* Technical Correspondence letter from Tom Thompson, WØIVJ. On page 53, in the text describing Figure 4, we incorrectly listed the wall thickness of the hobby brass tubing that Tom used to build his sampler as 0.14 inch. Of course that specification should have been 0.014 inch! I apologize for any confusion and inconvenience this may have caused our readers.

Zack and others also raised some questions about the resistor type Tom used, and also wondered how he formed the concave half-round end on the tubing to be soldered to the "top" of the T. Tom answered those questions and provided some additional construction information for those interested in duplicating his sampler. — *73, Larry Wolfgang, WR1B, ARRL HQ;* **wr1b@arrl.org**.

Here are some further comments about my RF Sampler Technical Correspondence:

1) The tubing is actually 9/16 inch OD with a 0.014 inch wall thickness.

2) The 15 Ω, 2 W resistor I used is a metal oxide resistor that I obtained locally. It measures 15 Ω with 20 nH of series inductance. Digikey carries a resistor that should work: part no. P15W-2BK-ND. I don't have this resistor but I have some 100 Ω resistors from the same manufacturer that measure 100 Ω with 4 nH of series inductance. The lower inductance should improve the VHF performance of the sampler.

3) The 34.8 Ω, ¼ W resistor is a 1% metal film component, Digikey part no. CMF34.8QFCT-ND.

**Construction Steps**

Construct the BNC/toroid assembly as shown at the top of Figure 4, on page 53 of May 2011 *QST*. Cut the tube to length so that the flanges on the BNC connectors just fit inside the through tube, and drill a ¼ inch hole in the center of the tube as shown in the Figure 4 photo. Insert the toroid assembly into the through tube and snake the toroid wires through the ¼ inch hole.

Solder the BNC flanges to the tube, filling the slight gap between the tube and the flanges with solder. Solder a ¼ inch wide brass strip to the through tube just below the ¼ inch hole. Cut the strip so it is just slightly longer than the 15 Ω, 2 W resistor. Cut the lead on one end of the 15 Ω resistor to about 1/8 inch and fold the other end 180° so that it is parallel with the resistor body and solder it to the far end of the brass strip.

Next, solder the 34.8 Ω resistor, along with one of the toroid wires, to the 1/8 inch stub of the 15 Ω resistor. Solder the other toroid wire to the near end of the brass strip. Connect the other end of the 34.8 Ω resistor to the BNC connector center pin.

File a notch into the T tube with a small round file, so that the curvature matches the curvature of the through tube, slip it over the resistor assembly and cut it to length so the BNC flange just reaches the end of the tubing.

Solder the tube to the BNC in the same manner as was done with the cross piece, and solder the curved part of the T tube to the through tube, filling in the gaps with solder. — 73,

Tom Thompson, WØIVJ, 990 Toedtli Drive, Boulder, CO 80305; **tlthompson@qwest.net**.

# ARRL Periodicals Archive – Search Results

## A membership benefit of ARRL and the ARRL Technical Information Service

**ARRL Members:** You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

**Need a higher quality reprint or scan?** Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

**QST on CD-ROM**: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

**Non-Members:** Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.

### Copyright/Reprint Notice

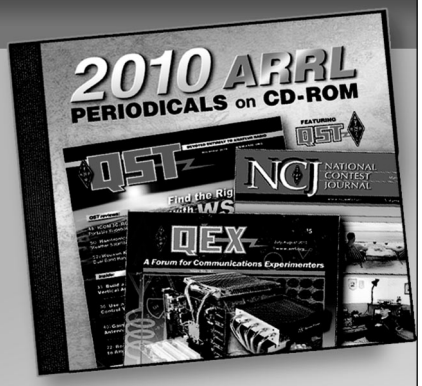
In general, all ARRL content is copyrighted. ARRL articles, pages, or documents--printed and online--are not in the public domain. Therefore, they may not be freely distributed or copied. Additionally, no part of this document may be copied, sold to third parties, or otherwise commercially exploited without the explicit prior written consent of ARRL. You cannot post this document to a Web site or otherwise distribute it to others through any electronic medium.

For permission to quote or reprint material from ARRL, send a request including the issue date, a description of the material requested, and a description of where you intend to use the reprinted material to the ARRL Editorial & Production Department: permission@arrl.org.

**QST Issue:** Jun 1995
**Title:** An RF Step Attenuator
**Author:** Denton Bramwell, K7OWJ

Click Here to Report a Problem with this File

ADVERTISEMENT




**2010 ARRL Periodicals**
on CD-ROM

**ARRL's popular journals are available on a compact, fully-searchable CD-ROM. Every word and photo published throughout 2010 is included!**

- ***QST*** **The official membership journal of ARRL**
- ***NCJ*** **National Contest Journal**
- ***QEX*** **Forum for Communications Experimenters**

**SEARCH** the full text of every article by entering titles, call signs, names—almost any word. **SEE** every word, photo (including color images), drawing and table in technical and general-interest features, columns and product reviews, plus all advertisements. **PRINT** what you see, or copy it into other applications.

**System Requirements**: Microsoft Windows™ and Macintosh systems, using the industry standard Adobe® Acrobat® Reader® software. The Acrobat Reader is a free download at **www.adobe.com**.

**2010 ARRL Periodicals on CD-ROM**

ARRL Order No. 2001
**Only $24.95***
*plus shipping and handling

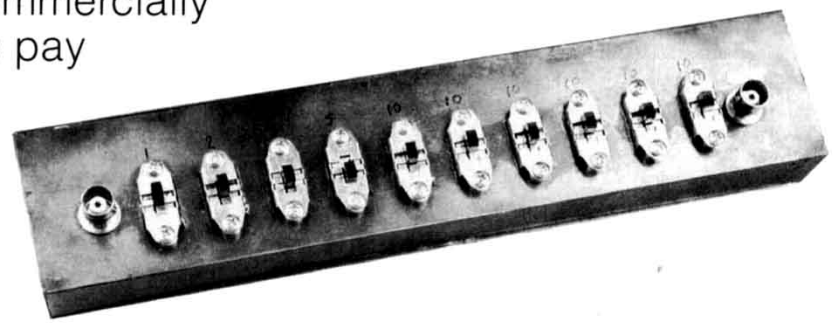
Additional sets available:
2009 Ed., ARRL Order No. 1486, $24.95
2008 Ed., ARRL Order No. 9406, $24.95
2007 Ed., ARRL Order No. 1204, $19.95
2006 Ed., ARRL Order No. 9841, $19.95
2005 Ed., ARRL Order No. 9574, $19.95
2004 Ed., ARRL Order No. 9396, $19.95
2003 Ed., ARRL Order No. 9124, $19.95
2002 Ed., ARRL Order No. 8802, $19.95
2001 Ed., ARRL Order No. 8632, $19.95

ARRL The national association for AMATEUR RADIO™
SHOP DIRECT *or call for a dealer near you.*
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)

Jun 1995 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

# An RF Step Attenuator

When you can find a good, commercially made RF attenuator, expect to pay the price. Here's where doing it yourself can provide you with what you need—inexpensively.



By Denton Bramwell, K7OWJ
3139 Royalton Hts Rd
St Joseph, MI 49085

If you're doing any serious receiver home brewing, a good RF step attenuator is one of the key pieces of equipment that belongs on your workbench. Good attenuators are hard to find at prices that won't injure your wallet. If you're lucky, you may find a used HP-355 series unit at a flea market, but it won't be cheap! Hams who have them—and know their value—don't surrender them easily!

Hams are accustomed to substituting ingenuity for money. That was my approach. The result: a home brew step attenuator that provides good performance, yet can be built with a few basic tools.[1] This unit is designed for use in 50-Ω line, provides a total attenuation of 71 dB in 1-dB steps, offers respectable accuracy and insertion loss through 225 MHz and can be used at 450 MHz (see Table 1).

**Table 1**
**Step Attenuator Performance at 148, 225 and 450 MHz**
**Measurements made in the ARRL Laboratory**

| *Attenuator set for maximum attenuation (71 dB)* | | *Attenuator set for minimum attenuation (0 dB)* | |
|---|---|---|---|
| *Frequency (MHz)* | *Attenuation (dB)* | *Frequency (MHz)* | *Attenuation (dB)* |
| 148 | 72.33 | 148 | 0.4 |
| 225 | 73.17 | 225 | 0.4 |
| 450 | 75.83 | 450 | 0.84 |

Note: Laboratory-specified measurement tolerance of ±1 dB.

### Description

Refer to Figure 1. The attenuator consists simply of 10 resistive π attenuator sections such as the one shown here. Each section consists of a DPDT slide switch and three 1/4-W, 1%-tolerance metal-film resistors whose values are chosen to obtain the desired amount of attenuation. The completed unit contains single 1, 2, 3 and 5-dB sections, and six 10-dB sections. Table 2 lists the resistor values required for each section.

The enclosure is made of brass sheet stock, readily available at many hardware stores. By selecting the right stock, you can avoid having to bend the metal and perform a minimal amount of cutting.

### Construction

I built the enclosure using only a nibbling tool, a drill press, metal shears and a butane-heated soldering iron. To cut the rectangular tubing to length, I used my drill press equipped with a small abrasive cutoff wheel. It's not an elegant solution, but it's inexpensive and works well enough.

Brass is easy to work and solder. For the enclosure, you'll need two precut 2 × 12 × 0.025-inch sheets and two 1 × 12 × 0.025-inch sheets. The 2-inch-wide stock is used for the front and back panels; the 1-inch-wide stock is used for the ends and sides. For the innards, you need a piece of 5/32 × 5/16-inch rectangular tubing, a 1/4 × 0.032-inch strip, and a few small pieces of 0.005-inch-thick stock to provide interstage shields and form the 50-Ω transmission lines that run from the BNC connectors to the switches at the ends of the step attenuator.

For the front panel, nibble or shear a piece of 2-inch-wide brass to a length of about 9 1/2 inches. Space the switches from each other so that a piece of the rectangular brass tubing lies flat and snugly between them. Drill holes for the #4-40 mounting screws and nibble or punch rectangular holes for the bodies of the slide switches.

Before mounting any parts, solder in place one of the 1-inch-wide chassis side pieces to make the assembly more rigid. Solder the side piece to the edge of the top plate that faces the "through" side of the switches; this makes later assembly easier (see Figure 2).

I installed BNC input and output connectors on the top (front) panel, which may not be the best location. You may be able to achieve a little better lead dress and high-frequency performance if you mount the connectors at the ends of the enclosure.

The DPDT slide switches I used are designed for sub-panel mounting. Their mounting holes are tapped for a #4-40 screw, so I slightly enlarged the holes to allow a #4-40 screw body to slide through. Before mounting the switches, make the "through" switch connection (see Figure 1) by bending the two lugs at one end of each switch to-

**Table 2**
**Closest 1%-Tolerance Resistor Values**

| *Attenuation (dB)* | *R1 (Ω)* | *R2 (Ω)* |
|---|---|---|
| 1.00 | 866.00 | 5.60 |
| 2.00 | 436.00 | 11.50 |
| 3.00 | 294.00 | 17.40 |
| 5.00 | 178.00 | 30.10 |
| 10.00 | 95.30 | 71.50 |

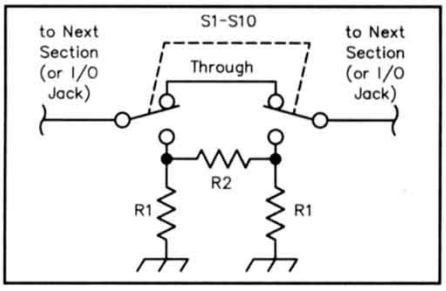



Figure 1—Schematic of one section of the attenuator. All resistors are 1/4-W, 1%-tolerance metal-film units. See Table 2. Any attenuation value from 1 to 71 dB, in 1-dB steps, can be selected by using the appropriate combination of switches. Good accuracy is maintained through 225 MHz and even 450 MHz (see Table 1).

S1-S10—DPDT slide switch (Digi-Key SW116-ND, Digi-Key Corp, 701 Brooks Ave S, PO Box 677, Thief River Falls, MN 56701-0677, tel 800-344-4539, 218-681-6674, fax 218-681-3880); equivalent switches can be substituted.

Misc: Brass sheet and rectangular tubing (see text), metal spacers, hardware and input/output connectors.

[1]Notes appear on page 34.

Jun 1995 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

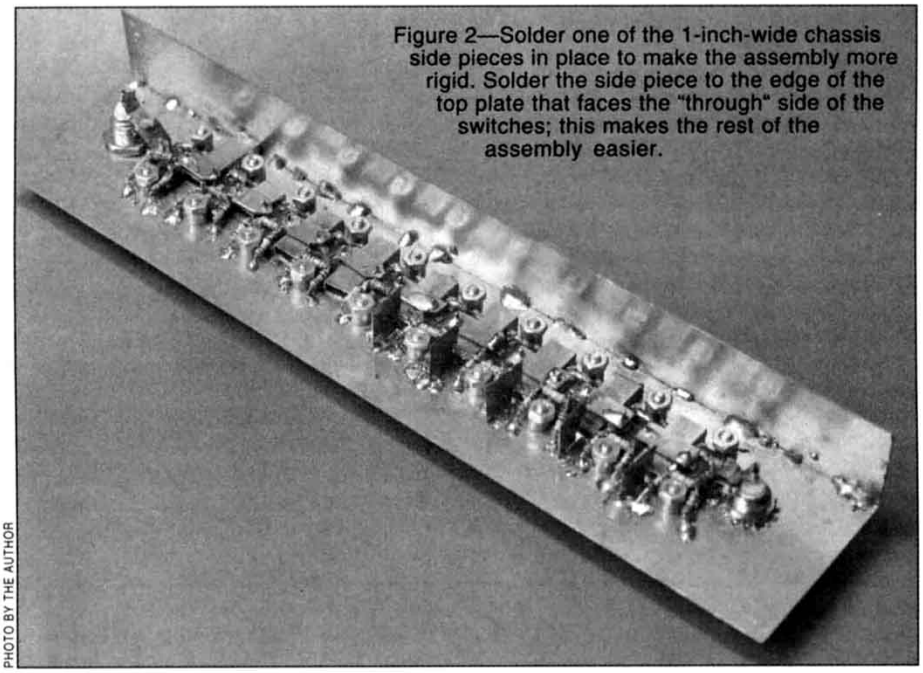


Figure 2—Solder one of the 1-inch-wide chassis side pieces in place to make the assembly more rigid. Solder the side piece to the edge of the top plate that faces the "through" side of the switches; this makes the rest of the assembly easier.

PHOTO BY THE AUTHOR

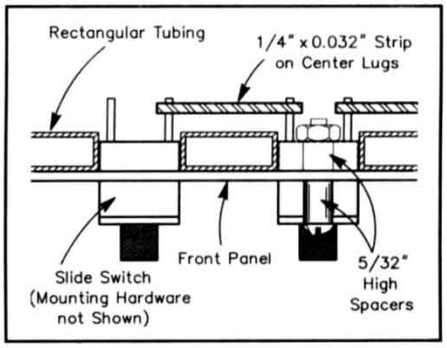



Figure 3— The trick to getting acceptable insertion loss in the "through" position is to make the whole device look as much as possible like 50-Ω coax. The rectangular tubing and the 1/4 × 0.032-inch brass strip between the switch sections form a 50-Ω stripline.

ward each other and soldering the lugs together. Or, solder a small strip of brass between the lugs and clip off the lug ends. Mount the switches above the front panel, using 5/32-inch-high by 7/32-inch-OD spacers. Use the same size spacer on the inside. On the inside, the spacer creates a small post that helps reduce capacitive coupling from one side of the attenuator to the other. The spacers position the switch so that the 50-Ω stripline can be formed later.

The trick to getting acceptable insertion loss in the "through" position is to make the attenuator look as much as possible like 50-Ω coax. That's where the rectangular tubing and the 1/4 × 0.032-inch brass strip come into the picture (see Figure 3); they form a 50-Ω stripline.

Cut pieces of the rectangular tubing, about 3/4-inch long, and sweat solder them to the front panel between each of the slide switches. Next, cut lengths of the 1/4-inch strip long enough to conveniently reach from switch to switch, then cut one more piece. Drill 1/16-inch holes near both ends of all but one of the 1/4-inch strips. The undrilled piece is used as a temporary spacer, so make sure it is flat and deburred.

Lay the temporary spacer on top of the rectangular tubing between the first two switches, then drop one of the drilled 1/4-inch pieces over it, with the center switch lugs through the 1/16-inch holes. Before soldering, check the strip to make sure there's sufficient clearance between the 1/4-inch strip and the switch lugs; trim the corners if necessary. Use a screwdriver blade to hold the strip flat and solder the lugs to the strip. Remove the temporary spacer. Repeat this procedure for all switch sections. When you're done, you'll have a 50-Ω stripline running the length of the attenuator.

Next, solder in place the three resistors of each section. Use 1%-tolerance resistors and keep their leads as short as possible. If you must leave a resistor ground lead longer than you like, use a generous blob of solder to make the lead less inductive. Install a 1/2-inch-square brass shield between each 10-dB section to ensure that signals don't couple around the sections at higher frequencies.

Use parallel 1/4-inch strips of 0.005-inch-thick brass spaced 0.033 inch apart to form 50-Ω feed lines from the BNC connectors to the switch contacts at each end of the stripline (see Figure 4).

Finally, solder in place the remaining enclosure side, cut and solder the end pieces, and solder brass #4-40 nuts to the inside walls of the case to hold the rear (or bottom) panel. (You may find it easier and sturdier to solder the screws to the case walls.—*Ed.*) Drill and attach the rear panel and round off the sharp corners to prevent scratching or cutting anyone or anything. Add stick-on feet and labels, and your step attenuator is ready for use.

## Summary

You'll find the attenuator a useful tool. Used with a simple broadband detector,[2] it provides a convenient way to make absolute power measurements up to 250 mW. For antenna gain, F/B and F/S measurements, the attenuator can be used in line, with the receiver acting as the detector. Calibrated attenuator steps are much more accurate for this purpose than most receivers' S meters. Of course, the attenuator is also useful for performing receiver MDS and two-tone dynamic-range measurements.

Remember: This unit is built with 1/4-W resistors, so it can't dissipate a lot of power. Remember, too, that for the attenuation to be accurate, the input to the attenuator must be a 50-Ω source and the output line must be terminated in a 50-Ω load.

### Notes

[1]See also Bob Shriner, WA0UZO, and Paul K. Pagel, N1FB, "A Step Attenuator You Can Build," *QST*, Sep 1982, pp 11-13. PC-board material cut to size and with switch and jack mounting holes made for the 1982 attenuator *enclosure only* is available from FAR Circuits, 18N640 Field Ct, Dundee, IL 60118-9269. Price: $8, plus $1.50 shipping.

[2]See the sidebar "Measuring Double-Tuned Circuit Performance" on page 31 in Wes Hayward, W7ZOI, "The Double-Tuned Circuit: An Experimenter's Tutorial," *QST*, Dec 1991, pp 29-34.

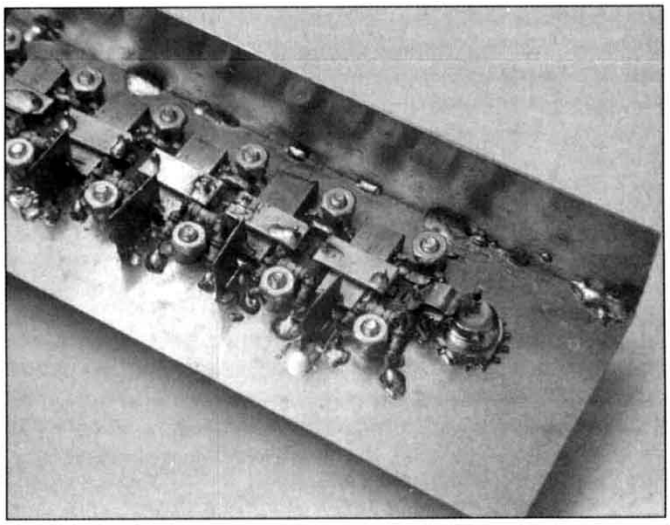


Figure 4—The attenuator before final mechanical assembly. The 1/4-inch strips are spaced 0.033 inch apart to form a 50-Ω connection from the BNC connector to the stripline. There are 1/2-inch square shields between 10-dB sections. The square shields have a notch in one corner to accommodate the end of the rectangular tubing.

# SOFTWARE FOR POWER/SWR METER

**Bill Kaune, W7IEQ**

## Microcontroller Overview

The purpose of this document is to provide a description of the software used in the power meter that I developed during the past few years and which is described in a QST article in the January, 2011 issue. To understand this document, you will need some basic familiarity with the PIC line of micro-controllers marketed by Microchip Inc and some familiarity with assembly language programming. If you have not already done so, I recommend that you download from Microchip's web site (www.microchip.com) and print a copy of the data sheet for the PIC16F87XA microcontroller. (The "data sheet" is actually 228 pages long). Reading this document can be tough going at times. I'll try to help by describing some of the tricks I learned to help me in understanding and remembering the contents of the data sheet.

I found it useful to make a second copy of pages 15-17, 19-21, and 160-165. Most of the questions that come up during the writing of a program are answered in these few pages.

I used the PIC16F876A for this project. This controller contains a 8-bit processor that is able to do addition, subtraction, and various logical, test, and data movement operations. The instruction set is fairly simple compared to most other processors that I am familiar with. There are only 35 instructions. By comparison the 8080 series of processors used in Windows (and now Macintosh) computers has many more instructions.

An interesting feature of this processor is that it contains separate memories for instructions and data. The PIC16F876A contains 8,192 flash memory locations for instructions. (Once a flash memory location is programmed, it retains its contents even when power is removed from the processor.) Each instruction is 14 bits wide. This processor also contains a set of 8-bit-wide file registers, described on page 17 of the manual. Some of these file registers are "special-purpose" registers used by the processor for various internal functions. Many of the special-purpose registers can be read and, in most cases, altered by the user. The remaining file registers are called general-purpose registers and are available to the user. In the PIC16F876A, there are 368 general-purpose file registers. The file registers are made from so-called dynamic access memory: They lose their contents whenever power is removed from the processor.

The PIC16F876A also contains 256 bytes (i.e., 8 bits wide) of electrically erasable programmable memory (EEPROM). These registers, which can be used to retain data even when the processor is turned off, are not used in my power/SWR meter program, but I have used them in other projects.

Each of the 8,192 instructions in the PIC16F876A instruction memory is given an address, starting at 0 and ending at 8,191. Since $8192 = 2^{13}$, the processor will need a total of 13 bits to encode all possible instruction addresses. The address of the next instruction to be executed is kept in an internal 13-bit register called the program counter register. Normally, instructions are executed in ascending order. However, two instructions (GOTO, CALL) contain address information that is used to determine the next instruction to be executed. Unfortunately, the instruction length of 14 bits is too short to contain both an operation code and a full 13-bit instruction address. Instead, these two instructions only contain the 11 least significant bits of an address. To complete the address, the instruction memory is divided into four pages. Page 0 extends from instruction address 0 to 2047, page 1 from 2048 to 4095, and so on. To fully specify an instruction location, an 11 bit instruction address has to be specified as well as the 2-bit page number. The page number is encoded using bits 3 and 4 of the PCLATH special purpose data register. Thus, if a program is executing in one page and you wish to go to an instruction in another page, or call a subroutine in another page, you must first set the appropriate bits in the PCLATH register before executing a GOTO or CALL instruction. Table 1 lists the PCLATH bit settings required to access various pages of instruction memory. For example, if it is desired to go to an instruction in page 3 from a location in page 0, it will be necessary to first set both bits 3 and 4 of PCLATH to 1 before executing the GOTO instruction.

The Power/SWR meter program being described here uses instruction located in all four pages of instruction memory.

A similar situation exists for the data file registers. Instructions that refer to the file registers have only room for 7 bits of an address, but in the 16F876A, 9 bits are required to fully specify a file register location. Thus, the file register memory is divided into four banks, numbered 0 through 3. The bank number is encoded by two bits (symbolically named RP0 and RP1) contained within the special-purpose STATUS register. The programmer has to make sure that these bits are set appropriately before executing any instruction that involves data stored in the file registers.

The Power/SWR meter program being described here uses data registers located both in banks 0 and 1. Table 2 shows how rp0 and rp1 must be set to access data in any of the four banks.

Table 1. Addressing for GOTO and CALL Instructions

| Page | Range of addresses | PCLATH Bit 4 | PCLATH Bit 3 | Portion of address included in instruction |
|---|---|---|---|---|
| Page 0 | 0 to 2047[1] | 0 | 0 | 0 to 2047[1] |
| Page 1 | 2048 to 4095[2] | 0 | 1 | 0 to 2047[1] |
| Page 2 | 4096 to 6143[3] | 1 | 0 | 0 to 2047[1] |
| Page 3 | 6144 to 8191[4] | 1 | 1 | 0 to 2047[1] |

[1]In hexadecimal, 0 to 7FF [2]Hexadecimal 800 to FFF

[3]Hexadecimal 1000 to 17FF [4]Hexadecimal 1800 to 1FFF

Table 2. File Register Addressing

| Register Bank | Address Range | rp1 | rp0 | Address contained in instruction |
|---|---|---|---|---|
| 0 | 0 to 127[1] | 0 | 0 | 0 – 127[1] |
| 1 | 128 to 255[2] | 0 | 1 | 0 – 127[1] |
| 2 | 256 to 383[3] | 1 | 0 | 0 – 127[1] |
| 3 | 384 - 511[4] | 1 | 1 | 0 – 127[1] |

[1]In hexadecimal, 0 to 7F [2]In hexadecimal, 80 to FF

[3]In hexadecimal, 100 to 17F [4]In hexadecimal, 180 to 1FF

What makes the PIC a micro-controller rather than just a microprocessor is that in incorporates various peripheral units in addition to its core numeric processor. The application being described here utilizes two of these peripheral units: A 10-bit analog-to-digital converter (ADC) and a pulse-width-modulated (PWM) output module. These will be described in more detail later.

**Power/SWR Meter Hardware Overview**

The hardware of the Power/SWR Meter is described in a QST article (January, 2011). In summary, RF power being generated by a transmitter is routed through a directional coupler on its way to an antenna tuner and an antenna (the load). The directional coupler contains passive circuits that sample the power flowing from the transmitter to the load (the so-called forward power) and, separately, the power flowing, after reflection, back from the load to the transmitter. Thus, the directional coupler has two outputs, one for the forward power sample and the other for the reflected power sample. Normally, the antenna and/or antenna tuner will be adjusted to minimize the reflected power flowing towards the transmitter.

The forward and reflected power samples are routed to two integrated circuits in the power meter whose outputs are voltages proportional to the logarithms of the rf input power. That is, the outputs are proportional to dBm (decibels relative to 1 milliwatt). The time constants in these two circuits are adjusted so that the outputs will follow the modulation envelopes of the input signals.

The dBm outputs are routed to two sample-and-hold integrated circuits that operate under the control of the 16F876A controller. The controller can send a signal to the two sample-and-hold units that will cause them to freeze (hold) their outputs at the input level present when the signal from the controller was received. In this way the forward and reflected power signals present at any particular time in the transmission line from the transmitter to the load can be held. The processor then reads in the forward signal and converts it (i.e., digitizes it) from an analog voltage to a corresponding 10-bit binary number. It requires 19.2 microseconds (μs) to perform this conversion. It then reads the reflected power signal and converts it to a 10-bit number.

The analog-to-digital conversions are always done with respect to some voltage reference. An input voltage equal in amplitude to the reference voltage will yield a converted binary value of the maximum possible, in the case of a 10-bit converter 1023. The reference voltage used in my unit comes from a special integrated circuit that produces a very stable 2.50-volt output.

Once the forward and reflected powers are converted into internal 10-bit numbers, the processor computes the peak and average values of a large number of such samples and calculates the

standing-wave-ratio (SWR). The averaging time can be either short (about 1.6 seconds) or long (about 4.8 seconds) and is determined by the setting of the front panel "Averaging" switch shown in Figure 1. Depending on the setting of a second front-panel "Power" switch, the processor then calculates the peak and average forward powers, or the peak and average load (i.e., forward minus reflected) powers, present on the transmission line and displays these values using a two-line, 20 characters per line, liquid crystal display. In addition, every 0.1 second, the processor calculates the SWR and updates a front-panel analog meter, thus showing on an effectively continuous basis SWR. This is very useful for tuning purposes.

The remainder of this document will describe in more detail the software and how it accomplishes the tasks summarized above.

**Power/SWR Meter Processor Circuit Connections**

My article in QST (January, 2011) describes in detail the overall operation of the circuit in the Power/SWR Meter. Here, I will describe in more detail those parts of the circuit that directly connect to the PIC16F876A processor.

Figure 2 is the schematic of the meter. U5 is the PIC16F876A processor. Power (+5 V) is supplied to the processor through pin 20 and ground connection is through pins 8 and 19. Pin 1 is the so-called master clear input (MCLR). When power is first supplied to the processor, it will not start executing instructions until the signal on pin 1 goes high. It is important that pin 1 not go high until the source voltage to the processor has stabilized. R3 and C17 form a network whose effect is to prevent the signal on pin 1 going high for several tenths of second.

Pin 2 is the analog input for the forward channel power signal and pin 3 is the corresponding input for the reflected power channel signal. Pin 4 is configured is a digital output and provides the control signal for U3 and U4, the sample-and-hold chips for the forward and reflected power channels, respectively. Pin 5 is the input for the very stable +2.50 volts generated by U7 that is used as the reference against which analog-to-digital conversions are performed.

Pins 9 and 10 connect to a 20-MHz crystal that determines the clock frequency for the processor. The period of the clock is 50 ns (nanoseconds). In PIC 16-bit processors, four clock cycles are required for the execution of each instruction. Thus, each instruction requires 200 ns to execute.

Pins 11 and 12 are not used. Pin 13 provides a rectangular wave output (pulse height ≈ 5.0 V) whose frequency is fixed at about 4.9 kHz. The pulse width, however, can be varied in software from 0% (i.e., output is always 0 V) to 100% (output is always 5 V) in 256 steps. The signal from pin 13 is directed to the network R7 and C21 which act as a low-pass filter to extract the dc component, that is, the average of the pulse signal. This signal is buffered by U6 and drives the front panel analog meter M1 shown in Figure 1. R8 should be adjusted so that a signal with a 100% duty factor will result is a full-scale deflection of M1.

Pins 14 and 18 are configured as digital inputs and are used to sense the position of the two front panel switches (Figure 1). One switch (SW1) sets the averaging time as either short (about 1.6 s) or long (about 4.8 s). The other switch (SW2) determines whether the power meter displays the peak and average forward power, or the peak and average load power (i.e., forward – reflected power), using its LCD display.

Pins 15-17 are digital outputs that send control signals to the LCD display and pins 21-28 send data to the LCD display. The display I used features two lines of 20 characters each.

**Software Overview**

A complete listing of the processor program for my Power/SWR meter is included in the Appendix. Don't worry about the length. One half of the program is a series of lookup tables that will be described shortly, another quarter is a floating-point package that I obtained from the Microchip website, and the last quarter is the actual operating software.

When the power meter is turned on, the software starts at instruction address 0. Some initialization is performed, and the software then moves into its normal operation. The software proceeds to repetitively hold and acquire samples of the forward and reflected powers. With the parameters I currently use, it takes 1,111 samples during one "cycle". Each sample requires 90 μs (microseconds) to acquire, so all 1,111 samples require 0.1 seconds. As they are acquired the software keeps track of the peak (i.e., maximum) forward power and the corresponding reflected power. As described earlier, the power signals that the processor directly measures are actually proportional to dBm. During each sample, the software converts the forward power from dBm to watts. In order to speed this process, the conversion is done using a lookup table. The lookup table for the forward power is held in page 2 of program memory. The software also sums all 1,111 values of the forward power, in watts, in preparation for calculating the average power

At the end of one cycle, the software checks to see if the new peak value exceeds the previous peak value then being displayed on the LCD display. If it does exceed the currently displayed value, the LCD display is updated to reflect the new peak forward (or load, depending on the setting of the front panel "Forward/Load" switch) power, and the SWR value being displayed is also updated, using the new peak forward power and corresponding reflected power for the SWR calculation. The new peak value will be held in the display until either a larger peak value is encountered in a subsequent cycle or a given number of cycles are executed without a larger peak value being encountered. In this latter event, after the given number of cycles is executed, the display is updated using the new peak value measured during the next cycle.

The SWR is calculated at the completion of each cycle, using the peak forward and corresponding reflected powers measured during that cycle. As noted above, this new value will be written to the display if the new peak forward power is larger than the forward power being displayed. However, it is always used to update the analog front panel meter display. Thus, this meter display is updated approximately every 0.1 seconds (actually the time to complete one cycle plus to overhead to calculate the SWR). The SWR is not displayed directly on the meter. Rather, the quantity $1-1/SWR$ is displayed, where a value of 1 corresponds to the full-scale position of the meter needle. Doing this makes the front panel meter a more effective tuning aid. If the SWR is 1, indicating a perfect match, the meter will read 0. If the SWR is 2, the meter will read 0.5, that is at half scale. If the SWR is 5, the meter reading will be 0.8, and so forth.

The procedure I use to update the peak forward (or load) power displayed on the LCD display may seem a little strange. I adopted this approach because I was concerned that updating the display after every cycle would result in a flickering and, thus, unreadable, displayed value of the peak power. The way I have it now programmed, the display will hold a peak power value for a maximum of 30 cycles (about 3 s) unless an even larger peak value is encountered before this period has expired. Once a larger peak value is measured, the display is updated and the 30-cycle waiting period is started again.

The *average* forward (or load) power is not updated after every cycle. Instead, it is updated after a specified number of cycles have been completed, a period which I call a "series". The number of cycles in a series is controlled by the front-panel averaging switch. In the "Short" position, the average will be updated every 16 cycles (a little more than 1.6 s), whereas in the "Long" position, the average will be updated every 48 cycles. I generally operate with the "Short" setting.

**Software Details**

Development System

In this section, I will get into the details of the software. It will help if you refer to the program listing in the Appendix. The software was written in assembly language and was assembled using the free assembler available from Microchip (www.microchip.com). The program was debugged using Microchip's IDE development system, which is also free. Once the program was debugged, a 16F876A controller was programmed using Microchip's PicStart Plus Development Programmer. This programmer is not cheap. There are other less expensive programmers that might be used for this purpose.

When I use the Microchip assembler, I always set software flags so that numbers are interpreted as decimal, unless they are explicitly preceded by "0x" in which case they are interpreted as hexadecimal. Thus, the number 71 will be interpreted as decimal 71, whereas 0x71 is hexadecimal (= 113 decimal).

The assembler program understands a substantial number of so-called assembler directives. For example, if one wishes to set the bank number to some value (between 0 and 3), one can write the code directly to do this (see Table 1) or one can, instead, use the assembler directive

```
bank    Address
```

where Address is an address located in the desired bank. When the program is actually assembled, the assembler will replace this directive with the required machine instructions.

Similarly, if one wants to set the page number to a desired value before executing a GOTO or CALL instruction, one can directly manipulate bits 3 and 4 (see Table 2) or one can instead include the directive

```
page    Address
```

where Address is an address located in the desired page. For various reasons (which might not make much sense), I have mostly chosen not to use these directives but to, instead, write code that directly sets/clears bits 3 and 4 of PCLATH and bits rp0 and rp1 of the status register.

I will mention other assembler directives as they are used.

Simulator and Simulate Mode

Microchip's IDE development system includes a simulator that I used to develop and test most parts of the program. This simulator, however, is not very effective at testing the analog-to-digital converter, the pulse-width-modulated unit, the front-panel analog meter, and the liquid crystal display. (In fact, the program will hang if you attempt to run it in the simulator.) To deal with this problem, I wrote the program to enable the user to select a simulate mode, where several of the components of the program are bypassed. At the very beginning of the program (Appendix 1, page 1), the user can set a variable, "Simulate" to 0 or 1. Setting it 1, by replacing the line

```
Simulate equ  0
```

with

```
Simulate equ 1
```

enables simulate mode. (The assembler directive equ simply defines a symbolic variable—Simulate in this example—and set its value to the indicated number. Whenever the assembler encounters this symbolic variable, it will replace it with its defined numerical value.)

Following this are two constants, ForwardOrLoadWatts and LongOrShortAvgTime, that are used only in simulate mode.

Note that, in any line, text following a semicolon character is treated as a comment and is ignored by the assembler.

Next in the listing is the definition of the variable LowPowerThreshold. When the peak forward power measured during a cycle of 1,111 measurements results in a digitized 10-bit value less than LowPowerThreshold, the processor writes the word "Low" to the LCD display. This variable is currently set to a value corresponding to an input forward power of about 1 mW (milliwatt).

Calibrate Mode

A little later on the first page, a new constant, "Calibrate" is defined. If Calibrate = 0, the normal mode of the program is enabled. However, if Calibrate = 1, a different program is assembled. The calibrate program is used for calibrating my power/SWR meter. This program continually reads the forward and reflected power channels 1024 times, then computes the average of the string of digitized 10-bit values, and displays the resulting 10-bit averages on the front panel display. As discussed in my QST article (January, 2011), the response of the power meter to an input RF signal can be written

$$dBm = \alpha \times ADC + \beta \ ,$$

where *dBm* is the measured known power input to the power meter and *ADC* is the recorded digitized value of the power. By inputting known powers and recording the resulting ADC values, one can experimentally determine the slope, $\alpha$ and intercept, $\beta$, for the above equation and, thus, calibrate the channel.

In addition, the calibrate routine outputs a maximum signal to the front panel meter (i.e., the pulse train on pin 13 of U5 has a 100% duty cycle). The variable resistor R8 should be adjusted so that the meter reads full scale.

Constant Definitions

Next in the program listing is a series of constants. The first constant is MaxNumPeakWaits. As noted above, the program will display a peak value measured during its first cycle and then either display a new peak value measured during a subsequent cycle or, if a larger peak value is not encountered, continue to display the first peak value for a specified number of cycles before updating the display with the next measured peak value. The specified maximum number of cycles during which a peak value is held is defined by the value of MaxNumPeakWaits. I use 30 for this, which corresponds approximately to a maximum wait of about 3 s.

The next constant is MaxNumSWRMeterWaits. Assume that a valid forward and reflected power measurement is followed by a series of measurements with peak forward powers less than 1 mW, that is, too low to display and trust. The power meter will continue to display on its front-panel analog meter the SWR measured during the last valid cycle until a specified number of

cycles of low power readings is taken, at which point the SWR meter will be set to display 0. MaxNumSWRMeterWaits specifies the number of cycles.

The next constant is NumSamples. This sets the number of samples taken in one cycle. Currently it is set to 1,111 so that one cycle takes 0.1 s to complete. (Actually, NumSamples is set equal to 100,000/SamplePeriod. SamplePeriod is the time in microseconds to make one measurement, currently set to 90. 0.1 s expressed in μs is 100,000. Dividing the latter by the former yields 1,111.11111111…, which rounds to 1,111.)

The following two constants, NumCyclesShort and NumCyclesLong, define the number of cycles in a "Short" or "Long" series. Remember that a front panel switch selects between Short and Long (Figure 1).

The next section of the listing in Appendix 1 describes the 16-bit floating point format that I use in part of the program. I will discuss this format a little latter in this document.

Memory Allocations

Go to page 2 of Appendix 1. Part way down the page begins a section labeled "Constant and Memory Allocations". The section provides the assignment of data memory addresses. The first statement is

```
cblock    0x20
```

This informs the assembler that this block of data memory starts with the address 0x20 (a hex number = 32 decimal). If you look at page 17 in the data sheet for the PIC16F876A, you will see that 0x20 is the first address of the general-purpose data registers located in bank 0. The following lines define the symbolic names for the registers in this block. Thus, 0x20 is given three names "exp", "aexp", and "aarg". These registers, as well as all the registers through "bargb2" are used by the 24-bit floating point package supplied by Microchip. I will describe most of the following registers as I describe the routines that use them. Note that a symbolic name without any modifier causes the allocation of one 8-bit data register. However, if the name is followed immediately by a colon and number, the number of registers allocated to the name will be determined by the number. Thus, the appearance of a name like FP:2 will cause the allocation of two consecutive data registers. On the other hand, aexp:0 will not allocate any data registers to the name.

Definitions of the data registers in bank 0 continue until the “endc” statement that can be found on page 3 of the listing. Later on this page are the definitions of data registers that the program uses at the very top of bank 0 (starting with the address 0x7D), and data registers located in bank 1, starting with the address 0xA0.

The material following defines a large number of symbolic names used by the Microchip floating point library routines and by my program.

Macro Definitions

We next come to a series of macros. Macros define sections of code that are used at multiple points in the program. Rather than having to type out these instructions repeatedly, we define a macro. Anytime this macro appears in the stream of instructions, the assembler replaces it with the block of code specified by the macro definition.

The first macro is MoveBytes. This macro name is defined by the statement

```
MoveBytes    macro    AA, BB, nbytes
```

The code that defines this macro follows until the “endm” assembler directive is encountered. This macro moves a number of bytes, given by the argument “nbytes”, from the data registers starting at the address “AA” to the data registers starting at the address “BB”. This macro will properly move bytes from bank 0 to bank 0, bank 0 to bank 1, bank 1 to bank 0, or bank 1 to bank 1. For example, suppose that one wishes to move three contiguous bytes of data, with the starting address aexp, to a location with starting address t1. To accomplish this, one need only write the macro

```
    MoveBytes     aexp,t1,3
```

and the appropriate machine instructions will automatically be generated by the assembler.

The next macro, StoreFPLiteral, stores a 24-bit literal floating point value into a given data register located in the current bank. (Details about 24-bit floating point values will be given shortly.) For example, suppose it is desired to store the number 1.0 into the registers beginning at

“aarg”. (The 24-bit floating point literal for 1.0 is 0x7F0000.) This can be done by writing the macro

```
StoreFPLiteral    0x7F0000, aarg
```

The next macro is named LCDDisplayText. The purpose of this macro is to write to the LCD display text and/or numbers that do not change during the course of the program’s operation. I am going to spend some time describing this macro because it is the first use of lookup tables, which are implemented in PIC processors in what was for me, at least, unique and interesting.

Suppose we want to be able to look up the various characters within a text string, for example, the string “abcd”. The way to do this using PIC processors is to first construct a series of four instructions as follows:

```
retlw     “a”
retlw     “b”
retlw     “c”
retlw     “d”
```

The instruction “retlw” means to return from a subroutine with a 8-bit literal value in the w register. (The w, or “working” register, is a special register in the PIC that is involved in most operations.) In this case, the first “retlw” will have encoded into it the ASCII code for the letter a, the second the code for “b”, and so on.

The second thing we need is a pointer to the letter we wish to retrieve. In this case, the pointer will have the values 0 (for “a”), 1 (“b”), 2 (“c”), and 3 (“d”).

Finally, we need a subroutine that is called with a value of the pointer and which, somehow, arranges to execute the “retlw” instruction that corresponds to the pointer value. The PIC 16-bit series of processors has what, in my experience, is a unique way to accomplish this. The PIC has an internal 13-bit register that keeps track of the next instruction that it is to be executed, the so-called program counter register. The lower 8 bits of this register are located in a special purpose file register named the PCL register. The upper 5 bits are held internally and are not directly accessible to the programmer. The interesting thing about the PIC 16-bit series is that a program can write directly to the PCL, in which case the next address to be executed is constructed from the 5 least significant bits in the PCLATH register and the new 8 bits in the PCL. An example of

this usage is my macro LCD_DisplayText that starts on page 5 of Appendix 1. To use this macro, the program must contain a statement with the following format:

```
LCD_DisplayText      TextString, UniqueAddress
```

TextString is the text to be printed (surrounded by quotation marks) and UniqueAddress is unique symbol that has the general format of an address. During the assembly of the program, the macro is replaced with the code following the macro definition starting on page 5. The code starts with 20 PIC instructions and then is followed by the assembler directive

```
dt        TextString,0
```

Suppose that TextString = "abc". The assembler directive, dt, above results in the inclusion of four "retlw" instructions. The first includes the ASCII code for "a", the second code for "b", the third code for "c", and the fourth the literal value 0. This last value is used to signal the end of the lookup table.

Now take the instructions that precede the "dt" directive. The first instruction sets the file register bank to 0 and the next two instructions set the instruction memory page to 0. The next instruction sets the text pointer, named LCDTextPtr, to 0 so that it points to the first character in TextString. We then move into a section of code that loops through all characters in TextString. We first call the subroutine that immediately follows this section of code. For this call, we use a relative address, $+7. The symbol $ means the current address. Thus, we are called a subroutine that begins 7 instructions further on in the code.

Upon entry, the subroutine temporarily stores the pointer and then loads the top 5 bits of the address of the lookup table (at $+7) into PCLATH. It then adds the pointer value to the bottom 8 bits of the address of the lookup table and increment PCLATH by 1 if there is a carry resulting from this addition. Then the bottom 8 bits are loaded into PCL, which causes the top 5 bits of the address counter to be taken from PCLATH. The next instruction executed is thus the appropriate entry in the lookup table. Since this instruction is "retlw", control is returned to the instruction following the original subroutine call, but now with the appropriate ASCII code in the w register. The program then checks to see if the w register contains 0, which would signify that all characters in the lookup table have been retrieved. If w is not 0, the character in w is written to the LCD display, via a call to the subroutine LCD_SendChar, the pointer is incremented by 1, and control is sent to the beginning of the loop to process the next character.

The next macro we encounter is ADC_Initialize. As it name suggests, this macro simply initializes the analog-to-digital converter internal to the PIC16F876A processor. This processor contains two special purpose file registers (ADCON0 and ADCON1) that control the ADC. In my application, the ADC is configured to use two input channels, channels 0 (denoted AN0 in PIC literature) and 1 (AN1). In addition, the ADC is configured to utilize an external reference voltage. Externally, channel 0 is connected to pin 2 of the processor chip, channel 1 to pin 3, and the reference voltage to pin 5. The ADC module requires its own clock signal, which is normally obtained from the main processor clock through a division process. The period of the ADC clock must be at least 1.6 μs, which means that since the period of the processor clock is 50 ns, the processor clock must be divided by (at least) 32. I use 32.

To initiate an ADC conversion, bit 2 of ADCON0 must be set. Bit 2 will remain set until the conversion is complete, at which point the ADC module will clear point bit 2. The 10-bit result of the conversion is returned in two special purpose registers, ADRESH in bank 0 and ADRESL in bank 1. The ADC can be configured to return the 10-bit result either right-justified or left-justified in these registers. I chose to have it left-justified (i.e., most significant 8 bits in ADRESH, least significant 2 bits in bit positions 7 and 6 of ADRESL). I no longer remember why I made this choice, but were I writing this program again, I might select right justification.

The next macro, ADC_StartRead, initiates an ADC conversion. The following macro, ADC_ReadFinished, simply monitors bit 2 of ADCON0 until it returns to 0 thereby indicating that the ADC conversion is finished. The following two macros, ADC_SetCh0 and ADC_SetCh1, configure the ADC to read either channel 0 or 1, respectively.

Initialization Code

We now come to the actual instruction code for the processor. When power is first applied to the processor, and the Master Clear signal on pin 1 goes high, the processor starts by executing the instruction at location 0 in its instruction memory. The processor incorporates a fairly elaborate interrupt system (more details later), and any interrupt causes control to transfer to the instruction located at address 4. My program uses one type of interrupt, so there is an interrupt service routine located at address 4. Thus, when the processor initially starts, the first instruction is a GOTO instruction that simply jumps around the interrupt service routine to the address StartUp. This address is located on page 7 of the program listing in Appendix 1.

In StartUp, the first thing the processor does is wait for 1 second (actually 10 tenth-of-one-second waits). It then sets the appropriate states of the various input-output pins of the processor. These states are controlled by three special purpose registers, TRISA (i.e., Tri-State Port A), TRISB, and TRISC. A bit set to 1 in one of these registers configures the respective pin in the respective port as an input whereas 0 makes an output.

My program includes three subroutines that can be used to wait for various lengths of time. We have already mentioned WaitTenthSeconds. The number of periods that one desires to wait is entered in the w register. Any value from 1 (0.1 s wait) to 255 (25.5 s wait) can be entered into w. The next routine is WaitMilliSeconds, which will wait from 1 ms (w = 1) to 255 ms (w = 255). Finally, WaitMicroSeconds will wait from 3 μs (w = 3) to 255 μs (w = 255). This routine will not act properly if w = 1 or w = 2. (These subroutines will only wait for these times on processors clocked at 20 MHz.)

Next the LCD display is initialized by calling the suboutine LCD_Init. I won't discuss the initialization procedure here. Suffice it to say that the procedure is rather complicated and, in some respects, mysterious. The display I used is documented in a manual[2]. This manual, however, is of little use because it does not provide any information on programming the display. The display uses a Hitachi controller/driver integrated circuit, and the programming of this device is well documented in a Hitachi user manual[3]. I also purchased a book[4] on programming PIC processors that I found helpful for programming this kind of display.

After the display is initialized, the text "W7IEQ Power Meter" is written to the first line of the display. (After the display is initialized, the "cursor" is set to character 1 of line 1.) The program next moves the cursor to character 1 of line 2 and writes out the text "Version 2". The program then waits for 3 seconds, enough time for the user to read the display.

The next step is to initialize the analog-to-digital converter module. This is done with the macro "ADC_Initialize" which has already been discussed. We next place the two sample-and-hold chips into sample mode by setting bit 2 of IO port A to 1. The pulse-width module (PWM) is next initialized by calling the subroutine "PWM_Initialize". The module is set to produce a steady train of pulses at a frequency of about 4.9 kHz. Initially, the pulse width is set to 0%, but it can be changed in software with 256 steps between 0 and 100%.

The next three steps in the program set up the interrupt system. (I will give details later.) These three instructions are:

```
bsf     status,rp0
bsf     0x7F & PIE1,TMR1IE
bsf     INTCON,PEIE
```

The two special-purpose registers, PIE1 (Peripheral Interrupt #1) and INTCON (Interrupt Control) that we wish to manipulate are located in bank 1 of the file registers. To reach bank 1, we must set the rp0 bit in status to 1, which the first instruction listed above does. The next instruction sets the bit TMR1IE (timer 1 interrupt enable) to 1.

Notice the odd construction 0x7F & PIE1 in the second instruction listed above. PIE1 is an address of a register located in bank 1 of the file registers. The Microchip assembler seems to always issue a warning whenever an address in a bank other than bank 0 is referenced in an instruction. Since, in my program, there are many such references, the list of warning messages the assembler produces is very lengthy. To avoid this, I precede all such addresses referenced in an instruction with the construction "0x7F & ". The symbol means logical bit-by-bit AND and has the effect of eliminating all but the least significant 7 bits of the address. But, all such 7 bit addresses are located in bank 0, so doing this avoids the warning message.

(I suspect there must be another way to avoid this problem. Perhaps the assembler has a switch that disables this particular warning message. However, I have looked and have never been able to find such a switch.)

24-Bit Floating Point Routines

This power meter is designed to respond to power levels from 1 mW to over 1,000 W. To handle such a range of powers, I early-on decided to use floating point arithmetic in most of the program. I searched the Microchip web site (specifically Application Notes) and found both 24-bit and 32-bit floating-point libraries. I chose to use the 24-bit floating-point library. The Microchip floating-point library is located in page 1 of instruction memory.

It is common to express decimal numbers (particularly very large and very small numbers) in power-of-ten format. For example, the speed of light, in meters per second, is 299,790,000 m/s. This would, most commonly, be written $2.9979 \times 10^8$ m/s.

A similar situation exists with binary arithmetic. Two formats are typically used, fixed point and floating point. In floating point, all binary numbers are written in the format

$$1.b_{-1}b_{-1}\cdots b_{-m}\times 2^{n},$$

where *m* is the number of significant digits and *n* is an integer exponent. Note that this number starts with the digit 1. Because binary has only two bit values, 0 and 1, *all* numbers start with the digit 1 except for 0. For the 24-bit format used by the Microchip library, *m* = 15. Thus, for example the numbers 1.0, 3.14159, and -0.1 would be written in this binary floating-point format as

$$\begin{aligned} 1.0 &= 1.000\ 0000\ 0000\ 0000\times 2^{0}\\ 3.14159 &= 1.100\ 1001\ 0001\ 0000\times 2^{1}\\ -0.1 &= -1.100\ 1100\ 1100\ 1101\times 2^{-4} \end{aligned}$$

The binary value is called the mantissa.

A 24-bit floating-point value consists of 24 bits (3 bytes) that encode this value. The first 8 bits contain the value *n* + 127, were *n* is the exponent (by adding 127 to the exponent, we avoid negative values) and *n* is restricted to the values $-126 \le n \le 128$. With this restriction, $1 \le n + 127 \le 255$. The value of 0 for the exponent byte is then reserved to signify that the overall number is 0. Since the leading bit of all nonzero mantissas is 1, we only include the 15 bits to the right of the decimal point. The 16th bit is used to indicate whether the number is positive or negative, with the value 1 indicating the number is negative. Thus the above three numbers, written in Microchip format, are

$$\begin{aligned} 1.0 &= 0111\ 1111\ 0000\ 0000\ 0000\ 0000 = 0\text{x}7\text{F}0000\\ 3.14159 &= 1000\ \ 0000\ 0100\ 1001\ 0001\ 0000 = 0\text{x}804910\\ -0.1 &= 0111\ 1011\ 1100\ 1100\ 1100\ 1101 = 0\text{x}7\text{BCCCD} \end{aligned}$$

The Microchip floating-point library contains subroutines to add (FPA24), subtract (FPS24), multiply (FPM24), and divide (FPD24) values. In all cases, the first argument must be placed in the data register aarg, aarg+1, aarg+2 and the second in barg, barg+1, barg+2. The result of the operation is returned in aarg, aarg+1, and aarg+2. The library also includes routines that convert 16-bit binary integers into 24-bit floating-point values (FLO1624), 24-bit integers into 24-bit floating point values (FLO2424), and 24-bit floating point values into 16- or 24-bit integers

(INT2416 and INT2424, respectively). Finally, the library includes subroutines that determine whether the number in aarg is < the number in barg (TALTB24), aarg ≤ barg (TALEB24), aarg > barg (TAGTB24), aarg ≥ barg (TAGEB24), aarg = barg (TAEQB24), and aarg ≠ barg (TANEB24). The entire Microchip library occupies page 1 of instruction memory. In addition to the Microchip routines, I included one additional routine that I wrote named AargTimes10 that multiplies the value in aarg by 10. This routine uses the fact that 10*aarg = 2*aarg + $2^3$*aarg to more quickly do this multiplication.

There were many times during the development and testing of this program where I needed to convert decimal numbers into the Microchip 24-bit floating-point format, or visa versa. To simplify this, I wrote a program for my programmable hand calculator ( a HP 48GX calculator). Since this calculator has built-in capability to convert decimal integer numbers to binary integer numbers, or the inverse, it was relatively simple to write this program. The program consisted of the following steps.

1. If $x = 0$, set the floating point value to 0x000000 and quit.
2. If $x < 0$, make it positive and continue with the following steps.
3. Let $x$ be the number to be converted. If $x$ is < 1, multiply repetitively by 2 until $x$ is ≥ 1. Keep track of the number of required multiplications. The negative of this value becomes the exponent.
4. If $x \geq 2$, divide repetitively by 2 until $x < 2$. Keep track of the number of required divisions. This value becomes the exponent.
5. Add 127 to the exponent deduced in Step 3 or Step 4 to obtain the offset exponent. Convert the resulting value to an 8-bit binary value. This is the first byte of the floating-point value.
6. Multiply the remaining value of $x$ by $2^{15}$, round it to the near integer value, and convert the resulting integer to a 16-bit binary value. (When the fractional value is exactly 0.5, round to the nearest even value. For example, if multiplying $x$ by $2^{15}$ yields 32768.5, round to 32768. But if the resulting value is 32769.5, round to 32770.)
7. If the original value of $x$ was positive, change the $16^{th}$ (i.e., most significant bit) to 0. If $x$ was negative, leave the $16^{th}$ bit = 1. The resulting 16-bit value comprises the second and third bytes of the floating-point value.

More Initialization

Beginning on page 8 of the program listing in Appendix 1, the Microchip floating-point package is initialized by setting certain flags. Following this, the program proceeds to calculate two normalization constants, FPInvTotalSamplesShort and FPInvTotalSamplesLong. Earlier, I explained that the power meter program computes the average forward power during a series of samples. A series consists of a specified number of cycles, NumCyclesShort for the short averaging time (selected from the front panel "Averaging" switch) and NumCyclesLong for the long averaging time. Furthermore, each cycle consists of NumSamples measurements of the forward power. Thus, to compute the average, we must sum all the forward power samples and then divide by NumSamples multiplied by NumCyclesShort for short averaging or NumSamples times NumCyclesLong for long averaging. This initialization simply calculates the quantities 1/(NumSamples*NumCyclesShort) and 1/(NumSamples*NumCyclesLong) for later use by the averaging routine.

Following along in the listing, the routine first loads the 16-bit integer value of NumSamples into aarg and aarg+1 and converts it to a floating-point value. Note that when I call a subroutine in page 1, I mask the address to keep only the least significant 11 bits. (This masking is done by performing a bit-wise logical and between the subroutine address and the number 0x7FF.) I do this to prevent the assembler issuing messages warning that the page number has to be set correctly. I do set the page number to page 1 via the following instruction:

```
    bsf     PCLATH,psel0
```

The resulting floating-point value of NumSamples is saved in t1 and is also placed in barg. Next, the NumCyclesShort is loaded into aarg+1 and aarg is cleared. (NumCyclesShort must be less than 256.) This value is then converted to floating point, and then multiplied by barg, that is, by NumSamples. The result is transferred to barg, and aarg is then loaded with the floating point value for 1.0, namely 0x7F0000. Aarg is then divided by barg, yielding finally 1/(NumSamples*NumCyclesShort). This result is saved in the 24-bit register FPInvTotalSamplesShort.

The next segment of code calculates and saves 1/(NumSamples*NumCyclesLong).

The next step in the initialization is to clear (i.e., set to 0) a data register named ModeStatusFlags. The various bits in this 8-bit register are used to keep track of certain conditions, as defined on page 2 of the listing. The conditions are covered are:

Bit 0 = 0 (or 1) means the peak forward power (in watts) measured during a cycle is greater than or equal to (or is less than) the low power threshold, defined by the constant LowPowerThreshold.

Bit 2 = 0 (or 1) means that the LCD display is showing the peak and average load (or forward) powers.

Bit 4 = 0 (or 1) means the average time switch in the in short (or long) position.

Bit 5 = 0 (or 1) means peak and average powers are being displayed with units of watts (or milliwatts)

The program next initializes the 16-bit register bForwardPeak to 0, the 8-bit register PeakWaitCounter to the value of the constant MaxNumPeakWaits, and the 8-bit register SWRMeterWaitCounter to the constant MaxNumSWRMeterWaits. The functions of these registers will be discussed later. Finally, the two-line LCD display is cleared and the text "SWR" is written near the right edge of the first line.

We now come to the line with the address StartSeries. (Any symbol that starts in column 0 of the listing is interpreted as an address by the assembler.) This point in the program will be entered whenever a new series of cycles is started. At this point, the program first sets bits to insure that page 0 of program memory and bank 0 of data memory are set. It then determines the setting of the front-panel Forward/Load Power switch (Figure 1). This switch is wired to bit 3 of input/output port C. Placing the switch in the "Forward" position will cause the level on this input to be high (i.e., 5 V) as shown in Figure 2.

The program then continues by further initializing the display by writing additional text that does not change during a measurement series. The program then determines the setting of the front-panel averaging time switch (Figure 1). This switch, wired to bit 7 of input/output port C, determines whether the averaging time is short or long. It sets or clears the appropriate bit in ModeStatusFlags and places the appropriate number of cycles per series into the 8-bit register CycleCounter. It then clears the 32-bit accumulator Fpaccum that is used to accumulate the running sum of measured forward powers. At the end of a cycle, the value in this accumulator will be divided by the total number of measurements in the cycle to determine the average The

program loads the 16-bit register SampleCounter with the number of measurements to be taken in the cycle and sets/clears one bit in the 8-bit register Flags. It then clears the 16-bit register bForward and calls the subroutine ISR to start the cycle.

During a cycle, a substantial number of measurements are taken. With the current settings in the program, 1,111 measurements are taken during one cycle, and measurements are repeated at 90 μs intervals. This timing is set using the interrupt system and an internal timer in the processor. Thus, let me digress briefly to discuss the interrupt system.

Interrupt System

Together with many types of processors, PIC processors have the capability of responding immediately to certain external and/or internal conditions. When such a condition occurs, the processor is interrupted, the instruction being executed is finished, the location of the next instruction to be executed is saved, and the processor's execution stream is diverted to an "interrupt service routine (ISR)" that starts at address 4 in program memory. Normally, this routine will perform actions to "service" the interrupt condition and will then return execution to the normal instruction stream using the memory location saved earlier. Let me now explain how this capability is used in my program.

The most basic activity of the program is to repetitively sample the forward and reflected powers between a transmitter and a load. As noted earlier, a sample is taken every 90 μs. This timing is implemented by having a clock (timer) internal to the processor generate interrupt signals every 90 μs to initiate a sample. To do this, I used the processor's internal 16-bit timer (called TIMER1 in the PIC16F876A data sheet). The contents of this timer are incremented by 1 every instruction cycle. When it reaches its maximum value (0xFFFF) and the next increment arrives, the counter's contents overflow to 0 and an interrupt is generated. The processor will then stop whatever it is doing, save its current location in instruction memory (by placing the address of the next instruction to be executed at the top of an internal stack), and then transfer to location 4 in instruction memory. The program has an interrupt service routine (ISR) starting at location 4 that saves the contents of the w and status registers, initializes TIMER1, takes samples of the forward and reflected power, and processes them.

Now to details of the ISR, whose listing starts on page 6 of Appendix 1. The first task is to save the w and status registers. This must be done in such a way that none of the status bits are changed. I followed a method described in the PIC16F876 data sheet. After that, bit 2 of Port A

is cleared, which causes the two sample-and-hold chips to go into hold mode. A bit in the register Flags is then cleared. (Note that Flags is located in bank 1 of the data file registers, so bit rp0 of status must be set before it is accessed.)

The next step is to turn the TIMER1 counter off. We then initialize this counter and turn it back on. The 16-bit counter will overflow when it reaches the count $2^{16} = 65{,}536$. The counter is incremented 4 times every microsecond, so to set it to overflow every SamplePeriod (in μs, currently set to 90), we must place the following count in the register: 65,536 –4*SamplePeriod + Deadtime, where Deadtime is the dead time in TIMER1 counts between the time the ISR was initially entered and the time that TIMER1 is turned back on after being initialized. In this case, Deadtime = 16 counts.

Next, the BusyFlag bit is set in Flags, indicating that the routine is currently processing a measurement. We then wait 16 additional microseconds to allow time for the Sample-and-Hold chips to stabilize their held data. We then read in the contents of channel 0 of the ADC. Channel 0 is connected to the forward power channel. The ADC is immediately switched to read channel 1, but the actual read is delayed until the channel 1 electronics stabilize around the reflected power channel data. The forward power ADC data are transferred to the 16-bit register bForwardNew, and the new value is compared to the previous peak value, held in the 16-bit register bForward. If the new value is greater, it will replace the value in bForward.

Comparison of bForwardNew and bForward is done using the subtract operation. I found this procedure quite confusing at first, particularly the function of the carry bit, until I discovered a way of thinking about it that made the interpretation easy. To illustrate this, suppose we have two 8-bit quantities, *A* and *B*, in memory that we wish to compare. We can do this by subtracting *B* from *A*. Subtraction in the PIC processors is performed using 2's complement arithmetic. To subtract *B* from *A*, the 2's complement of *B* is first formed. I will denote the 2's complement as -*B*. To do this, every bit in *B* is interchanged, that is, 1's become 0's and 0's become 1's. It is easy to see that this is exactly equivalent to subtracting *B* from 255, that is, 255 – *B*. Then, 1 is added to the result. Thus,

$$-B = 256 - B$$

Once the 2's complement is formed, subtraction of B from A is accomplished by simple addition, that is

$$A - B = A + (-B) = A + (256 - B) = 256 + (A - B)$$

If the result of the addition is greater than 255, the carry bit will be set. On the other hand, if the result is less than or equal to 255, the carry bit will be cleared. Thus, we arrive at the rule:

If A ≥ B, the carry bit = 1

If A < B, the carry bit = 0

Now go back to the program listing (p. 7). Remember that the 10-bit result of an ADC conversion is left justified. Thus, the most significant 8 bits of the result are in bForwardNew and the least significant 2-bits are the left-most bits in bForwardNew+1. The first step is to compare the most significant 8 bits of the bForward and bForwardNew. If bForwardNew > bForward, the program jumps to the address ISR_012 where bForward is replaced by bForwardNew. If on the other hand, if bForward = bForwardNew, the bottom two bits are compared.

The program next reaches the instruction location ISR_019, where the subroutine GetForwardWatts is called. Remember that the signal from the forward channel is a voltage proportional to logarithmic power in units of dBm. The purpose of GetForwardWatts is to convert from dBm to watts. I use lookup tables to accomplish this conversion as rapidly as possible.

The program occupies much of page 0 of instruction memory and the Microchip floating-point package occupies much of page 1. Thus, pages 2 and 3 were available for the lookup tables, a total of 4096 values. A 10-bit ADC can output 1024 values. Thus, the forward and reflected power channels have 2048 possible inputs, which means that, at most, there are two instruction locations available to table an ADC value. Unfortunately, a 24-bit floating-point value requires three locations to encode a value. Thus, I developed a 16-bit floating point format that I used for this purpose. The next section describes this format.

16-bit Floating-Point Format

Since this format will be used to convert 10-bit binary integers, representing forward and reflected power in units of dBm, to 16-bit floating point values equal to the corresponding power in watts, we only need 10 bits of mantissa to retain accuracy. Furthermore, all of these values will be positive, so there is no need to have a sign bit. Finally, the range of values is limited, from about 1 μW (microwatt) to several thousand watts. Thus, this format consists of a 6-bit exponent, occupying the most significant 6 bits of the first byte, and a 10-bit "reduced"

mantissa. The mantissa is reduced by not including the leading 1 bit, which is present in all values except for 0. The exponent is offset by adding 27 to avoid negative values. Finally, an exponent of 0 is used to indicate that the number is 0.

For example, consider the number 1, which can be written in binary as $1.0000000000 \times 2^0$. The offset exponent is $27 + 0 = 27 = 011011$. The mantissa is just 0000000000 since the leading 1 is suppressed. Thus, the 16-bit floating-point value is 0110 1100 0000 0000 = 0x6C00.

As another example, take 439. In binary, this is $1.1011\ 0111\ 00 \times 2^8$. Thus, the exponent is 27 + 8 = 35 = 100011 and the mantissa is 10 1101 1100, so the floating-point values is 1000 1110 1101 1100 = 0x8EDC.

The largest and smallest numbers that be encoded in this format are approximately $1.37 \times 10^{11}$ and $1.49 \times 10^{-8}$, respectively.

Interrupt Service Routine (Continued)

The program continues at location ISR_019 by calling the subroutine GetForwardWatts to obtain the forward power in watts. This subroutine uses the Microchip standard method to implement lookup tables (see the discussion of lookup tables in the Macro section earlier in this report). The lookup tables for the forward power conversion occupy all of page 2 of memory and are organized into eight 256-word subtables. The first four subtables contain the right-hand bytes (low bytes) of the forward floating point power, and the next four subtables contain the left-hand bytes (high bytes). The least-significant 2 bits of the ADC result are used as an index into the four low-byte and the four high-byte subtables, and the most significant 8 bits are used as an index into the selected 256-word subtables. The result of this conversion is returned in the 16-bit data register FP.

After the lookup is complete, the processor initiates the actual ADC conversion of the reflected power signal. While this conversion is underway, the processor calls the subroutine SumP16, located in page 1 of instruction memory. This subroutine adds the forward power, in watts, to a running sum located in FPaccum. The program then checks to see if the ADC read of the reflected power is finished. (If it is not finished, the macro will wait until it is finished.) Once the read is finished, the processor sets the ADC to channel 0, checks to see if a new peak forward

power was measured and, if so, places the corresponding reflected power in the 16-bit data register bReflected.

Next, starting at the instruction following the address ISR_021, the 16-bit register, SampleCounter, is decremented by 1 and tested to see if it is 0. If it is 0, a bit named SampleDoneFlag is set in the bank 1 data register Flags. Then, at ISR_040, the status and w registers are restored to their state before the interrupt occured and program control is returned to the program location just prior to the interrupt. This return is accomplished using the retfie instruction, which returns to the appropriate location and also arms the interrupt system so that it will respond to future interrupts.

Routine at the End of a Cycle

The code immediately following the "call ISR" instruction on page 10 of Appendix 1 simply repetitively clears the bit BusyFlag in Flags and checks the SampleDoneFlag in Flags. When it finds this flag set, indicating that all samples for the current cycle have been taken, the program disarms the interrupt system with the instruction

```
bcf     INTCON,GIE
```

and proceeds to process the results obtained during the last cycle.

The first step in the process is to determine whether the peak forward power measured during the last cycle, and held in the 16-bit register bForward, is larger than the current peak power being displayed on the front panel LCD display, and held in the 16-bit register bForwardPeak. If the latest peak is not larger, control passes to location hh003 in the program. The register PeakWaitCounter is decremented by 1 and checked to see if it is 0. If it is not yet 0, nothing more is done, i.e., the previous peak power will continue to be displayed on the front panel LCD display. If, however, PeakWaitCounter has been decremented to the point that it is 0, or the forward peak measurement during the latest cycle is larger, the program will jump to the location hh001 where the value bForwardPeak will be replaced by the new peak value in bForward. This value is then converted to watts, using the subroutine GetForwardWatts, the resulting value is converted to a 24-bit floating-point value via the subroutine P16ToFP24, and saved in PeakFowardWatts. (It is necessary to convert to a 24-bit value so that the Microchip 24-bit floating-point library can be used for subsequent analyses.) Then, the corresponding reflected power will be converted to 16-bit floating-point format and then to 24-bit floating-

point format. Then, the subroutine CalculateSWRandSetMeter is called to determine the SWR from the 24-bit floating-point values of forward and reflected power and set the front panel meter to reflect this new value. Next, the new SWR and peak power values are written to the LCD display using the routine DisplayPeakPowerAndSWR. Finally, the 8-bit counter PeakWaitCounter is initialized to the value MaxNumPeakWaits.

If a new peak forward power was not found during the last cycle, the peak forward and corresponding reflected powers that were measured during this cycle are used to update the front-panel analog SWR meter. This code starts at location hh002.

The program next arrives at the location c002 (p. 12) where the CycleCounter is decremented and checked to see if it is 0. If it is not 0, control is passed to the location NextCycle where processing for a new cycle begins. If, on the other hand, CycleCounter = 0, the series of cycles is complete and the program moves to the processing that follows the completion of a series.

Routine Following Completion of a Series

The only processing that occurs at the end of a series involves the calculation and displaying of the average forward (or load) power. First, the three most significant bytes of the 32-bit floating point accumulator are moved to the register FP, converted to 24-bit floating-point Microchip format, and are multiplied by the inverse of the total number of measurements, contained in the data registers FPInvTotalSamplesShort and FPInvTotalSamplesLong for short and long averaging times, respectively. The resulting average forward (or load) power is written to the LCD display using the subroutine DisplayAvgPower. Then, the program transfers control to the instruction location StartSeries, which starts the next measurement series.

Calibrate Program

The next part of the program, starting at the bottom of page 12 of the Appendix, is assembled only if the constant Calibrate is given the value 1 at the beginning of the program. This function of this program is to write to the LCD display the average of 1024 ADC measurements of the signals entering the forward and reflected power input pins of the processor

In the remainder of this document, I will give notes about some of the subroutines used by my SWR/Power Meter program.

Subroutine CalculateSWRandSetMeter

Standing wave ratio (SWR) is defined as

$$\text{SWR} = V_{\max} / V_{\min} ,$$

where $V_{max}$ and $V_{min}$ are maximum and minimum voltages, respectively, present on the transmission line connecting the transmitter to load. Let $P_F$ and $P_R$ be the forward and reflected powers on the transmission line, respectively. The voltages, $V_F$ and $V_R$, of the forward and reflected waves, respectively, are related to $P_F$ and $P_R$ as follows:

$$V_F = \sqrt{P_F Z_0}$$
$$V_R = \sqrt{P_R Z_0}$$

where $Z_0$ is the characteristic impedance of the transmission line. The maximum voltage, $V_{max}$, will occur where the two waves are in phase, in which case $V_{max} = V_F + V_R$. Similarly, the minimum voltages on the line will occur where the forward and reflected waves are out of phase, so $V_{min} = V_F - V_R$. Thus,

$$\text{SWR} = \frac{V_F + V_R}{V_F - V_R} = \frac{\sqrt{P_F Z_0} + \sqrt{P_R Z_0}}{\sqrt{P_F Z_0} - \sqrt{P_R Z_0}}$$

$$= \frac{\sqrt{P_F} + \sqrt{P_R}}{\sqrt{P_F} - \sqrt{P_R}} = \frac{1 + \sqrt{P_R / P_F}}{1 - \sqrt{P_R / P_F}}$$

The subroutine, however, uses a different form of this formula to reduce the number of calculations. This modification is

$$\text{SWR} = \frac{-2}{\sqrt{P_R / P_F} - 1} - 1 .$$

To calculate the square root, I use a subroutine sqrt that I wrote. This subroutine was written to emphasize speed over accuracy. The accuracy (0.2% or better) is sufficient for our purposes. The routine begins by using a linear approximation to estimate the square root of a number between 1 and 4. (All other numbers can be scaled to values between 1 and 4 by multiplying or dividing by $2^N$ or where $N$ is an even integer.) This linear approximation is further improved as follows. Let $x$ be the number whose square root we wish to obtain, and let $s_1$ be the estimate obtained from the linear approximation. Our goal is to obtain a better estimate, $s_2$. I use the following formula to accomplish this:

$$s_2 = \frac{s_1 + x/s_1}{2}$$

Once the SWR is calculated, it is displayed on the front panel analog meter. As discussed earlier, this display is facilitated using a pulse-width-modulated output signal from the processor. The pulse width can be varied from 0 (i.e., the output signal is always 0 V) to 255 (the output signal is always high, at about 5 V). The circuitry driving the meter must be adjusted so that an output of 255 will cause a full scale deflection of the front panel analog meter.

Actually, the SWR is not directly displayed. Rather, the quantity 255 × (1 – 1/SWR) is displayed. I have found that this provides a superior meter indication to use when adjusting an antenna tuner for lowest SWR.

Subroutine DisplayPeakPowerAndSWR

This routine writes the peak forward or load power (depending on the setting of the front panel power switch shown in Figure 1) to the LCD display. First, the routine determines if the power is large enough to write to the screen. If not, it writes the word “low”. It next determines if the forward or load power is to be written. If needed, load power, $P_L$, is calculated from the formula

$$P_L = \left(1 - |\rho|^2\right) P_F ,$$

where $|\rho|$ is the reflection coefficient and $P_F$ is the measured peak forward power. $|\rho| = \sqrt{P_R/P_F}$ is calculated during the SWR calculation and its square is stored in the data register LoadFactor. Once the peak power is calculated, the subroutine FormatWatts is called. This

routine decides whether the power is to be displayed in units of watts (W) or milliwatts (mW). The power is next displayed with the appropriate units.

Finally, this subroutine formats the SWR for LCD display, by calling the subroutine FormatSWR, and then displays the formatted SWR.

Subroutine DisplayAvgPower

This subroutine first checks to make sure the peak power was large enough to display. If so, it then calculates the load power if needed, calls FormatWatts to format the power for display, and then writes the average forward (or load) power to the front panel LCD display.

Subroutine LCD_DisplayASCII_Num

This subroutine writes a number to the LCD display. The decimal digits (in ASCII format) of the number to be written must already be in the data registers ASCII_Num through ASCII_Num+4. The sign of the number (i.e., positive or negative) must be indicated by the presence of 0 (for positive) or 1 in the register ASCII_Sign. The location of the decimal point is contained in ASCII_Decimal: The number in this register gives the number of digits to the right of the decimal point. Finally, ASCII_MinField gives the minimum number of spaces to allocate to the number.

Subroutine Int16ToASCII_Num

This subroutine converts a 16-bit integer, initially located in aargb0 and aargb1, to ASCII form. The result is placed in the registers ASCII_Num through ASCII_Num+4.

The largest number that can be contained in 16 binary bits is $2^{16} - 1 = 65{,}535$. The routine thus first repetitively subtracts 1000 from the integer to be converted until the result goes negative. The total number of subtractions before the remainder went negative is the first decimal digit. It is converted to ASCII form by adding 0x30. Next, 100 is repetitively subtracted from the remainder, thus determining the second digit. This process continues until all five decimal digits are converted.

Subroutine GetForwardWatts

As discussed previously, this subroutine is called with a 10-bit value from the forward power ADC and returns with the corresponding forward power in watts. A system of lookup tables is used to accomplish this task.

The listing for this program starts on page 27 of Appendix 1. The first instruction sets rp0 in status so that bank 1 of data memory can be accessed. The program then calls _GetForwardLowByte. The purpose of this program is to obtain the low (least significant) byte of the forward power. The subroutine is also on page 27. It first sets bits in PCLATH to access page 2 of program memory, where the forward power lookup tables are located. It then checks bits 6 and 7 of bForwardNew+1. bForwardNew holds the 10-bit digitized value of forward power. The most significant 8 bits are located in bForwardNew and the least significant 2 bits are bits 7 and 6 of bForwardNew+1. Depending on the contents of these latter two bits, bits in PCLATH are set appropriately so one of four 256-word-long lookup tables can be accessed. Finally, the contents of bForwardNew (i.e., the most significant 8 bits) are loaded into PCL, the lower byte of the program counter. As discussed earlier, the effect of this final operation is to cause the processor to next execute the instruction pointed to by PCLATH and PCL. The lookup tables begin on page 51 of Appendix 1 and consist of nothing more than a whole string of retlw instructions. The value of PCLATH determines which of four tables will be accessed, and the value of PCL will determine which retlw instruction within the table will be executed. Whichever retlw is executed will cause the processor to return to the instruction following the last subroutine call with the value of the low byte of the 16-bit floating-point power in watts loaded in the w register.

Finally, let me describe how I construct these tables. As mentioned earlier, the response of the forward (or reflected) power channel can be characterized by the linear relation

$$dBm = \alpha \times ADC + \beta$$

where *dBm* is the power input to the forward channel, measured in units of dBm and ADC is the resulting digitized value of this power. The constants, $\alpha$ and $\beta$, must be determined experimentally. The power in units of dBm can be converted to power, *P*, in units of watts using the following formula:

$$P = (0.001\,\mathrm{W}) \times 10^{dBm/10}\,.$$

Combining these formulae:

$$P = (0.001\,\mathrm{W}) \times 10^{(\alpha \times ADC + \beta)/10} .$$

The lookup tables included in Appendix 1 are the ones I used for my power meter based on the calibration data acquired with it. Unfortunately, they are not terribly accurate because I do not have a good way to accurately measure rf power in watts. I estimate their uncertainty to be about ±10%. One of my goals is to acquire the equipment to more accurately measure rf power. Consequently, it would be good if anyone who duplicates my power meter would do their own calibration. This will, of course, require that the lookup tables in Appendix 1 be changed to match the new calibration data.

The best way to do the calibration would be feed known powers through the directional coupler and record the resulting ADC readings. (Doing this way calibrates the directional coupler, the 20-dB attenuators, and the AD8307 chips.) Then, the powers in watts should be converted to units of dBm. Next a straight line should be fit to the dBm data as a function of the ADC data. If more than two power points are obtained, this line could be fit using the least-squares method. The slope and intercept of the resulting line are the constants $\alpha$ and $\beta$, respectively, defined above.

The new forward-power and reflected-power lookup tables can then be constructed by hand using these formulae. This would, however, be very tedious. As an alternative, I wrote a computer program to do this. This program, written in Microsoft Visual Basic 6.0, enables the user to enter values for $\alpha$ and $\beta$ (named the "slope" and "intercept" in the program), tell the program whether the forward or reflected power channel lookup tables are being constructed, and then construct them. The tables are shown in a box. These tables can then be saved to a file. The file can be later opened using a text editor such as Notepad, the contents copied to the clipboard, and then pasted into the assembly language listing.

The lookup table program is a Windows program and is installed in the same way normal windows programs are installed. There are three files involved in this process: Power Tables.Cab, setup.exe, and SETUP.LST. They are included on the ARRL website. Follow this procedure to install this program on your computer:

1. Download the three files listed in the preceding paragraph from the ARRL website into a convenient temporary file.

2. Using Windows Explorer (or equivalent) create a subdirectory to the Program Files directory on your main hard drive named “PowerMeterLookup Tables”.
3. Start (double click) setup.exe
4. Follow the onscreen prompts. At one point, the installation program will ask where you want to place the program files. Using the Browse button, select \Program Files\PowerMeterLookupTables.
5. Finish the installation process.
6. To run the program, hit the Start button, select programs, then select the “Power Meter Lookup Tables” folder, and double click on the program “Power Meter Lookup Tables”. On my computer, the first time you do this, the system will ask you to help find the program. Hit the “Browse” button, and select the program directory where you placed the files. Apparently, you only have to do this once.

# Appendix 1. List of Power/SWR Meter Software

```
;**POWER METER, Hardware Version 2, Software Version 2*****************************************

    include  <p16f876a.inc>
;*******************Constants**********************************

; To facilitate simulation, use the Simulate variable. It has the following
; values:
;
;       Value      Function
;         0        No simulation
;         1        Simulate

Simulate         equ 0

ADRES_STEP       equ     0

; Switch to determine whether software runs in calibrate (Calibrate = 1) or normal (Calibrate = 0) mode.
; In calibrate mode, the processor will read the forward and reflected power channels

Calibrate        equ     0


; The following constants are only used if Simulate = 1

ForwardOrLoadWatts            equ 1   ; Set 1 for Forward, 0 for Load Watts
LongOrShortAvgTime equ    1   ; Set 1 for long averaging time


; This constant defines the binary low-power threshold, where the display shows
; "Low power".
LowPowerThreshold equ     404/4   ; Corresponds to forward powers < about 1 mW


; Time (microseconds) between consecutive samples
SamplePeriod     equ 90
TMR1_Init        equ     65536 - 5*SamplePeriod + 16



 if Calibrate == 1
NumSamples               equ     1024    ; This value must not be changed.
 else
    if Simulate > 0
MaxNumPeakWaits      equ 30  ; Number of waits in peak mode
MaxNumSWRMeterWaits equ   16  ; Number of consecutive low power readings to set SWR meter to 0
NumSamples               equ 100000/SamplePeriod  ; Number of average-mode samples in one cycle
NumCyclesShort       equ 16  ; Number of cycles in a series for "short" averaging
NumCyclesLong        equ 48  ; Number of cycles in a series for "long" averaging
    else
MaxNumPeakWaits      equ 30  ; Number of waits in peak mode
MaxNumSWRMeterWaits equ   16  ; Number of consecutive low power readings to set SWR meter to 0
NumSamples               equ 100000/SamplePeriod  ; Number of average-mode samples in one cycle
NumCyclesShort       equ 16  ; Number of cycles in a series for "short" averaging
NumCyclesLong        equ 48  ; Number of cycles in a series for "long" averaging
    endif
 endif



;**********Packed 16 bit floating point values*********************************

; This routine uses a packed 16-bit floating point value that I developed to reduce
; memory usage in a look-up table relating power in watts to measured adc values for
; forward and reflected powers.
;
; The packed values contain a 10-bit "reduced" mantissa (m) and a 6 bit shifted exponent (s).
; All values are assumed positive so there is no need for a sign bit. Also, the first bit
; in the mantissa, which is always 1 except when the number id 0, is suppressed.
; The offset value for the exponent is 27. A shifted exponent of 0 means the number is 0.
;
; For example, the number 0 has a packed 16 bit value of 0x0000.

; Next, consider the number 1. This can be written as 1.0 * 2^0. The reduced mantissa
; is 0 and the shifted exponent is 27 + 0 = 27 = 0x2B = 0b011011. Thus, the packed 16-bit
; value is 0b0110110000000000 = 0x6C00
```

```
;
;  Next take the number 1107 = 0x453. In binary, this can be written
;  10001010011 = 1.0001010011 * 2^10. The reduced mantissa is 00 0101 0011 and the exponent
;  is 27 + 10 = 37 = 0x25 = 100101. Thus the packed value is 1001 0100 0101 0011
; = 0x9453
;
; The largest value that can be represented is in binary 1.11 1111 1111 & 2^(63 - 27)
; = 111 1111 1111 * 2^26 = 0x7FF * 2^26 = 137,371,844,608 = 1.37371844608E+11
; and the smallest value is 1.00 0000 0000 * 2^(1 - 27) = 1 * 2^-26 = 1.49011611938E-8.

; This 16-bit format is used only in the interrupt service routine and for looking up
; powers give a particular ADC reading. All other arithmetic is done using a 24-bit format
; developed by Microchip.







;*****************CONSTANTS & MEMORY ALLOCATIONS*****************************
;
;  First make general register assignments

    cblock       0x20
; Next group of variables used by MicroChip floating point routines.
      exp:0, aexp:0
      aarg,aargb0, aargb1, aargb2
      aargb3, aargb4, aargb7
      sign
      fpflags
      bexp:0, barg,bargb0,bargb1,bargb2

; Following variables used by my floating point routines and by
; Watts lookup table routines

      isign, intPart
      temp:0, tempb0, tempb1
      t1:3, t2:3, t3:3


; Variables for construction of and display of ASCII numbers
      ASCII_num:5, ASCII_Sign, ASCII_Decimal, ASCII_MinField

; More temporaray variables
      Timer, Timer2, Timer3
     LCDTemp, LCDChar, LCDTextPtr

; Floating point variables for power meter data
      SWR:3, PeakForwardWatts:3


; This variable denotes various statuses associated with power levels
;        Bit  Value  Means
;        ---------------------------------------------
;        0      0     PeakForwardWatts >= LowPowerThreshold
;        0      1     Peak ForwardWatts <  LowPowerThreshold
;
;        2      0     Display Mode = Forward - Reflected Power (Watts)
;        2      1     Display Mode = Forward Power (Watts)
;
;        4      0     Short averaging time
;        4      1     Long averaging time
;
;        5      0     Displays power in units of W
;        5      1     Displays power in units of mW

      ModeStatusFlags

; Register to count # of cycles.
    CycleCounter

; This variable holds the total number of samples used to calculate average power
      FPInvTotalSamplesShort:3
      FPInvTotalSamplesLong:3

; Counter for number of waits, after peak is found
      PeakWaitCounter

; Counter for number of waits, while power is low, before SWR meter set to 0
```

```
      SWRMeterWaitCounter

; More temporary variables
    temp1, temp2, temp3, temp4, TableTemp

; Reflection Coefficient squared
    LoadFactor:3

; Debug Registers
    Debug1,Debug2

    endc


; The definitions used to access bits in ModeStatusFlags

LowPeakWatts     equ 0
ForwardWatts     equ 2
LongAvgTime          equ 4
MilliWattsFlag   equ 5


; Temporary storage for critical variables during interrupt service

    cblock       0x7D

    w_temp, status_temp, PCLATH_temp

    endc


; Variables located in Block 1

    cblock       0xA0

; Accumulator for forward power running sum, and FP packed value.
    Ch0Avg:0, FPaccum:4, Ch1Avg:0, FP:4
    B1Temp1, B1Temp2, B1Temp3


; Variables used to hold digital valuesl produced by ADC
    Ch0Max:0, bForward:2
    Ch0Min:0, bForwardPeak:2
    Ch1Max:0, bForwardNew:2
    Ch1Min:0, bReflected:2

; Status flags
    Flags

; Loop counter for average and peak modes
      SampleCounter:2

; Temporary holders for ADC values
    Ch0Value:2, Ch1Value:2

; Temp registers used only in simulate mode
    S_ADRESL, S_ADRESH

; Bank1 debug registers
    B1_Debug1, B1_Debug2

    endc

; Definition of bits of Flags

NewPeak equ      0
SampleDoneFlag   equ 1
LZflag  equ      2
BusyFlag         equ 3

; Constants used for selecting pages

psel0            equ     3
psel1            equ     4


bit0 equ 0x0
bit1 equ 0x1
bit2 equ 0x2
bit3 equ 0x3
bit4 equ 0x4
```

```
bit5 equ 0x5
bit6 equ 0x6
bit7 equ 0x7

B0   equ 0
B1   equ 1
B2   equ 2
B3   equ 3
B4   equ 4
B5   equ 5
B6   equ 6
B7   equ 7

MSB  equ 7
LSB  equ 0

    #define      _C       STATUS,0
    #define      _Z       STATUS,2

;
;       FLOATING POINT SPECIFIC DEFINITIONS
;
;       literal constants
;
EXPBIAS         equ     D'127'
;
;       floating point library exception flags
;

IOV equ     0       ; bit0 = integer overflow flag
FOV     equ     1       ; bit1 = floating point overflow flag
FUN     equ     2       ; bit2 = floating point underflow flag
FDZ     equ     3       ; bit3 = floating point divide by zero flag
NAN equ 4       ; bit4 = not-a-number exception flag
DOM equ 5        ; bit5 = domain error exception flag
RND     equ     6       ; bit6 = floating point rounding flag, 0 = truncation
SAT     equ     7       ; bit7 = floating point saturate flag, 0 = terminate on
                                ; exception without saturation, 1 = terminate on
                                ; exception with saturation to appropriate value

;*************MACRO DEFINITIONS***********************************


MoveBytes  macro  AA, BB, nbytes      ; Macro that moves nbytes bytes from

    if (0x80 & AA) == (0x80 & BB)
       if (0x80 & AA) != 0
                bsf     status,rp0
       else
                bcf     status,rp0
       endif
    endif

     local i = 0          ; a through a + nbytes - 1 to
     while i < nbytes     ; b through b + nbytes - 1
       if (0x80 & AA) != (0x80 & BB)
                if (0x80 & AA) != 0
                    bsf     status,rp0
                else
                    bcf     status,rp0
                endif
       endif

       movf     0x7F & (AA + i)),w

       if (0x80 & AA) != (0x80 & BB)
                if (0x80 & BB) != 0
                    bsf     status,rp0
                else
                    bcf     status,rp0
                endif
       endif

       movwf    (0x7f & (BB + i))
i        set    i + 1

      endw
    endm

; This macro stores a 24-bit floating point literal in three consecutive
; memory locations starting at Address in Bank 0.
```

```
StoreFPLiteral macro  Literal,Address
    movlw        (0xff0000 & Literal) >> 16    ; Get top 8 bits
    movwf        Address
    movlw        (0xff00 & Literal) >> 8       ; Get middle 8 bits
    movwf        Address + 1
    movlw        (0xff & Literal)     ; Get lower 8 bits
    movwf        Address + 2
    endm


;---------LCD_DisplayText-----------------------------------------
; This macro sends text to the LCD display, starting at the current
; cursor position. The first argument of the macro is the text to be displayed. The
; second is a unique address which is used internally within the routine.
;
; For example, suppose one wishes to output the string "Test output" to the
; LCD display, starting at the current cursor position.
; The following code would do this:
;
;   LCD_DisplayText  "Test output",Adr8898



    if Simulate > 0
LCD_DisplayText  macro  TextString, UniqueAddress
    endm

    else
LCD_DisplayText  macro  TextString, UniqueAddress
    bcf          status,rp0               ; Select bank 0
    bcf          PCLATH,psel0         ; Select page 0
    bcf          PCLATH,psel1

    clrf         LCDTextPtr               ; Start with first character in string, i.e., with 0 offset

    movf         LCDTextPtr,w     ; Load offset
    call         ($+7)        ; Get next character from table
    andlw        0xff         ; Test if character is 0
    btfsc        status,z     ; Skip if not zero
    goto         UniqueAddress            ; Finished-go to end
    call         LCD_SendChar
    incf         LCDTextPtr,f     ; Increment to point to next character
    goto         $-7          ; Do next character

; The next 8 instruction and the lookup table that follows it comprise a subroutine.

    movwf        TableTemp        ;Store offset temporarily
    movlw        HIGH ($+7)       ;High part of address of table
    movwf        PCLATH           ;stored in PCLATH
    movf         TableTemp,w      ;Load offset back into w
    addlw        LOW ($+4)        ;Offset into table
    btfsc        status,c     ;Carry on add?
    incf         PCLATH,f     ;Yes
    movwf        PCL          ;Enter address of table entry
    dt  TextString,0      ;Defines table
UniqueAddress
    endm
    endif


; Macros for ADC use

; Initialize the ADC unit.
ADC_Initialize   macro
    bsf          status,rp0
    movlw        0x05     ; Select AN0, AN1, Vref as analog inputs, left justified result
    movwf        0x7f & ADCON1
    bcf          status,rp0

    movlw        0x81     ; Set up for AN0 and turn ADC on
    movwf        ADCON0

    bcf          PCLATH,psel0 ; Select Page 0
    bcf          PCLATH,psel1

    movlw        7        ; Wait 7 microseconds for system
    call         WaitMicroSeconds ; to stabilize
    endm
```

```
ADC_StartRead    macro
        bsf          ADCON0,2
 if Simulate==1
    movlw        LOW(ADRES_STEP)                          ; Subtract ADRES_STEP from S_ADRES-
    bsf          status,rp0
    subwf        0x7F & S_ADRESL,f
    btfss        status,c
    decf         0x7F & S_ADRESH,f
    movlw        HIGH(ADRES_STEP)
    subwf        0x7F & S_ADRESH,f
    bcf          status,rp0
 endif
    endm

ADC_ReadFinished macro
    if Simulate==0
       btfsc     ADCON0,2
       goto      $-1
    else
       bsf           status,rp0
       movf      0x7F & S_ADRESL,w
       movwf     0x7F & ADRESL
       movf      0x7F & S_ADRESH,w
       bcf           status,rp0
       movwf     ADRESH
    endif
    endm

ADC_SetCh0       macro
    bcf          ADCON0,3
 endm

ADC_SetCh1       macro
    bsf          ADCON0,3
 endm


 if Calibrate == 0
;********* CODE FOR NORMAL MODE ************************************************************

    org          0x0
    goto         StartUp



; This interrupt driven routine is the inner sampling loop.

    org          0x04
ISR
    movwf        w_temp
    swapf        status,w
    movwf        status_temp

    bcf          status,rp0
    bcf          PORTA,2                  ; S/H converters -> hold
    bsf          status,rp0
    bcf          0x7F & Flags, NewPeak
    bcf          status,rp0

    bcf          T1CON,TMR1ON             ; Turn Timer1 off and initialize it

    movlw        LOW(TMR1_Init)
    movwf        TMR1L
    movlw        HIGH(TMR1_Init)
    movwf        TMR1H
    bsf          T1CON,TMR1ON         ; Turn Timer 1 on

    bcf          PIR1,TMR1IF              ; Turn off Timer1 interrupt flag

    bsf          status,rp0
    btfsc        0x7F & Flags,BusyFlag    ; Are we already processing a measurement? If so, do not start another one

    goto         ISR_040

    bsf          0x7F & Flags,BusyFlag    ; Set busy flag, indicating a new measurement is being processed.
    bcf          status,rp0

    movlw        16
    call         WaitMicroSeconds     ; Wait a total of 12 microseconds for S/H to settle of held value

    ADC_StartRead
```

```
    ADC_ReadFinished
    ADC_SetCh1                      ; Set ADC to Channel 1 to read reflected power.

    movf        ADRESH,w            ; Transfer result of ADC to bForwardNew
    bsf         status,rp0
    movwf       0x7F & bForwardNew
    movf        0x7F & ADRESL,w
    movwf       0x7F & bForwardNew + 1



; Is the newly measured value of Forward power greater than the previous maximum?
    movf        0x7F & bForwardNew,w
    subwf       0x7F & bForward,w   ; 256 + (bForward - bForwardNew): c = 0 means bForwardNew > bForward
    btfss       status,c
    goto        ISR_012                     ; bForwardNew < bForward, so new maximum found
    btfss       status,z
    goto        ISR_019                     ; bForward < bForwardNew so no new maximum

    movf        0x7F & bForwardNew+1,w          ; MSB's equal so check LSB's
    subwf       0x7F & bForward+1,w     ; 256 + ((bForward+1) - (bForwardNew+1))
    btfsc       status,c
    goto        ISR_019                 ; (bForwardNew+1) > (bForward+1)

ISR_012                                     ; New maximum found so replace bForward with bForwardNew
    movf        0x7F & bForwardNew,w
    movwf       0x7F & bForward
    movf        0x7F & bForwardNew+1,w
    movwf       0x7F & bForward+1
    bsf         0x7F & Flags,NewPeak    ; Set flag for latter use

ISR_019
    call        GetForwardWatts
    bcf         status,rp0

    ADC_StartRead                   ; Start Read of reflected power

    bcf         PCLATH,psel1        ; Select Page 1
    bsf         PCLATH,psel0

    call        0x7FF & SumP16          ; Update accumlated value
    bcf         PCLATH,psel0

    bcf         status,rp0

    ADC_ReadFinished

    bsf         PORTA,2                 ; S/H to sample mode
    ADC_SetCh0                      ; ADC to Channel 0 (Forward power)
    bsf         status, rp0
    btfss       (0x7F & Flags),NewPeak      ; Was a new maximum forward power measured?
    goto        ISR_021
    bcf         status,rp0
    movf        ADRESH,w                ; Yes, so update bReflected with new value
    bsf         status,rp0
    movwf       0x7F & bReflected
    movf        0x7F & ADRESL,w
    movwf       0x7F & bReflected+1

ISR_021                                     ; Decrement two-byte sample counter
    decfsz      0x7F & SampleCounter+1,f  ; Decrement LSByte
    goto        ISR_031                     ; Byte not zero
    bsf         0x7F & Flags,LZflag
    movf        0x7F & SampleCounter,f
    btfsc       status,z
    bsf         0x7F & Flags,SampleDoneFlag
    goto        ISR_040

ISR_031
    btfss       0x7F & Flags,LZflag
    goto        ISR_040
    decf        0x7F & SampleCounter,f
    bcf         0x7F & Flags,LZflag

ISR_040
    swapf       status_temp,w
    movwf       status
    swapf       w_temp,f
    swapf       w_temp,w
    retfie
```

```
StartUp

; Initialize I/O ports
    bsf         status,rp0
    movlw       0xFB    ; PORTA: all 5 bits inputs except for bit 2
    movwf       0x7F & TRISA

    movlw       0x00    ; PORTB: all 8 bits inputs normally
    movwf       0x7F & TRISB

    movlw       0x8B    ; PORTC: Bits 0, 1 unused. Bit 2 CCP1 output. Bit 3 S1 input. Bits 4-6 LCD control pins (outputs)
    movwf       0x7F & TRISC     ; Bit 7 S2 input

; Initialize LCD module and display introductory screen
    call        LCD_Init
    LCD_DisplayText  " W7IEQ Power Meter", uAdr10
    movlw       Line2 + 1
    call LCD_SetLineAndColumn
    LCD_DisplayText "     Version 2", uAdr237
    movlw 30
    call  WaitTenthSeconds


; Initialize the ADC unit.
    ADC_Initialize

    bcf         status,rp0
    bsf PORTA,2          ; Put S/H chips into sample mode

    call        PWM_Initialize           ; Initialize pulse width modulated output

; Setup Timer 1 interrupts for sample control

    bsf         status,rp0
    bsf         0x7F & PIE1,TMR1IE           ; Arm Timer 1 interrupt
    bsf         INTCON,PEIE

; Initialize floating point routines by selecting "rounding" and "saturation"

    bcf         status,rp0
    movlw       0xC0
    movwf       fpflags


; Convert NumSamples*NumCyclesShort to floating point, and invert

    movlw       LOW(NumSamples)
    movwf       aargb1
    movlw       HIGH(NumSamples)
    movwf       aargb0
    bsf         PCLATH,psel0
    call        0x7ff & FLO1624
    MoveBytes   aarg,t1,3
    MoveBytes aarg,barg,3
    movlw       NumCyclesShort
    movwf       aargb1
    clrf        aargb0
    call        0x7ff & FLO1624
    call        0x7ff & FPM24
    MoveBytes aarg,barg,3
    movlw       0x7f
    movwf       aarg
    clrf        aarg+1
    clrf        aarg+2
    call        0x7ff & FPD24
    MoveBytes aarg,FPInvTotalSamplesShort,3

; Convert NumSamples*NumCyclesLong to floating point, and invert

    movlw       NumCyclesLong
    movwf       aargb1
    clrf        aargb0
    call        0x7ff & FLO1624
    MoveBytes   t1,barg,3
    call        0x7ff & FPM24
    MoveBytes aarg,barg,3
    movlw       0x7f
    movwf       aarg
    clrf        aarg+1
```

```
    clrf         aarg+2
    call         0x7ff & FPD24
    MoveBytes aarg,FPInvTotalSamplesLong,3
    bcf          PCLATH,psel0


; Initialize for normal operation

    clrf         ModeStatusFlags

 if Simulate==1
; Initialize the simulated ADCRES registers
    bsf          status,rp0
    movlw        0x40
    movwf        0x7F & S_ADRESL
    movlw        0x97
    movwf        0x7F & S_ADRESH
    bcf          status,rp0
 endif


;Initialize for Peak Hold mode

    bsf          status,rp0
    clrf         0x7F & bForwardPeak       ; Start with 0 watts
    clrf         0x7F & bForwardPeak+1
    bcf          status,rp0
    movlw        MaxNumPeakWaits
    movwf        PeakWaitCounter

; Initialize Meter wait counter

    movlw        MaxNumSWRMeterWaits
    movwf        SWRMeterWaitCounter

;Initialize the display.
    bcf          PCLATH,psel0
    call         LCD_ClearDisplay
    LCD_DisplayText  "                SWR", uAdr397

StartSeries
    bcf          PCLATH,psel0
    bcf          status,rp0
; Determine display mode: Forward or Load Power.

 if Simulate == 0
    bcf ModeStatusFlags,ForwardWatts
    btfsc        PORTC,3          ; Test setting of mode switch
    bsf ModeStatusFlags,ForwardWatts
  else
    if ForwardOrLoadWatts == 0
       bcf       ModeStatusFlags,ForwardWatts
    else
       bsf       ModeStatusFlags,ForwardWatts
    endif
  endif

; Finish Initializing the Display
    btfss        ModeStatusFlags,ForwardWatts
    goto         Main_401
    movlw        Line1+1
    call         LCD_SetLineAndColumn
    LCD_DisplayText  "PEP-F=",uadr8133
    movlw        Line2+1
    call         LCD_SetLineAndColumn
    LCD_DisplayText  "AEP-F=",uadr8134
    goto         Main_402
Main_401
    movlw        Line1+1
    call         LCD_SetLineAndColumn
    LCD_DisplayText  "PEP-L=",uadr8135
    movlw        Line2+1
    call         LCD_SetLineAndColumn
    LCD_DisplayText  "AEP-L=",uadr8136
Main_402


 if Simulate==0
    btfss        PORTC,7          ; Is Averaging time switch set?
    goto         $+4
    movlw        NumCyclesLong
```

```
    bsf          ModeStatusFlags,LongAvgTime
    goto         $+3
    movlw        NumCyclesShort
    bcf ModeStatusFlags,LongAvgTime
 else
   if LongOrShortAvgTime == 1
      movlw      NumCyclesLong
      bsf        ModeStatusFlags,LongAvgTime
   else
      movlw      NumCyclesShort
      bcf        ModeStatusFlags,LongAvgTime
   endif
 endif

    movwf        CycleCounter

; Clear the forward power accumulator

    bsf          status,rp0
    clrf         0x7F & FPaccum
    clrf         0x7F & FPaccum + 1
    clrf         0x7F & FPaccum + 2
    clrf         0x7F & FPaccum + 3



NextCycle
    bsf          status,rp0

    movlw        HIGH(NumSamples)      ; Initialize inner loop counter
    movwf        0x7F & SampleCounter
    movlw        LOW(NumSamples)
    movwf        0x7F & SampleCounter+1

    clrf         0x7F & Flags
 if LOW(NumSamples) == 0
    bsf          0x7F & Flags,LZflag
 endif

    clrf         0x7F & bForward
    clrf         0x7F & bForward+1


; Start a sampling cycle

    call         ISR                          ; Call interrupt routine to start sampling loop

; Now loop until SampleCounter has been decremented to 0
    bsf          status,rp0

ZeroCheck
    bcf          0x7F & Flags,BusyFlag
    btfss        0x7F & Flags,SampleDoneFlag
    goto         $-2


; Sample counter is 0. Disable interrupt and continue with
; processing at end of one cycle.

    bcf          INTCON,GIE


; Is this peak greater than the previous peak?

    movf         0x7F & bForward,w
    subwf        0x7F & bForwardPeak,w ;256 + (bForwardPeak - bForward): c = 0 means new peak
    btfss        status,c
    goto         hh001                ; New peak value found
    btfss        status,z         ; Are MSB's equal. If so, check lower bytes
    goto         hh003                ; Not a new peak
    movf         0x7F & bForward+1,w
    subwf        0x7F & bForwardPeak+1,w   ; 256 + ((bForwardPeak+1) - (bForward+1)): c = 0 means new peak
    btfss        status,c
    goto         hh001                ; New peak value found


; There is no new peak value. Keep the previous peak for a maximum
; of MaxNumPeakWaits sampling cycles.
hh003
    bcf          status,rp0
    decfsz       PeakWaitCounter,f
```

```
    goto        hh002       ; Keep previous peak but update SWR meter

; Come here if a new peak has been found, or the old peak value has existed for
; MaxNumPeakWaits sampling cycles.

hh001
    MoveBytes   bForward,bForwardPeak,2
    MoveBytes   bForward,bForwardNew,2

    call        GetForwardWatts

    bcf         PCLATH,psel1            ; Set to page 1
    bsf         PCLATH,psel0
    clrf        0x7F & FP+2
    call        0x7FF & P16ToFP24           ; Convert P16 forward power to FP24 format in aarg

    MoveBytes aarg,PeakForwardWatts,3

    bcf         PCLATH,psel0

    call        GetReflectedWatts
    clrf        0x7F & FP+2
    bcf         PCLATH,psel1
    call        0x7FF & P16ToFP24

    MoveBytes   PeakForwardWatts,barg,3
    bcf         PCLATH,psel0

    call        CalculateSWRandSetMeter
    bcf         PCLATH,psel0
    call        DisplayPeakPowerAndSWR

    movlw       MaxNumPeakWaits
    movwf       PeakWaitCounter
    goto        c002

; Come here if not a new peak.

hh002
    MoveBytes   bForward,bForwardNew,2

    call        GetForwardWatts
    clrf        0x7F & FP+2
    bcf         PCLATH,psel1            ; Set to page 1
    bsf         PCLATH,psel0

    call        0x7FF & P16ToFP24           ; Convert P16 forward power to FP24 format in aarg

    MoveBytes aarg,PeakForwardWatts,3

    bcf         PCLATH,psel0

    call        GetReflectedWatts
    clrf        0x7F & FP+2
    bcf         PCLATH,psel1
    call        0x7FF & P16ToFP24

    MoveBytes   PeakForwardWatts,barg,3
    bcf         PCLATH,psel0

    call        CalculateSWRandSetMeter

c002

    decfsz      CycleCounter,f
    goto        NextCycle

    MoveBytes   FPaccum,FP,3
    bsf         PCLATH,psel0
    call        0x7FF & P16ToFP24
    bcf         PCLATH,psel0

    btfss       ModeStatusFlags,LongAvgTime     ; Is averaging time long?
    goto        cc031               ; No
    MoveBytes FPInvTotalSamplesLong,barg,3
    goto        cc032
cc031
    MoveBytes FPInvTotalSamplesShort,barg,3
cc032
    bsf         PCLATH,psel0
    call        0x7ff & FPM24
```

```
    bcf          PCLATH,psel0
    call         DisplayAvgPower


    goto         StartSeries

 endif


 if Calibrate == 1

;************* CODE for CALIBRATE MODE ***************************************************************

    org 0x0
    goto         StartUp


    org 0x5
CalMode
StartUp

    bsf          status,rp0
    bcf          0x7F & TRISA,2
    bcf          status,rp0

    movlw        10
    call WaitTenthSeconds

; General initialization

; Initialize I/O ports
    bsf          status,rp0
    movlw        0xFB     ; PORTA: all 5 bits inputs except for bit 2
    movwf        0x7F & TRISA

    movlw        0x00     ; PORTB: all 8 bits outputs normally
    movwf        0x7F & TRISB

    movlw        0x8B     ; PORTC: Bits 0, 1 unused. Bit 2 CCP1 output. Bit 3 S1 input. Bits 4-6 LCD control pins (outputs)
    movwf        0x7F & TRISC     ; Bit 7 S2 input
    bcf          status,rp0

; Initialize LCD module and display introductory screen
    call         LCD_Init

    LCD_DisplayText  " W7IEQ Power Meter", uAdr10
    movlw        Line2 + 1
    call LCD_SetLineAndColumn
    LCD_DisplayText " Calibrate Mode", uAdr237
    movlw 50
    call  WaitTenthSeconds


; Initialize the ADC unit.
    ADC_Initialize

    bcf          status,rp0
    bsf PORTA,2           ; Put S/H chips into sample mode

; Initialize for normal operation
    bsf          status,rp0
    clrf         0x7F & Flags

;Initialize the display.
    call         LCD_ClearDisplay
    LCD_DisplayText  "CH0=", uAdr397
    movlw        Line2 + 1
    call LCD_SetLineAndColumn
    LCD_DisplayText  "CH1=", uAdr137


; Initialize the simulated ADRES registers
    bsf          status,rp0
    clrf         0x7F & S_ADRESL
    clrf         0x7F & S_ADRESH

; Initialize the PWM unit and set duty cycle to 100%

    bcf          status,rp0
    call         PWM_Initialize
    movlw        255
```

```
    call        PWM_SetDutyCycle


StartSeries
    bcf         PCLATH,psel0
    bsf         status,rp0

    movlw       HIGH(NumSamples)
    movwf       0x7F & SampleCounter
    movlw       LOW(NumSamples)
    movwf       0x7F & SampleCounter+1


; Clear the accumulators for Channel 1 and Channel 2


    clrf        0x7F & Ch0Avg
    clrf        0x7F & Ch0Avg + 1
    clrf        0x7F & Ch0Avg + 2
    clrf        0x7F & Ch0Avg + 3
    clrf        0x7F & Ch1Avg
    clrf        0x7F & Ch1Avg + 1
    clrf        0x7F & Ch1Avg + 2
    clrf        0x7F & Ch1Avg + 2
    clrf        0x7F & Ch1Avg + 3
    clrf        0x7F & Ch0Max
    clrf        0x7F & Ch0Max+1
    clrf        0x7F & Ch1Max
    clrf        0x7F & Ch1Max+1
    movlw       0xFF
    movwf       0x7F & Ch0Min
    movwf       0x7F & Ch1Min
    movlw       0xC0
    movwf       0x7F & Ch0Min+1
    movwf       0x7F & Ch1Min+1



NextSample

    bcf         status,rp0
    bcf         PORTA,2                     ; Set S/H to hold
    ADC_SetCh0                          ; Select ADC channel 0
    movlw       30
    call        WaitMicroSeconds
    ADC_StartRead                       ; Start AD conversion Channel 0
    ADC_ReadFinished                  ; Wait for conversion to finish

    movf        ADRESH,w
    bsf         status,rp0
    movwf       0x7F & Ch0Value
    movf        0x7F & ADRESL,w
    movwf       0x7F & Ch0Value+1
    bcf         status,rp0

    ADC_SetCh1                          ; Select ADC channel 0
    movlw       30
    call        WaitMicroSeconds
    ADC_StartRead                       ; Start AD conversion Channel 0
    ADC_ReadFinished                  ; Wait for conversion to finish

    movf        ADRESH,w
    bsf         status,rp0
    movwf       0x7F & Ch1Value
    movf        0x7F & ADRESL,w
    movwf       0x7F & Ch1Value+1

    bcf         status,rp0
    bsf         PORTA,2                     ; S/H to sample
    bsf         status,rp0

    movf        0x7F & Ch0Value+1,w         ; Add new value to Ch0Avg
    addwf       0x7F & Ch0Avg+3,f
    btfss       status,c
    goto        CalMode_101
    incf        0x7F & Ch0Avg+2,f
    btfss       status,z
    goto        CalMode_101
    incf        0x7F & Ch0Avg+1,f
    btfsc       status,z
```

```
    incf        0x7F & Ch0Avg,f
CalMode_101
    movf        0x7F & Ch0Value,w
    addwf       0x7F & Ch0Avg+2,f
    btfss       status,c
    goto        CalMode_102
    incf        0x7F & Ch0Avg+1,f
    btfsc       status,z
    incf        0x7F & Ch0Avg,f
CalMode_102


    movf        0x7F & Ch1Value+1,w         ; Add new value to Ch1Avg
    addwf       0x7F & Ch1Avg+3,f
    btfss       status,c
    goto        CalMode_103
    incf        0x7F & Ch1Avg+2,f
    btfss       status,z
    goto        CalMode_103
    incf        0x7F & Ch1Avg+1,f
    btfsc       status,z
    incf        0x7F & Ch1Avg,f
CalMode_103
    movf        0x7F & Ch1Value,w
    addwf       0x7F & Ch1Avg+2,f
    btfss       status,c
    goto        CalMode_104
    incf        0x7F & Ch1Avg+1,f
    btfsc       status,z
    incf        0x7F & Ch1Avg,f
CalMode_104

; Now see if new maximum values found

    movf        0x7F & Ch0Value,w
    subwf       0x7F & Ch0Max,w                 ; 256 + (Ch0Max - Ch0Value); c = 0 means new maximum found
    btfss       status,c
    goto        CalMode_111                     ; New maximum found
    btfss       status,z
    goto        CalMode_120                     ; Not a new maximum
    movf        0x7F & Ch0Value+1,w             ; MSB are equal so check LSB
    subwf       0x7F & Ch0Max+1,w           ; 256 + ((Ch0Max+1) - (Ch0Value+1)); c = 0 means new maximum found
    btfsc       status,c
    goto        CalMode_120                     ; Not a new maximum
CalMode_111                                     ; New maximum found
    movf        0x7F & Ch0Value,w
    movwf       0x7F & Ch0Max
    movf        0x7F & Ch0Value+1,w
    movwf       0x7F & Ch0Max+1

CalMode_120
    movf        0x7F & Ch1Value,w
    subwf       0x7F & Ch1Max,w                 ; 256 + (Ch1Max - Ch1Value); c = 0 means new maximum found
    btfss       status,c
    goto        CalMode_121                     ; New maximum found
    btfss       status,z
    goto        CalMode_130                     ; Not a new maximum
    movf        0x7F & Ch1Value+1,w             ; MSB are equal so check LSB
    subwf       0x7F & Ch1Max+1,w           ; 256 + ((Ch1Max+1) - (Ch1Value+1)); c = 0 means new maximum found
    btfsc       status,c
    goto        CalMode_130                     ; Not a new maximum
CalMode_121                                     ; New maximum found
    movf        0x7F & Ch1Value,w
    movwf       0x7F & Ch1Max
    movf        0x7F & Ch1Value+1,w
    movwf       0x7F & Ch1Max+1

CalMode_130
; Now see if new minimum values found

    movf        0x7F & Ch0Min,w
    subwf       0x7F & Ch0Value,w               ; 256 + (Ch0Value - Ch0Min); c = 0 means new Minimum found
    btfss       status,c
    goto        CalMode_131                     ; New Minimum found
    btfss       status,z
    goto        CalMode_140                     ; Not a new Minimum
    movf        0x7F & Ch0Min+1,w           ; MSB are equal so check LSB
    subwf       0x7F & Ch0Value+1,w             ; 256 + ((Ch0Value+1) - (Ch0Min+1)); c = 0 means new Minimum found
    btfsc       status,c
    goto        CalMode_140                     ; Not a new Minimum
CalMode_131                                     ; New Minimum found
```

```
    movf        0x7F & Ch0Value,w
    movwf       0x7F & Ch0Min
    movf        0x7F & Ch0Value+1,w
    movwf       0x7F & Ch0Min+1


CalMode_140
; Now see if new minimum values found

    movf        0x7F & Ch1Min,w
    subwf       0x7F & Ch1Value,w               ; 256 + (Ch1Value - Ch1Min); c = 0 means new Minimum found
    btfss       status,c
    goto        CalMode_141                     ; New Minimum found
    btfss       status,z
    goto        CalMode_150                     ; Not a new Minimum
    movf        0x7F & Ch1Min+1,w           ; MSB are equal so check LSB
    subwf       0x7F & Ch1Value+1,w             ; 256 + ((Ch1Value+1) - (Ch1Min+1)); c = 0 means new Minimum found
    btfsc       status,c
    goto        CalMode_150                     ; Not a new Minimum
CalMode_141                                     ; New Minimum found
    movf        0x7F & Ch1Value,w
    movwf       0x7F & Ch1Min
    movf        0x7F & Ch1Value+1,w
    movwf       0x7F & Ch1Min+1

CalMode_150
;   Have we taken enough samples?
    movlw       1
    subwf       0x7F & SampleCounter+1,f
    btfss       status,c
    decf        0x7F & SampleCounter,f
    movf        0x7F & SampleCounter+1,f
    btfss       status,z
    goto        NextSample
    movf        0x7F & SampleCounter,f
    btfss       status,z
    goto        NextSample
; All samples taken. Prepare data for display

; Divide Ch0Avg and Ch1Avg by 1024
; First Ch0Avg

    movlw           0x80
    subwf           0x7F & Ch0Avg+2,w    ; 256 + ((Ch0AVg+2) - 0x80)  Ch0Avg + 2 first 8 bits of remainder
    btfss           status,c             ; Skip if Ch0AVg+2 >= 0x80
    goto            CalMode_210              ; Round down
    btfsc           status,c             ; Is difference 0.
    goto            CalMode_203              ; Yes--Round so as to make result even number

CalMode_201                                  ; Round up
    incf            0x7F & Ch0Avg+1,f
    btfsc           status,z
    incf            0x7F & Ch0Avg,f
    goto            CalMode_210

CalMode_203                                  ; Come here to determine whether result odd or even
    btfsc           0x7F & Ch0Avg,0
    goto            CalMode_201              ; Result is odd so round up

CalMode_210

; Next Ch1Avg

    movlw           0x80
    subwf           0x7F & Ch1Avg+2,w    ; 256 + ((Ch0AVg+2) - 0x80)  Ch0Avg + 2 first 8 bits of remainder
    btfss           status,c             ; Skip if Ch0AVg+2 >= 0x80
    goto            CalMode_220              ; Round down
    btfsc           status,c             ; Is difference 0.
    goto            CalMode_213              ; Yes--Round so as to make result even number

CalMode_211                                  ; Round up
    incf            0x7F & Ch1Avg+1,f
    btfsc           status,z
    incf            0x7F & Ch1Avg,f
    goto            CalMode_220

CalMode_213                                  ; Come here to determine whether result odd or even
    btfsc           0x7F & Ch1Avg,0
    goto            CalMode_211              ; Result is odd so round up

CalMode_220
```

```
; Now display result on LCD. First Channel0
    bcf         status,rp0
    movlw       Line1+5
    call LCD_SetLineAndColumn
    MoveBytes   Ch0Avg,aarg+1,2
    bcf         status,rp0
    call        Int16ToASCII_Num
    clrf        ASCII_Decimal
    clrf        ASCII_Sign
    movlw       4
    movwf       ASCII_MinField
    call        LCD_DisplayASCII_Num
    LCD_DisplayText " (",uadr8771

    bsf         status,rp0
    movlw       6
    movwf       0x7F & B1Temp1
    bcf         status,c
CalMode_230
    rrf         0x7F & Ch0Min,f
    rrf         0x7F & Ch0Min+1,f
    decfsz      0x7F & B1Temp1,f
    goto        CalMode_230

    MoveBytes   Ch0Min,aarg+1,2
    bcf         status,rp0
    call        Int16ToASCII_Num
    call        LCD_DisplayASCII_Num
    LCD_DisplayText ",",uadr8772

    bsf         status,rp0
    movlw       6
    movwf       0x7F & B1Temp1
    bcf         status,c
CalMode_231
    rrf         0x7F & Ch0Max,f
    rrf         0x7F & Ch0Max+1,f
    decfsz      0x7F & B1Temp1,f
    goto        CalMode_231

    MoveBytes   Ch0Max,aarg+1,2
    bcf         status,rp0
    call        Int16ToASCII_Num
    call        LCD_DisplayASCII_Num
    LCD_DisplayText ")",uadr8773

; Then Channel 1
    movlw       Line2+5
    call LCD_SetLineAndColumn

    MoveBytes   Ch1Avg,aarg+1,2
    bcf         status,rp0
    call        Int16ToASCII_Num
    clrf        ASCII_Decimal
    clrf        ASCII_Sign
    movlw       4
    movwf       ASCII_MinField
    call        LCD_DisplayASCII_Num
    LCD_DisplayText " (",uadr8774

    bsf         status,rp0
    movlw       6
    movwf       0x7F & B1Temp1
    bcf         status,c
CalMode_232
    rrf         0x7F & Ch1Min,f
    rrf         0x7F & Ch1Min+1,f
    decfsz      0x7F & B1Temp1,f
    goto        CalMode_232

    MoveBytes   Ch1Min,aarg+1,2
    bcf         status,rp0
    call        Int16ToASCII_Num
    call        LCD_DisplayASCII_Num
    LCD_DisplayText ",",uadr8775

    bsf         status,rp0
    movlw       6
    movwf       0x7F & B1Temp1
    bcf         status,c
```

```
CalMode_233
    rrf         0x7F & Ch1Max,f
    rrf         0x7F & Ch1Max+1,f
    decfsz      0x7F & B1Temp1,f
    goto        CalMode_233

    MoveBytes   Ch1Max,aarg+1,2
    bcf         status,rp0
    call        Int16ToASCII_Num
    call        LCD_DisplayASCII_Num
    LCD_DisplayText ")",uadr8776

    goto        StartSeries




 endif



 if Calibrate == 0

;*****************************************************************
; This subroutine calculates the SWR from forward and reflected powers
; (or these powers multiplied by a constant) according to the formula

;   SWR = (1 + sqrt(Pr/Pf))/(1 - sqrt(Pr/Pf)).

; This will actually be calculated as follows:
;
;   SWR = (-2/(sqrt(Pr/Pf) - 1) -1
;
; The subroutine is entered with Pr in aarg and Pf in barg.
; The SWR is returned in aarg


CalculateSWRandSetMeter
; Is SWR based on values of peak power too low to display?
; First check forward power
    bsf         status,rp0
    movlw       LowPowerThreshold
    subwf       0x7F & bForward,w    ; bForward - LowPowerThreshold
    bcf         status,rp0
    btfsc       status,c     ; Is <0?
    goto        mm03
    bsf         ModeStatusFlags,LowPeakWatts  ; Power too low
    decfsz      SWRMeterWaitCounter,f ; Have we waited enough times when power is low?
    return                   ; No
    clrw                     ; We have waited enough. Set SWR meter to 0
    call        PWM_SetDutyCycle
    return
mm03
    bcf ModeStatusFlags,LowPeakWatts   ; Power not too low
    movlw       MaxNumSWRMeterWaits  ; Initialize meter wait counter
    movwf       SWRMeterWaitCounter


    bsf         PCLATH,psel0
    call        0x7ff & FPD24        ; Yes, calculate sqrt(Pr/Pf) = rho
    MoveBytes   aarg,LoadFactor,3         ; LoadFactor = rho^2
    call        0x7ff & sqrt     ; Calculate rho
    StoreFPLiteral   0x7f0000,barg    ; Load 1.0
    call        0x7ff & FPS24        ; rho - 1
    MoveBytes   aarg,barg,3
    StoreFPLiteral 0x808000,aarg  ; Store -2
    call        0x7ff & FPD24        ; Calculate  -2/(rho - 1)
    StoreFPLiteral   0x7F8000,barg    ; Load -1
    call        0x7FF & FPA24        ; This is the SWR
    bcf         PCLATH,psel0
    andlw       0xff         ; is w = 0 ?
    btfss       status,z
    goto        mm01         ; w != 0: Divide by 0 so SWR large
    btfss       aarg+1,7     ; Is SWR negative? If so, SWR large
    goto        mm02

mm01
    StoreFPLiteral 0x887A00,aarg  ; Set SWR to 1000
mm02
```

```
    MoveBytes aarg,SWR,3


; This next section takes a 24-bit floating point SWR value, in aarg,
; and displays it on the panel meter.
; The meter reading will be 256*(1 - 1/SWR), where
; 0 is at the bottom of the scale and 256 is full scale

    MoveBytes aarg,barg,3
    movlw       0x7f         ; Load 1.0
    movwf       aarg
    clrf        aarg+1
    clrf        aarg+2
    bsf         PCLATH,psel0
    call        0x7ff & FPD24        ; Result =1/SWR
    movlw       0x7f         ; Load 1.0
    movwf       barg
    clrf        barg+1
    clrf        barg+2
    call        0x7ff & FPS24    ; This is 1/SWR - 1. It should always be negative. If it is positive, then SWR = 1
    bcf         PCLATH,psel0
    btfsc       aarg+1,7     ; Skip if value positive.
    goto        mm10
    clrw                         ; Set meter to 0, indicating 1:1 SWR
    call        PWM_SetDutyCycle
    return
mm10
    bcf         aarg+1,7     ; Result = 1 - 1/SWR
    movlw       8            ; Multiply by 256
    addwf       aarg,f
    bsf         PCLATH,psel0
    call        0x7ff & INT2416
    bcf         PCLATH,psel0
    movlw       0xff
    andwf       aargb0,f     ; Test to see if result > 255
    btfsc       status,z     ; Skip if too large
    movf        aargb1,w     ; Otherwise load actual value
    call        PWM_SetDutyCycle
    return

;*****************************************************************
; This subroutine writes the peak forward power and SWR to the LCD

DisplayPeakPowerAndSWR
; First determine if power is large enough to display
    btfss       ModeStatusFlags,LowPeakWatts      ; Is power too low?
    goto        t001
; Power too low
    movlw       Line1 + 8
    call        LCD_SetLineAndColumn
    LCD_DisplayText " Low      ", uAdr5
    movlw       Line2 + 16
    call        LCD_SetLineAndColumn
    LCD_DisplayText "     ", uAdr5a
    return

; Power large enough to display

t001
    MoveBytes PeakForwardWatts,aarg,3
    btfsc       ModeStatusFlags,ForwardWatts      ;Display Forward or Load Watts
    goto        t002
; Calculate watts delivered to load
    StoreFPLiteral   0x7F0000,aarg                ; Load 1.0
    MoveBytes    LoadFactor,barg,3            ; Load rho^2
    bsf         PCLATH,psel0
    call        0x7ff & FPS24                         ; = 1 - rho^2
    MoveBytes    PeakForwardWatts,barg,3
    call        0x7ff & FPM24                         ; Load power = Pf*(1 - rho^2)
    bcf         PCLATH,psel0

; Display power in watts
t002
    call        FormatWatts
    movlw       4
    movwf       ASCII_MinField
    movlw       Line1 + 8
    call        LCD_SetLineAndColumn
    call        LCD_DisplayASCII_Num
    btfsc       ModeStatusFlags,MilliWattsFlag
    goto        t003                              ; Units are milliwatts
```

```
    LCD_DisplayText " W ", uAdr57             ; or units are watts
    goto        t004
t003
    nop
    LCD_DisplayText " mW",uAdr7891
t004
; Display SWR
    call        FormatSWR
    movlw       5
    movwf       ASCII_MinField
    movlw       Line2 + 16
    call        LCD_SetLineAndColumn
    call        LCD_DisplayASCII_Num
    return



;*****************************************************************
; This subroutine writes the average forward or load power to the LCD. Enter with
; the forward power in aarg.

DisplayAvgPower
; First determine if power is large enough to display
    bcf         status,rp0
    btfss       ModeStatusFlags,LowPeakWatts
    goto        t101
; Power too low
    movlw       Line2 + 8
    call        LCD_SetLineAndColumn
    LCD_DisplayText " Low    ", uAdr67
    return

; Power large enough to display

t101
    btfsc       ModeStatusFlags,ForwardWatts      ;Display forward or load power
    goto        t102

; Calculate load power
    MoveBytes   aarg,t1,3                         ; Temporary storage of average power
    StoreFPLiteral   0x7F0000,aarg                ; Load 1.0
    MoveBytes   LoadFactor,barg,3; Load rho^2
    bsf         PCLATH,psel0
    call        0x7ff & FPS24                         ; = 1 - rho^2
    MoveBytes   t1,barg,3
    call        0x7ff & FPM24                         ; Load power = Pf*(1 - rho^2)
    bcf         PCLATH,psel0

t102
    call        FormatWatts
    movlw       4
    movwf       ASCII_MinField
    movlw       Line2 + 8
    call        LCD_SetLineAndColumn
    call        LCD_DisplayASCII_Num
    btfsc       ModeStatusFlags,MilliWattsFlag    ; Units are W or mW?
    goto        t103
    LCD_DisplayText " W ", uAdr76
    return
t103
    LCD_DisplayText " mW", uAdr761145
    return



;*****************************************************************
; This subroutine sets up Watts data for output

FormatWatts
; Determine scale of power.
    clrf        ASCII_Decimal
    movlw       5
    movwf       temp2
    bcf         ModeStatusFlags,MilliWattsFlag
    StoreFPLiteral   0x844800,barg    ; Load value 20

FW_02
    bsf PCLATH,psel0
    call        0x7ff & TAGEB24       ; Is 10^(4 - count)*watts >= 20?
    bcf         PCLATH,psel0
    andlw       0xff
```

```
    btfss       status,z
    goto        FW_01
    bsf         PCLATH,psel0        ; No, < 20
    call        0x7ff & AargTimes10
    bcf         PCLATH,psel0
    incf        ASCII_Decimal,f
    movlw       3
    subwf       ASCII_Decimal,w
    btfss       status,z
    goto        FW_03
    clrf        ASCII_Decimal
    bsf         ModeStatusFlags,MilliWattsFlag
FW_03
    decfsz      temp2,f
    goto        FW_02

FW_01
    clrf        ASCII_Sign
    bsf         PCLATH,psel0
    StoreFPLiteral   0x7E0000,Barg
    call        0x7ff & FPA24
    call        0x7ff & INT2416          ; Integer floor
    pagesel     Int16ToASCII_Num
    call        Int16ToASCII_Num
    return




;*****************************************************************
; This routine formats SWR data for output

FormatSWR
    MoveBytes SWR,aarg,3
    clrf        ASCII_Sign
    StoreFPLiteral   0x832000,barg        ; Load 20
    pagesel     TAGEB24
    call        0x7ff & TAGEB24
    pagesel     SWRge20
    andlw       0xff        ; Determine if w = 0
    btfss       status,z    ; Skip if w != 0, i.e., SWR <20
    goto        SWRge20         ; Comes here if SWR >= 20
    StoreFPLiteral   0x822000,barg    ; Load 10
    pagesel     TAGEB24
    call        0x7ff & TAGEB24
    pagesel     SWRge10
    andlw       0xff        ; Determine if w = 0
    btfss       status,z    ; Skip if w != 0, i.e., SWR <20
    goto        SWRge10         ; Comes here if SWR >= 20
SWRlt10
    pagesel     AargTimes10
    call        0x7ff & AargTimes10
    call        0x7ff & AargTimes10
    movlw       2
    movwf       ASCII_Decimal
    pagesel     ss001
    goto        ss001
SWRge10
    pagesel     AargTimes10
    call        0x7ff & AargTimes10
    movlw       1
    movwf       ASCII_Decimal
ss001
    StoreFPLiteral   0x7e0000,barg    ; Load 0.5
    pagesel     FPA24
    call        0x7ff & FPA24
    call        0x7ff & INT2416
    pagesel     Int16ToASCII_Num
    call        Int16ToASCII_Num
    return
SWRge20
    movlw       0x30
    movwf       ASCII_Num
    movwf       ASCII_Num + 1
    movwf       ASCII_Num + 4
    movlw       0x3e
    movwf       ASCII_Num + 2
    movlw       0x32
    movwf       ASCII_Num + 3
    clrf        ASCII_Decimal
```

```
    return


 endif

;*****************************************************************
; This routine Displays an ASCII Number, in ASCII_Num through ASCII_Num+5
; on the LCD.
; ASCII_Sign = 1 if number negative, 0 if positive
; ASCII_Digits gives the location of the decimal point (i.e., #
; of digits to right of decimal point)
; ASCII_MinField gives the minimum field width in which the number will
; be displayed. The number will always be right justified in this field.


LCD_DisplayASCII_Num
    bcf status,rp0
    clrf        Temp1        ; Flag for leading decimal point
    movlw       5       ; 5 characters in ASCII_Num
    movwf       Temp2
    movlw       ASCII_Num    ; Pointer to first character
    movwf       FSR
s001 movf       Temp2,w      ; Is next character decimal?
    subwf       ASCII_Decimal,w
    btfsc       status,z ; Skip if it is decimal point
    goto        s002
    movlw       0x30     ; Otherwise check to see if leading 0
    subwf       INDF,w
    btfss       status,z ; Skip if it is leading 0
    goto        s002+2
    incf        FSR,f
    decfsz      Temp2,f      ; Temp2 should never be 0. If so
    goto        s001     ; nothing to print so simply print out a 0

    decf        ASCII_MinField,w
    movwf       Temp1
s007
    movlw       0x20     ; Print out leading blanks
    call        LCD_SendChar
    decfsz      Temp1,f
    goto        s007
    movlw       0x30     ; Print out single 0
    call        LCD_SendChar
    return

s002 bsf Temp1,1       ; Yes, decimal point is leading
    incf        Temp2,f
    movlw       0xff     ; Is there a decimal point at all?
    andwf       ASCII_Decimal,w
    btfss       status,z ; Skip if there is no decimal point
    incf        Temp2,f      ; Provide space for decimal
    movlw       0xff     ; Is there a minus sign?
    andwf       ASCII_Sign,w
    btfss       status,z ; Skip if no minus sign
    incf        Temp2,f      ; Provide space for minus sign

; Now output actual characters
; First output spaces, if any, needed to fill up field
    movf        Temp2,w        ; Load total actual field size
    subwf       ASCII_MinField,w  ; Subtract from minimum field size
    btfss       status,c
    goto s006
    btfsc       status,z
    goto s006
    movwf       Temp2        ; Output spaces to fill field
    movlw       0x20
    call        LCD_SendChar
    decfsz      Temp2,f
    goto        $-3

; Determine # of characters excluding sign and decimal point
s006 movf       FSR,w
    sublw       ASCII_Num+5
    movwf       Temp2
    btfss       ASCII_Sign,0 ; Skip if number negative
    goto        $+3
    movlw       0x2d     ; Display minus sign
    call        LCD_SendChar
    btfss       Temp1,1      ; Place 0 in front of leading decimal?
    goto        $+3
    movlw       0x30
```

```
    call        LCD_SendChar
s05 movf        ASCII_Decimal,w  ; Display decimal point?
    subwf       Temp2,w
    btfss       status,z ; Skip if place for decimal point
    goto        s03      ;
    movlw       0x2e     ; Load decimal point
    call        LCD_SendChar
s03 movf        INDF,w       ; Load character
    call        LCD_SendChar ; Display character
s04 incf        FSR,f        ; Increment pointer to next character
    decfsz      temp2,f      ; Decrement character counter
    goto        s05
    return



;*****************************************************************
;*****************************************************************
;
; These routines control the PWM1 module.
;
; PWM_Initialize initializes the PWM1 module for this purpose
; PWM_SetDutyCycle sets the duty cycle for the PWM1 module.
;

;   Setup PWM to have a pulse frequency of 4.88 kHz and a duty cycle resolution of 1 part in 256


PWM_Initialize
    movlw       0x3f                ; Will count up to 64
    bsf status,rp0
    movwf       PR2 & 0x7f
    bcf status,rp0
    clrf        CCPR1L                  ; Start with duty cycle = 0
    bsf status,rp0
    bcf (TRISC & 0x7f),2  ; Set RC2 as output
    bcf status,rp0
    movlw       0x06                ; 1:16 prescale, 1:1 postscale, turn counter on
    movwf       T2CON
    movlw       0x0C                ; Set PWM mode.
    movwf       CCP1CON
    return


;*****************************************************************
; This subroutine sets the 8-bit duty cycle of PWM1.
; Enter with the 8-bit duty cycle in the w register

PWM_SetDutyCycle
    bcf         status,rp0
    movwf       temp     ; Store duty cycle in temp
r111
    movf        TMR2,w       ; Determine how close to the end of the
    bsf         status,rp0
    subwf       PR2 & 0x7f,w ; Form PR2 - TMR2
    bcf         status,rp0
    andlw       0xf0     ; This will be nonzero if PR2 - TMR2 > 15
    btfsc       status,z ; If zero, too close, wait for overflow
    goto        r111
    bcf         CCP1CON,ccp1x    ; Update LSB's in CCP1CON. First 0 them
    bcf         CCP1CON,ccp1y
    rrf         temp,f       ; Rotate LSB of duty cycle into C
    btfsc       status,c ; If 1, set the ccp1y bit
    bsf         CCP1CON,ccp1y
    rrf         temp,f       ; Rotate next LSB into C
    btfsc       status,c ; If 1, set the ccp1x bit
    bsf         CCP1CON,ccp1x
    movf        temp,w
    andlw       0x3f
    movwf       ccpr1l       ; Place in ccpr1l register
    return


;*****************************************************************
;*********************************************************************
;
;  These routines implement an 8 bit interface to a Hitachi
;  LCD module, busy flag used when valid.  The data lines
;  are on port B, E is on port_c bit6,
;  R/W is on port_c bit 5, RS is on port_c bit 4.

LCDDataPORT  EQU     PORTB        ;LCD data port
```

```
LCDDataTRIS  EQU     TRISB    ;LCD data TRIS register
LCDControlPORT EQU     PORTC       ;LCD control port
LCDControlTRIS EQU     TRISC    ;LCD Control TRIS register

E       EQU     6    ;LCD enable signal
RW      EQU     5    ;LCD R/W signal
RS      EQU     4    ;LCD register select


;*  LCD_SendChar - Sends character contained in register W to LCD    *

LCD_SendChar
    bcf          status,rp0
    movwf   LCDChar             ;Character to be sent is in W
    call    LCD_BusyCheck       ;Wait for LCD to be ready
    movf    LCDChar,w
    movwf   LCDDataPORT             ;Send data to LCD
    bcf     LCDControlPORT,RW        ;Set LCD in read mode
    bsf     LCDControlPORT,RS        ;Set LCD in data mode
    nop
    bsf     LCDControlPORT,E         ;toggle E for LCD
    nop
    bcf     LCDControlPORT,E
    return


;*****************************************************************
;*  LCD_SendCmd - Sends command contained in register W to LCD    *

LCD_SendCmd
    bcf          status,rp0
    movwf   LCDChar             ;Command to be sent is in W
    call    LCD_BusyCheck       ;Wait for LCD to be ready
    movf    LCDChar,w
    movwf   LCDDataPORT             ;Send data to LCD
    bcf     LCDControlPORT,RW        ;Set LCD in read mode
    bcf     LCDControlPORT,RS        ;Set LCD in command mode
    nop
    bsf     LCDControlPORT,E         ;toggle E for LCD
    nop
    bcf     LCDControlPORT,E
    return


;*****************************************************************
;*  This routine sets the DDRAM address to the line and column # contained in w.
; Suppose it is desired to place the "cursor" at Line 2, column 7. The following
; code will do this:
;
;   movlw       Line2 + 7
;   call        LCD_SetLineAndColumn

Line1   equ      0x7f
Line2   equ      0xbf

LCD_SetLineAndColumn
   call LCD_SendCmd
   return



;*****************************************************************
; This routine clears display, moves cursor to beginning of line 1

LCD_ClearDisplay
    movlw       0x01
    call        LCD_SendCmd
    return


;*****************************************************************
;* This routine checks the busy flag, returns when not busy   *
;*  Affects:                                                   *
;*      LCDTemp - Returned with busy/address                      *


LCD_BusyCheck
 if Simulate==0
    ;Set Data for input
    bsf     STATUS,RP0
    movlw   0xFF
```

```
    movwf   0x7f&LCDDataTRIS
    bcf     STATUS,RP0
    bcf     LCDControlPORT,RS        ;Set LCD for command mode
    bsf     LCDControlPORT,RW        ;Setup to read busy flag
LCDBS001
    bsf     LCDControlPORT,E         ;Set E high
    nop
    movf    LCDDataPORT,W           ;Read busy flag, DDram address
    bcf     LCDControlPORT,E         ;Set E low
    andlw       0x80                 ; Test Bit 7, the busy flag
    btfss       status,z
    goto    LCDBS001

    bcf     LCDControlPORT,RW
    bsf     STATUS,RP0      ;Set Data for output
    clrf        0x7f & LCDDataTRIS
    bcf     STATUS,RP0
    return
 else
    return
 endif

;*****************************************************************
;* This routine initializes the LCD module                       *
;*  Affects:                                                     *
;*      LCDTemp - Returned with busy/address                        *

LCD_Init
    bcf         status,rp0
    movlw       20
    call        WaitMilliSeconds     ; Wait for 20 ms

    bsf     STATUS,RP0            ;---Bank 1
    clrf    0x7f&LCDDataTRIS          ; Set all 8 data pins as outputs
    bcf     0x7f&LCDControlTRIS,E     ; Set three pins as outputs--Others inputs
    bcf     0x7f&LCDControlTRIS,RW
    bcf     0x7f&LCDControlTRIS,RS
    bcf     STATUS,RP0            ;---Bank 0

    clrf        LCDDataPORT
    bcf     LCDControlPORT,E
    bcf     LCDControlPORT,RW
    bcf     LCDControlPORT,RS

    movlw   0x38         ;Set LCD to 8 bit interface
    movwf   LCDDataPORT
    bsf     LCDControlPORT,E         ;toggle E for LCD
    nop
    bcf     LCDControlPORT,E
    movlw   6
    call    WaitMilliSeconds          ; Wait for 6 ms

    movlw   0x38         ;Function set to 2 lines
    movwf   LCDDataPORT             ;of 5x7 bit chars
    bsf     LCDControlPORT,E         ;toggle E for LCD
    nop
    bcf     LCDControlPORT,E
    movlw   1
    call         WaitMilliSeconds        ; Wait for 1 msec

;Busy flag should be valid after this point
    movlw       0x38
    call        LCD_SendCmd

    movlw   0x06         ;Increment cursor, no shift
    call    LCD_SendCmd

    movlw   0x0C         ;Turn display, cursor off
    call    LCD_SendCmd

    movlw   0x01         ;Clear display, cursor to begin
    call    LCD_SendCmd       ; of line 1

    return




;*****************************************************************
;***************************************************************
;
```

```
; This routine takes a 16 bit integer (always positive) located in
; aargb0 (most significant 8 bits) and aargb1 and produces a six character
; ASCII string that expresses the number in decimal characters. The result is placed
; in ASCII_Num, ASCII_Num+1, ... , ASCII_Num+4.
;

Int16ToASCII_Num
    bcf status,rp0        ; Data in bank 0
    movlw       0x27          ; Load integer 10000 into bargb0, 1
    movwf       bargb0
    movlw       0x10
    movwf       bargb1
    call        DivideDown       ; Subtract 10,000 repeatedly until
    movwf       ASCII_Num        ; total less than 10,000. # of subtractions
                        ; is 10000's digit. STore in ASCII_num

    movlw       0x03          ; load 1000
    movwf       bargb0
    movlw       0xe8
    movwf       bargb1
    call        DivideDown
    movwf       ASCII_Num + 1

    movlw       0x00          ; load 100
    movwf       bargb0
    movlw       0x64
    movwf       bargb1
    call        DivideDown
    movwf       ASCII_Num + 2

    movlw       0x00          ; load 10
    movwf       bargb0
    movlw       0x0a
    movwf       bargb1
    call        DivideDown
    movwf       ASCII_Num + 3

    movf        aargb1,w      ; Remainder in aargb1 is units digit
    addlw       0x30          ; Add 0x30 to convert to ASCII
    movwf       ASCII_Num + 4
    return

;*****************************************************************
; This routine repeatedly subtracts bargb0:b1 from aargb0:b1 until result is no longer positive

DivideDown
    clrf        temp1             ; Counts # of succesful subtractions
r11 call        Int16Sub      ; subtract bargb0:b1 from aargb0:b1
    andlw       0xff          ; Check is result < 0
    btfss       status,z
    goto        d1            ; Yes, result was negative
    incf        temp1,f       ; No, result still positive
    goto        r11           ; Subtract again
d1  movf        temp1,w       ; Last subtraction resulted in - result
                        ; aargb0:b1 restored to last + value
    addlw       0x30          ; Add 0x30 to count to convert to ASCII
    return

;*****************************************************************
; This routine does a 16 bit integer subtraction if the result is positive

Int16Sub
    MoveBytes aargb0,t1,2 ; Save aargb0:b1 in case result of subtraction is -
    movf        bargb1,w      ; Subtract least significant 8 bits
    subwf       aargb1,f      ; Result goes in aargb1
    btfsc       status,c      ; Was a borrow necessary?
    goto        r1            ; No, jump to r1
    movlw       1             ; Yes. Subtract 1 from aargb0
    subwf       aargb0,f      ; Place result back in aargb0
    btfsc       status,c      ; If borrow necessary, overall result negative.
    goto        r1            ; No borrow
    MoveBytes t1,aargb0,2 ; Borrow necessary-result negative-restore aargb0:b1
    retlw       0xff          ; Return with w=0xff: result was negative
r1  movf        bargb0,w      ; Subtract most significant 8 bits
    subwf       aargb0,f      ; Place result back in aargb0
    btfsc       status,c      ; Was borrow necessary?
    retlw       0x00          ; No: result was positive, return
    MoveBytes t1,aargb0,2 ; Yes, restore aargb0:b1 to previous value
    retlw       0xff          ; REturn with w=0xff; result was negative
```

```
;*****************************************************************************
;
;       This routine sets a time delay in units of microseconds.
;       Enter with the number of units to wait in the w register.
;       The minumum wait is 3 microseconds.


WaitMicroSeconds
 if Simulate>0
    return
 endif
    banksel      0
    movwf        Timer
    decf         Timer,f
    decf         Timer,f
    nop
    nop
    nop
    nop
    decfsz       Timer,f
    goto         $-3
    return


;*****************************************************************
; This routine waits for units of milliseconds.
; Enter with the number of units to wait in w register.
; The minimum wait is 1 millisecond


WaitMilliSeconds
 if Simulate>0
    return
 endif
    banksel      0
    movwf        Timer2
    decfsz       Timer2,f
    goto         WMS1
    goto         WMS2
WMS1
    movlw        250
    Call         WaitMicroSeconds
    movlw        250
    Call         WaitMicroSeconds
    movlw        250
    Call         WaitMicroSeconds
    movlw        248
    Call         WaitMicroSeconds
    nop
    nop
    nop
    decfsz       Timer2,f
    goto         WMS1
    nop
    nop
WMS2
    movlw        250
    Call         WaitMicroSeconds
    movlw        250
    Call         WaitMicroSeconds
    movlw        250
    Call         WaitMicroSeconds
    movlw        247
    Call         WaitMicroSeconds
    return



;*****************************************************************
; This routine waits for units of tenths of seconds.
; Enter with the number of units to wait in w register.
; The minimum wait is 0.1 second.


WaitTenthSeconds
 if Simulate>0
    return
 endif
    banksel 0
```

```
    movwf       Timer3
    decfsz      Timer3,f
    goto        WTS1
    goto        WTS2
WTS1
    movlw       99
    call        WaitMilliSeconds
    movlw       250
    call        WaitMicroSeconds
    movlw       250
    call        WaitMicroSeconds
    movlw       250
    call        WaitMicroSeconds
    movlw       248
    call        WaitMicroSeconds
    nop
    nop
    decfsz      Timer3,f
    goto        WTS1
    nop
    nop
WTS2
    movlw       99
    call        WaitMilliSeconds
    movlw       250
    call        WaitMicroSeconds
    movlw       250
    call        WaitMicroSeconds
    movlw       250
    call        WaitMicroSeconds
    movlw       246
    call        WaitMicroSeconds
    nop
    nop
    nop
    nop
    return


;************* GetForwardWatts ********************************************************************************
; This routine uses a lookup table to translate 10-bit binary ADC values, located in bForwardNew, into packed-16 values
; of the forward power. The result is returned in FP and FP+1
; The actually lookup table takes all of page 2 of memory.


GetForwardWatts
    bsf         status,rp0
    call        _GetForwardLowByte
    movwf       (0x7F & FP)+1
    bcf         PCLATH,psel1
    call        _GetForwardHighByte
    movwf       (0x7F & FP)
    return

_GetForwardLowByte
    movlw       0x10            ; Lookup table begins at beginning of page 2.
    movwf       PCLATH
    btfsc       0x7F & bForwardNew+1,6
    bsf         PCLATH,0
    btfsc       0x7F & bForwardNew+1,7
    bsf         PCLATH,1
    movf        0x7F & bForwardNew,w
    movwf       PCL


_GetForwardHighByte
    movlw       0x14            ; Lookup table begins at beginning of page 2.
    movwf       PCLATH
    btfsc       0x7F & bForwardNew+1,6
    bsf         PCLATH,0
    btfsc       0x7F & bForwardNew+1,7
    bsf         PCLATH,1
    movf        0x7F & bForwardNew,w
    movwf       PCL



;************* GetReflectedWatts ******************************************************************************
; This routine uses a lookup table to translate 10-bit binary ADC values into packed-16 values of the reflected power.
; The actual lookup table takes all of page 3 of memory.
```

```
GetReflectedWatts
    bsf         status,rp0
    call        _GetReflectedLowByte
    movwf       (0x7F & FP)+1
    bcf         PCLATH,psel0
    bcf         PCLATH,psel1
    call        _GetReflectedHighByte
    movwf       (0x7F & FP)
    return

_GetReflectedLowByte
    movlw       0x18            ; Lookup table for reflected power begins at beginning of page 3.
    movwf       PCLATH
    btfsc       0x7F & bReflected+1,6
    bsf         PCLATH,0
    btfsc       0x7F & bReflected+1,7
    bsf         PCLATH,1
    movf        0x7F & bReflected,w
    movwf       PCL


_GetReflectedHighByte
    movlw       0x1C            ; Lookup table for reflected power begins at beginning of page 3.
    movwf       PCLATH
    btfsc       0x7F & bReflected+1,6
    bsf         PCLATH,0
    btfsc       0x7F & bReflected+1,7
    bsf         PCLATH,1
    movf        0x7F & bReflected,w
    movwf       PCL







 if Calibrate != 1

;*****************************************************************************
;   Locate floating point routines in page 1

  org   0x0800


; ************** SumP16  ***************************************************************************************
;
; This routine adds the packed 16-bit value in FP (Bank 1) to the packed 32-bit running sum in FPaccum (Bank 1). The result is
returned
; in FPaccum.
; My packed format works as follows: All numbers are positive. The first six bits are the exponent, which is offset by
; 27. An exponent of 0 means the value is zero. The mantissa is the remaining 10 bits, where the leading 1 is suppressed.


SumP16
    bsf         status,rp0
    movf        (0x7F & FPaccum),w                  ; Calculate difference of exponents and
    andlw       0xFC                            ; determine which exponent is larger
    movwf       0x7F & B1Temp1
    movf        0x7F & FP,w
    andlw       0xFC
    subwf       0x7F & B1Temp1,w            ; 4*(Exp(FPaccum) - Exp(FP))
    btfsc       status,c
    goto        SumP16_001                      ; FPaccum has larger exponent.

; Come here if Exp(FP) > Exp(FPaccum). We always want the opposite, so switch the two values.

    sublw       0                               ; Calculate negative of exponent difference
    movwf       0x7F & B1Temp3                  ; B1Temp1 = 4*(Exp(FPaccum) - Exp(FP))
    movf        0x7F & FP,w                     ; Now interchange FP and FPaccum
    movwf       0x7F & B1Temp1
    movf        0x7F & FP+1,w
    movwf       0x7F & B1Temp2
    movf        0x7F & FPaccum,w
    movwf       0x7F & FP
    movf        0x7F & FPaccum+1,w
    movwf       0x7F & FP+1
    movf        0x7F & FPaccum+2,w
```

```
    movwf       0x7F & FP+2
    movf        0x7F & FPaccum+3,w
    movwf       0x7F & FP+3
    movf        0x7F & B1Temp1,w
    movwf       0x7F & FPaccum
    movf        0x7F & B1Temp2,w
    movwf       0x7F & FPaccum+1
    clrf        0x7F & FPaccum+2
    clrf        0x7F & FPaccum+3
    movf        0x7F & FPaccum,w            ; Now put larger exponent * 4 into B1Temp1
    andlw       0xFC
    movwf       0x7F & B1Temp1
    goto        SumP16_001A

SumP16_001
    movwf       (0x7F & B1Temp3)                ; Store exponent difference.
    clrf        (0x7F & FP)+2
    clrf        (0x7F & FP)+3

SumP16_001A
    movf        0x7F & FP,w                     ; Is FP 0? If so return with FPaccum unchanged
    andlw       0xFC
    btfsc       status,z
    return

    bcf         status,c                    ; B1Temp3 = 4(Exp(FPaccum) - Exp(FP)). Divide by 4
    rrf         (0x7F & B1Temp3),f
    rrf         (0x7F & B1Temp3),f                  ; Two right shifts and B1Temp3 = Exp(FPaccum) - Exp(FP)

    movlw       28                      ; If a shift of 28 or more bits required, FP is too small to affect FPaccum
    subwf       (0x7F & B1Temp3),w
    btfsc       status,c
    return

    movlw       0x3                             ; Extract 10-bit mantissa and add leading implied 1's to FPaccum and FP
    andwf       (0x7F & FP),f
    bsf         (0x7F & FP),2
    andwf       (0x7F & FPaccum),f
    bsf         (0x7F & FPaccum),2


    movlw       24                      ; If a shift of 24 or more bits required, shift bytes three places to right
    subwf       (0x7F & B1Temp3),w
    btfss       status,c
    goto        SumP16_002
    movwf       0x7F & B1Temp3
    rlf         (0x7F & FP)+1,f         ; This bit becomes the round-off bit
    movf        (0x7F & FP),w           ; Shift three bytes over
    movwf       (0x7F & FP)+3
    clrf        (0x7F & FP)+2
    clrf        (0x7F & FP)+1
    clrf        (0x7F & FP)
    goto        SumP16_005

SumP16_002
    btfss       (0x7F & B1Temp3),4      ; If a shift of 16-23 bits required, just move bytes
    goto        SumP16_003
    rlf         0x7F & FP+2,f           ; This bit becomes round-off bit
    movf        (0x7F & FP)+1,w         ; Shift two bytes over
    movwf       (0x7F & FP)+3
    movf        (0x7F & FP),w
    movwf       (0x7F & FP)+2
    clrf        (0x7F & FP)+1
    clrf        (0x7F & FP)
    bcf         (0x7F & B1Temp3),4
    goto        SumP16_005

SumP16_003
    btfss       (0x7F & B1Temp3),3      ; If a shift of 8-15 bits required, just move bytes
    goto        SumP16_005
    rlf         0x7F & FP+3,f
    movf        0x7F & FP+2,w
    movwf       0x7F & FP+3
    movf        (0x7F & FP)+1,w
    movwf       (0x7F & FP)+2
    movf        (0x7F & FP),w
    movwf       (0x7F & FP)+1
    clrf        (0x7F & FP)
    bcf         (0x7F & B1Temp3),3

SumP16_005
```

```
    movf        (0x7F & B1Temp3),w                  ; Any more bits to shift?
    btfsc       status,z
    goto        SumP16_007              ; no
SumP16_006
    bcf         status,c            ; Shift FP to right
    rrf         (0x7F & FP),f
    rrf         (0x7F & FP)+1,f
    rrf         (0x7F & FP)+2,f
    rrf         (0x7F & FP)+3,f
    decfsz      (0x7F & B1Temp3),f
    goto        SumP16_006

SumP16_007
    btfss       status,c            ; Do we round up or down?
    goto        SumP16_100              ; Round down
    incf        (0x7F & FP)+3,f
    btfss       status,z
    goto        SumP16_100
    incf        (0x7F & FP)+2,f
    btfss       status,z
    goto        SumP16_100
    incf        (0x7F & FP)+1,f
    btfsc       status,z
    incf        (0x7F & FP),f



SumP16_100                              ; Two values now have same exponent so can be directly added.
    movf        (0x7F & FP)+3,w         ; Add fourth bytes
    addwf       (0x7F & FPaccum)+3,f
    btfss       status,c            ; Is there a carry? If so, ripple carry through remaining bytes.
    goto        SumP16_101
    incf        (0x7F & FPaccum)+2,f
    btfss       status,z
    goto        SumP16_101
    incf        (0x7F & FPaccum)+1,f
    btfsc       status,z
    incf        (0x7F & FPaccum),f

SumP16_101
    movf        (0x7F & FP)+2,w         ; Add Third bytes, and handle carry if necessary.
    addwf       (0x7F & FPaccum)+2,f
    btfss       status,c
    goto        SumP16_102
    incf        (0x7F & FPaccum)+1,f
    btfsc       status,z
    incf        (0x7F & FPaccum),f

SumP16_102
    movf        (0x7F & FP)+1,w         ; Add second bytes, and handle carry if necessary.
    addwf       (0x7F & FPaccum)+1,f
    btfsc       status,c
    incf        (0x7F & FPaccum),f

    movf        (0x7F & FP),w           ; Add first (most significant) bytes
    addwf       (0x7F & FPaccum),f

    btfss       (0x7F & FPaccum),3      ; Has there been a carry to the fourth bit of the most significant byte?
    goto        SumP16_200

SumP16_105
    movlw       0x4                     ; Yes there has been a carry. Renormalize result by shifting to right
    addwf       (0x7F & B1Temp1),f      ; and incrementing the exponent by 1. First do exponent.
    bcf         status,c
    rrf         (0x7F & FPaccum),f
    rrf         (0x7F & FPaccum)+1,f
    rrf         (0x7F & FPaccum)+2,f
    rrf         (0x7F & FPaccum)+3,f
    btfss       status,c            ; If carry, must round result up.
    goto        SumP16_200
    incf        (0x7F & FPaccum)+3,f
    btfss       status,z
    goto        SumP16_200
    incf        (0x7F & FPaccum)+2,f
    btfss       status,z
    goto        SumP16_200
    incf        (0x7F & FPaccum)+1,f
    btfsc       status,z
    incf        (0x7F & FPaccum),f
    btfsc       (0x7F & FPaccum),3      ; Has there been a carry to the fourth bit of the most significant byte?
    goto        SumP16_105              ; Yes, so normalize again.
```

```
SumP16_200
    bcf         (0x7F & FPaccum),2      ; Clear the leading bit which is always 1.
    movf        (0x7F & B1Temp1),w      ; Add exponent to leading byte
    addwf       (0x7F & FPaccum),f
    return



;*************** P16ToFP24 ********************************************************************************
; This routine converts a packed-16 floating point value in FP (Bank1) to a FP24 value in AARG

P16ToFP24
    bsf         status,rp0
    movf        0x7F & FP,w                 ; Get exponent of packed value
    andlw       0xFC
    btfss       status,z                    ; Is exponent, and value, 0?
    goto        $+6
    bcf         status,rp0
    clrf        AARG                        ; Set FP24 value to 0
    clrf        AARG+1
    clrf        AARG+2
    return
    bcf         status,rp0
    movwf       AARG
    bcf         status,c
    rrf         AARG,f                          ; Shift value two bits to right to occupy all of AARG
    rrf         AARG,f
    movlw       100                             ; Now change offset from 27 to 127
    addwf       AARG,f
    bsf         status,rp0                      ; Shift packed mantissa 3 bits to right and place in AARG
    rrf         0x7F & FP,f
    rrf         0x7F & FP+1,f
    rrf         0x7F & FP+2,f
    rrf         0x7F & FP,f
    rrf         0x7F & FP+1,f
    rrf         0x7F & FP+2,f
    bcf         status,c
    rrf         0x7F & FP+1,w
    bcf         status,rp0
    movwf       AARG+1
    bsf         status,rp0
    rrf         0x7F & FP+2,w
    bcf         status,rp0
    movwf       AARG+2
    btfss       status,c                        ; Round up or down?
    return
    incf        AARG+2,f                        ; Round up
    btfss       status,z
    return
    incf        AARG+1,f
    btfss       AARG+1,7
    return
    incf        AARG,f
    clrf        AARG+1
    return



;*********************************************************************
; This routine estimates the square root of the argument x. The argument
; is placed in aarg, and the result is returned in the same register. If the
; result is valid, w will be returned with 0 in it. Otherwise (i.e., argument
; is negative), w is returned containing 0xff.
; The accuracy of the routine is better than 0.2%.

sqrt
    btfsc       aarg+1,7 ; Is argument positive?
    retlw       0xff     ; No--return with 0xff in w
    movlw       0x7f     ; Subtract offset from exponent
    subwf       aarg,w
```

```
    movwf       temp1
    bcf status,c  ; Prepare to do arithmetic right shift of temp1
    btfsc       temp1,7      ; temp 1 positive?
    bsf status,c  ; No--right shift in 1
    rrf temp1,f       ; Divide exponent by 2
    bcf status,c
    rlf temp1,w       ; Multiply by 2--Result always even
    subwf       aarg,f       ; Normalize argument to be >=1 and <4
    MoveBytes aarg,t1,3   ; Temporary save of normalized argument
; Next estimate square root using linear approximation
    StoreFPLiteral 0x7d37f6,barg  ; Load 0.35930, slope of line
    call        FPM24        ; Multiplay by normalized x
    StoreFPLiteral 0x7e2405,barg  ; Load 0.64070, intercept
    call        FPA24
; Now improve estimate as 0.5*(sqrt + x/sqrt)
    MoveBytes aarg,t2,3   ; Save estimate
    MoveBytes aarg,barg,3
    MoveBytes t1,aarg,3   ; Load normalized x
    call        FPD24
    MoveBytes t2,barg,3
    call        FPA24
    decf        aarg,f       ; Divide by 2.
    movf        temp1,w      ; Now multiply by power of 2
    addwf       aarg,f
    retlw       0



;*****************************************************************************************
;*************************************************************************
;

;       PIC16 24 BIT FLOATING POINT LIBRARY
;
;       Place library at beginning of Page 1 of memory



;*****************************************************************************************
;

;       Integer to float conversion
;       Input:  16 bit 2's complement integer right justified in AARGB0, AARGB1
;       Use:    CALL    FLO1624 or      CALL    FLO24
;       Output: 24 bit floating point number in AEXP, AARGB0, AARGB1
;       Result: AARG  <--  FLOAT( AARG )
;       Max Timing:     11+72 = 83 clks         SAT = 0
;                       11+77 = 88 clks         SAT = 1

;       Min Timing:     7+14 = 21 clks                  AARG = 0
;                       7+18 = 25 clks

;       PM: 11+26 = 37                                  DM: 6

;-----------------------------------------------------------------------------------------

FLO1624

FLO24           MOVLW           D'15'+EXPBIAS      ; initialize exponent and add bias
                MOVWF           EXP
                MOVF            AARGB0,W
                MOVWF           SIGN
                BTFSS           AARGB0,MSB     ; test sign
                GOTO            NRM2424
                COMF            AARGB1,F       ; if < 0, negate and set MSB in SIGN
                COMF            AARGB0,F
                INCF            AARGB1,F
                BTFSC           _Z
                INCF            AARGB0,F

;*****************************************************************************************

;       Normalization routine

;       Input:  24 bit unnormalized floating point number in AEXP, AARGB0, AARGB1,
;               with sign in SIGN,MSB and other bits zero.

;       Use:    CALL    NRM2424 or      CALL    NRM24

;       Output: 24 bit normalized floating point number in AEXP, AARGB0, AARGB1

;       Result: AARG  <--  NORMALIZE( AARG )
```

```
;       Max Timing:     10+6+7*7+7 = 72 clks            SAT = 0
;                       10+6+7*7+1+11 = 77 clks SAT = 1

;       Min Timing:     14 clks                         AARG = 0
;                       5+9+4 = 18 clks

;       PM: 26                                          DM: 6

;-----------------------------------------------------------------------------------------

NRM2424
NRM24
        CLRF            TEMP            ; clear exponent decrement
                MOVF            AARGB0,W        ; test if highbyte=0
                BTFSS           _Z
                GOTO            NORM2424
                MOVF            AARGB1,W        ; if so, shift 8 bits by move
                MOVWF           AARGB0
                BTFSC           _Z          ; if highbyte=0, result=0
                GOTO            RES024
                CLRF            AARGB1
                BSF             TEMP,3

NORM2424        MOVF            TEMP,W
                SUBWF           EXP,F
                BTFSS           _Z
                BTFSS           _C
                GOTO            SETFUN24

                BCF             _C          ; clear carry bit

NORM2424A       BTFSC           AARGB0,MSB      ; if MSB=1, normalization done
                GOTO            FIXSIGN24
                RLF             AARGB1,F        ; otherwise, shift left and
                RLF             AARGB0,F        ; decrement EXP
                DECFSZ          EXP,F
                GOTO            NORM2424A

                GOTO            SETFUN24                ; underflow if EXP=0

FIXSIGN24       BTFSS           SIGN,MSB
                BCF             AARGB0,MSB      ; clear explicit MSB if positive
                RETLW           0

RES024          CLRF            AARGB0          ; result equals zero
                CLRF            AARGB1
        CLRF            AARGB2          ; clear extended byte
                CLRF            EXP
                RETLW           0


;*****************************************************************************************
;*****************************************************************************************

;       Integer to float conversion

;       Input:  24 bit 2's complement integer right justified in AARGB0, AARGB1, AARGB2

;       Use:    CALL    FLO2424

;       Output: 24 bit floating point number in AEXP, AARGB0, AARGB1

;       Result: AARG  <--  FLOAT( AARG )

;       Max Timing:     14+94 = 108 clks                RND = 0
;                       14+103 = 117 clks               RND = 1, SAT = 0
;                       14+109 = 123 clks               RND = 1, SAT = 1

;       Min Timing:     6+28 = 34 clks                  AARG = 0
;                       6+22 = 28 clks

;       PM: 14+51 = 65                                  DM: 7

;-----------------------------------------------------------------------------------------

FLO2424         MOVLW           D'23'+EXPBIAS       ; initialize exponent and add bias
                MOVWF           EXP
                CLRF            SIGN
                BTFSS           AARGB0,MSB      ; test sign
                GOTO            NRM3224
                COMF            AARGB2,F        ; if < 0, negate and set MSB in SIGN
```

```
                COMF            AARGB1,F
                COMF            AARGB0,F
                INCF            AARGB2,F
                BTFSC           _Z
                INCF            AARGB1,F
                BTFSC           _Z
                INCF            AARGB0,F
                BSF             SIGN,MSB

;*****************************************************************************************

;       Normalization routine

;       Input:  32 bit unnormalized floating point number in AEXP, AARGB0, AARGB1,
;               AARGB2, with sign in SIGN,MSB

;       Use:    CALL    NRM3224

;       Output: 24 bit normalized floating point number in AEXP, AARGB0, AARGB1

;       Result: AARG  <--  NORMALIZE( AARG )

;       Max Timing:     21+6+7*8+7+4 = 94 clks  RND = 0
;                       21+6+7*8+20+4 = 103 clks        RND = 1, SAT = 0
;                       21+6+7*8+19+11 = 109 clks       RND = 1, SAT = 1

;       Min Timing:     22+6 = 28 clks                  AARG = 0
;                       5+9+4+4 = 22 clks

;       PM: 51                                          DM: 7

;-----------------------------------------------------------------------------------------

NRM3224         CLRF            TEMP                    ; clear exponent decrement
                MOVF            AARGB0,W        ; test if highbyte=0
                BTFSS           _Z
                GOTO            NORM3224
                MOVF            AARGB1,W        ; if so, shift 8 bits by move
                MOVWF           AARGB0
                MOVF            AARGB2,W
                MOVWF           AARGB1
                CLRF            AARGB2
                BSF             TEMP,3                  ; increase decrement by 8

                MOVF            AARGB0,W        ; test if highbyte=0
                BTFSS           _Z
                GOTO            NORM3224
                MOVF            AARGB1,W        ; if so, shift 8 bits by move
                MOVWF           AARGB0
                CLRF            AARGB1
                BCF             TEMP,3                  ; increase decrement by 8
                BSF             TEMP,4

                MOVF            AARGB0,W        ; if highbyte=0, result=0
                BTFSC           _Z
                GOTO            RES024

NORM3224        MOVF            TEMP,W
                SUBWF           EXP,F
                BTFSS           _Z
                BTFSS           _C
                GOTO            SETFUN24

                BCF             _C                      ; clear carry bit

NORM3224A       BTFSC           AARGB0,MSB      ; if MSB=1, normalization done
                GOTO            NRMRND3224
                RLF             AARGB2,F        ; otherwise, shift left and
                RLF             AARGB1,F        ; decrement EXP
                RLF             AARGB0,F
                DECFSZ          EXP,F
                GOTO            NORM3224A
                GOTO            SETFUN24                ; underflow if EXP=0

NRMRND3224      BTFSC           FPFLAGS,RND
                BTFSS           AARGB1,LSB
                GOTO            FIXSIGN24
        BTFSS         AARGB2,MSB        ; round if next bit is set
                GOTO            FIXSIGN24
        INCF          AARGB1,F
                BTFSC           _Z
```

```
                INCF            AARGB0,F

                BTFSS           _Z                      ; has rounding caused carryout?
                GOTO            FIXSIGN24
                RRF             AARGB0,F        ; if so, right shift
                RRF             AARGB1,F
                INCF            EXP,F
                BTFSC           _Z                      ; check for overflow
                GOTO            SETFOV24
                GOTO            FIXSIGN24

;*****************************************************************************************
;*****************************************************************************************

;       Float to integer conversion

;       Input:  24 bit floating point number in AEXP, AARGB0, AARGB1

;       Use:    CALL    INT2416         or      CALL    INT24

;       Output: 16 bit 2's complement integer right justified in AARGB0, AARGB1

;       Result: AARG  <--  INT( AARG )

;       Max Timing:     29+6*6+5+13 = 83 clks           RND = 0
;                       29+6*6+5+19 = 89 clks           RND = 1, SAT = 0
;                       29+6*6+5+22 = 92 clks           RND = 1, SAT = 1

;       Min Timing:     18+5+7 = 30 clks

;       PM: 63                                  DM: 6

;-----------------------------------------------------------------------------------------

INT2416
INT24
        MOVF            EXP,W           ; test for zero argument
        BTFSC           _Z
        RETLW           0x00

        MOVF              AARGB0,W      ; save sign in SIGN
                MOVWF           SIGN
                BSF             AARGB0,MSB      ; make MSB explicit

                MOVLW           EXPBIAS+D'15'       ; remove bias from EXP
                SUBWF           EXP,F
        BTFSS           EXP,MSB
                GOTO            SETIOV16
        COMF            EXP,F
        INCF            EXP,F

                MOVLW           8                       ; do byte shift if EXP >= 8
                SUBWF           EXP,W
                BTFSS           _C
                GOTO            TSHIFT2416
                MOVWF           EXP
                RLF             AARGB1,F        ; rotate next bit for rounding
                MOVF            AARGB0,W
                MOVWF           AARGB1
                CLRF            AARGB0

                MOVLW           8                       ; do byte shift if EXP >= 8
                SUBWF           EXP,W
                BTFSS           _C
                GOTO            TSHIFT2416
                MOVWF           EXP
                RLF             AARGB1,F        ; rotate next bit for rounding
                CLRF            AARGB1
        MOVF            EXP,W
        BTFSS           _Z
        BCF             _C
        GOTO            SHIFT2416OK

TSHIFT2416      MOVF            EXP,W                   ; shift completed if EXP = 0
                BTFSC           _Z
                GOTO            SHIFT2416OK

SHIFT2416       BCF             _C
                RRF             AARGB0,F        ; right shift by EXP
                RRF             AARGB1,F
                DECFSZ          EXP,F
```

```
                GOTO            SHIFT2416

SHIFT2416OK     BTFSC           FPFLAGS,RND
                BTFSS           AARGB1,LSB
                GOTO            INT2416OK
        BTFSS         _C          ; round if next bit is set
                GOTO            INT2416OK
        INCF          AARGB1,F
                BTFSC           _Z
                INCF            AARGB0,F

                BTFSC           AARGB0,MSB      ; test for overflow
                GOTO            SETIOV16

INT2416OK       BTFSS           SIGN,MSB                ; if sign bit set, negate
                RETLW           0
                COMF            AARGB1,F
                COMF            AARGB0,F
                INCF            AARGB1,F
                BTFSC           _Z
                INCF            AARGB0,F
                RETLW           0

SETIOV16        BSF             FPFLAGS,IOV             ; set integer overflow flag
                BTFSS           FPFLAGS,SAT             ; test for saturation
                RETLW           0xFF                    ; return error code in WREG

                CLRF            AARGB0          ; saturate to largest two's
                BTFSS           SIGN,MSB                ; complement 16 bit integer
                MOVLW           0xFF
                MOVWF           AARGB0          ; SIGN = 0, 0x 7F FF
                MOVWF           AARGB1          ; SIGN = 1, 0x 80 00
                RLF             SIGN,F
                RRF             AARGB0,F
                RETLW           0xFF                    ; return error code in WREG


;*****************************************************************************************
;*****************************************************************************************
;

;       Float to integer conversion

;       Input:  24 bit floating point number in AEXP, AARGB0, AARGB1

;       Use:    CALL    INT2424

;       Output: 24 bit 2's complement integer right justified in AARGB0, AARGB1, AARGB2

;       Result: AARG  <--  INT( AARG )

;       Max Timing:     41+6*7+6+16 = 105 clks      RND = 0
;                       41+6*7+6+24 = 113 clks      RND = 1, SAT = 0
;                       41+6*7+6+26 = 115 clks      RND = 1, SAT = 1

;       Min Timing:     5 clks

;       PM: 82                                          DM: 6

;-----------------------------------------------------------------------------------------

INT2424
                CLRF            AARGB2
        MOVF          EXP,W             ; test for zero argument
        BTFSC         _Z
        RETLW         0x00

        MOVF            AARGB0,W        ; save sign in SIGN
                MOVWF           SIGN
                BSF             AARGB0,MSB      ; make MSB explicit

                MOVLW           EXPBIAS+D'23'       ; remove bias from EXP
                SUBWF           EXP,F
                BTFSS           EXP,MSB
                GOTO            SETIOV24
        COMF          EXP,F
        INCF          EXP,F

                MOVLW           8                       ; do byte shift if EXP >= 8
                SUBWF           EXP,W
                BTFSS           _C
```

```
                GOTO            TSHIFT2424
                MOVWF           EXP
                RLF             AARGB2,F        ; rotate next bit for rounding
                MOVF            AARGB1,W
                MOVWF           AARGB2
                MOVF            AARGB0,W
                MOVWF           AARGB1
                CLRF            AARGB0

                MOVLW           8                       ; do another byte shift if EXP >= 8
                SUBWF           EXP,W
                BTFSS           _C
                GOTO            TSHIFT2424
                MOVWF           EXP
                RLF             AARGB2,F        ; rotate next bit for rounding
                MOVF            AARGB1,W
                MOVWF           AARGB2
                CLRF            AARGB1

                MOVLW           8                       ; do another byte shift if EXP >= 8
                SUBWF           EXP,W
                BTFSS           _C
                GOTO            TSHIFT2424
                MOVWF           EXP
                RLF             AARGB2,F        ; rotate next bit for rounding
                CLRF            AARGB2
        MOVF            EXP,W
        BTFSS           _Z
        BCF             _C
        GOTO            SHIFT2424OK

TSHIFT2424      MOVF            EXP,W                   ; shift completed if EXP = 0
                BTFSC           _Z
                GOTO            SHIFT2424OK

SHIFT2424       BCF             _C
                RRF             AARGB0,F        ; right shift by EXP
                RRF             AARGB1,F
                RRF             AARGB2,F
                DECFSZ          EXP,F
                GOTO            SHIFT2424

SHIFT2424OK     BTFSC           FPFLAGS,RND
                BTFSS           AARGB2,LSB
                GOTO            INT2424OK
                BTFSS           _C
                GOTO            INT2424OK
                INCF            AARGB2,F
                BTFSC           _Z
                INCF            AARGB1,F
                BTFSC           _Z
                INCF            AARGB0,F
                BTFSC           AARGB0,MSB      ; test for overflow
                GOTO            SETIOV24

INT2424OK       BTFSS           SIGN,MSB                ; if sign bit set, negate
                RETLW           0
                COMF            AARGB0,F
                COMF            AARGB1,F
                COMF            AARGB2,F
                INCF            AARGB2,F
                BTFSC           _Z
                INCF            AARGB1,F
                BTFSC           _Z
                INCF            AARGB0,F
                RETLW           0

IRES024         CLRF            AARGB0          ; integer result equals zero
                CLRF            AARGB1
                CLRF            AARGB2
                RETLW           0

SETIOV24        BSF             FPFLAGS,IOV             ; set integer overflow flag
                BTFSS           FPFLAGS,SAT             ; test for saturation
                RETLW           0xFF                    ; return error code in WREG

                CLRF            AARGB0          ; saturate to largest two's
                BTFSS           SIGN,MSB                ; complement 24 bit integer
                MOVLW           0xFF
                MOVWF           AARGB0          ; SIGN = 0, 0x 7F FF FF
                MOVWF           AARGB1          ; SIGN = 1, 0x 80 00 00
```

```
                MOVWF           AARGB2
                RLF             SIGN,F
                RRF             AARGB0,F
                RETLW           0xFF                    ; return error code in WREG


;*****************************************************************************

;     Multiply by 10
;
;     This routine multiplies the 24-bit floating point number in
;     aarg by 10.

AargTimes10
    movlw       3
    bcf status,rp0
    addwf       aexp,f          ; Multiply aarg by 3
    clrf        Temp1           ; Set 1 if aarg is negative
    btfsc       aarg+1,7
    incf        Temp1,f
    bsf aarg+1,7      ; Make number positive & insert
                      ; implied MSB
    MoveBytes  aarg+1,barg+1,2    ; Copy fraction to barg
    rrf barg+1,f      ; Shift barg+1, barg+2 right 2 times
    rrf barg+2,f
    bcf barg+1,7
    rrf barg+1,f
    rrf barg+2,w
    bcf barg+1,7
    addwf       aarg+2,f     ; add aarg+1,2 to barg+1,2
    btfsc       status,c
    incf        barg+1,f
    movf        barg+1,w
    addwf       aarg+1,f
    btfss       status,c     ; Skip if carry
    goto        $+4          ; No carry
    rrf aarg+1,f      ; Shift right 1 place
    rrf aarg+2,f      ; and add 1 to exponent
    incf        aexp,f
    bcf aarg+1,7
    btfsc       Temp1,0          ; Is number negative?
    bsf aarg+1,7      ; Yes, set sign bit
    return


;*************************************************************************************************
;*************************************************************************************************

;       Floating Point Multiply

;       Input:  24 bit floating point number in AEXP, AARGB0, AARGB1
;               24 bit floating point number in BEXP, BARGB0, BARGB1

;       Use:    CALL    FPM24

;       Output: 24 bit floating point product in AEXP, AARGB0, AARGB1

;       Result: AARG  <--  AARG * BARG

;       Max Timing:     25+15*16+15+18 = 298 clks       RND = 0
;                       25+15*16+15+29 = 309 clks       RND = 1, SAT = 0
;                       25+15*16+15+33 = 313 clks       RND = 1, SAT = 1

;       Min Timing:     6+5 = 11 clks                   AARG * BARG = 0
;                       24+15*11+14+15 = 218 clks

;       PM: 80                                  DM: 11

;-------------------------------------------------------------------------------------------

FPM24           MOVF            AEXP,W                  ; test for zero arguments
                BTFSS           _Z
                MOVF            BEXP,W
                BTFSC           _Z
                GOTO            RES024

M24BNE0         MOVF            AARGB0,W
                XORWF           BARGB0,W
                MOVWF           SIGN                    ; save sign in SIGN

                MOVF            BEXP,W
```

```
                ADDWF           EXP,F
                MOVLW           EXPBIAS-1
                BTFSS           _C
                GOTO            MTUN24

                SUBWF           EXP,F
                BTFSC           _C
                GOTO            SETFOV24                ; set multiply overflow flag
                GOTO            MOK24

MTUN24          SUBWF           EXP,F
                BTFSS           _C
                GOTO            SETFUN24

MOK24
                MOVF            AARGB0,W
                MOVWF           AARGB2          ; move result to AARG
                MOVF            AARGB1,W
                MOVWF           AARGB3
        BSF             AARGB2,MSB      ; make argument MSB's explicit
                BSF             BARGB0,MSB
                BCF             _C
                CLRF            AARGB0          ; clear initial partial product
                CLRF            AARGB1
                MOVLW           D'16'
                MOVWF           TEMP                    ; initialize counter

MLOOP24         BTFSS           AARGB3,LSB      ; test next bit
                GOTO            MNOADD24

MADD24          MOVF            BARGB1,W
                ADDWF           AARGB1,F
                MOVF            BARGB0,W
                BTFSC           _C
                INCFSZ          BARGB0,W
                ADDWF           AARGB0,F

MNOADD24        RRF             AARGB0,F
                RRF             AARGB1,F
                RRF             AARGB2,F
                RRF             AARGB3,F
                BCF             _C
                DECFSZ          TEMP,F
                GOTO            MLOOP24

                BTFSC           AARGB0,MSB      ; check for postnormalization
                GOTO            MROUND24
                RLF             AARGB2,F
                RLF             AARGB1,F
                RLF             AARGB0,F
                DECF            EXP,F

MROUND24        BTFSC           FPFLAGS,RND
                BTFSS           AARGB1,LSB
                GOTO            MUL24OK
        BTFSS           AARGB2,MSB      ; round if next bit is set
                GOTO            MUL24OK
        INCF            AARGB1,F
                BTFSC           _Z
                INCF            AARGB0,F

                BTFSS           _Z                      ; has rounding caused carryout?
                GOTO            MUL24OK
                RRF             AARGB0,F        ; if so, right shift
                RRF             AARGB1,F
                INCF            EXP,F
                BTFSC           _Z                      ; check for overflow
                GOTO            SETFOV24

MUL24OK         BTFSS           SIGN,MSB
                BCF             AARGB0,MSB      ; clear explicit MSB if positive

                RETLW           0

SETFOV24        BSF             FPFLAGS,FOV             ; set floating point underflag
                BTFSS           FPFLAGS,SAT             ; test for saturation
                RETLW           0xFF                    ; return error code in WREG

                MOVLW           0xFF
                MOVWF           AEXP                    ; saturate to largest floating
                MOVWF           AARGB0          ; point number = 0x FF 7F FF
```

```
                MOVWF           AARGB1          ; modulo the appropriate sign bit
                RLF             SIGN,F
                RRF             AARGB0,F
                RETLW           0xFF                    ; return error code in WREG

;*****************************************************************************************
;*****************************************************************************************

;       Floating Point Divide

;       Input:  24 bit floating point dividend in AEXP, AARGB0, AARGB1
;               24 bit floating point divisor in BEXP, BARGB0, BARGB1

;       Use:    CALL    FPD24

;       Output: 24 bit floating point quotient in AEXP, AARGB0, AARGB1

;       Result: AARG  <--  AARG / BARG

;       Max Timing:     32+13+15*26+25+12 = 472 clks    RND = 0
;                       32+13+15*26+25+34 = 494 clks    RND = 1, SAT = 0
;                       32+13+15*26+25+38 = 498 clks    RND = 1, SAT = 1

;       Min Timing:     7+5 = 12 clks

;       PM: 120                                 DM: 11

;-----------------------------------------------------------------------------------------

FPD24           MOVF            BEXP,W                  ; test for divide by zero
                BTFSC           _Z
                GOTO            SETFDZ24

                MOVF            AEXP,W
                BTFSC           _Z
                GOTO            RES024

D24BNE0         MOVF            AARGB0,W
                XORWF           BARGB0,W
                MOVWF           SIGN                    ; save sign in SIGN
                BSF             AARGB0,MSB      ; make argument MSB's explicit
                BSF             BARGB0,MSB

TALIGN24        CLRF            TEMP                    ; clear align increment
                MOVF            AARGB0,W
                MOVWF           AARGB2          ; test for alignment
                MOVF            AARGB1,W
                MOVWF           AARGB3

                MOVF            BARGB1,W
                SUBWF           AARGB3, f
                MOVF            BARGB0,W
                BTFSS           _C
                INCFSZ          BARGB0,W
                SUBWF           AARGB2, f

                CLRF            AARGB2
                CLRF            AARGB3

                BTFSS           _C
                GOTO            DALIGN24OK

                BCF             _C                      ; align if necessary
                RRF             AARGB0,F
                RRF             AARGB1,F
                RRF             AARGB2,F
                MOVLW           0x01
                MOVWF           TEMP                    ; save align increment

DALIGN24OK      MOVF            BEXP,W                  ; compare AEXP and BEXP
                SUBWF           EXP,F
                BTFSS           _C
                GOTO            ALTB24

AGEB24          MOVLW           EXPBIAS-1
                ADDWF           TEMP,W
                ADDWF           EXP,F
                BTFSC           _C
                GOTO            SETFOV24
                GOTO            DARGOK24                ; set overflow flag
```

```
ALTB24          MOVLW           EXPBIAS-1
                ADDWF           TEMP,W
                ADDWF           EXP,F
                BTFSS           _C
                GOTO            SETFUN24                ; set underflow flag

DARGOK24        MOVLW           D'16'           ; initialize counter
                MOVWF           TEMPB1

DLOOP24         RLF             AARGB3,F        ; left shift
                RLF             AARGB2,F
                RLF             AARGB1,F
                RLF             AARGB0,F
                RLF             TEMP,F

                MOVF            BARGB1,W        ; subtract
                SUBWF           AARGB1,F
                MOVF            BARGB0,W
                BTFSS           _C
                INCFSZ          BARGB0,W
                SUBWF           AARGB0,F

                RLF             BARGB0,W
                IORWF           TEMP,F

                BTFSS           TEMP,LSB                ; test for restore
                GOTO            DREST24

                BSF             AARGB3,LSB
                GOTO            DOK24

DREST24         MOVF            BARGB1,W        ; restore if necessary
                ADDWF           AARGB1,F
                MOVF            BARGB0,W
                BTFSC           _C
                INCF            BARGB0,W
                ADDWF           AARGB0,F
                BCF             AARGB3,LSB

DOK24           DECFSZ          TEMPB1,F
                GOTO            DLOOP24

DROUND24        BTFSC           FPFLAGS,RND
                BTFSS           AARGB3,LSB
                GOTO            DIV24OK
                BCF             _C
                RLF             AARGB1,F        ; compute next significant bit
                RLF             AARGB0,F        ; for rounding
                RLF             TEMP,F

                MOVF            BARGB1,W        ; subtract
                SUBWF           AARGB1,F
                MOVF            BARGB0,W
                BTFSS           _C
                INCFSZ          BARGB0,W
                SUBWF           AARGB0,F

                RLF             BARGB0,W
                IORWF           TEMP,W
                ANDLW           0x01

                ADDWF           AARGB3,F
                BTFSC           _C
                INCF            AARGB2,F

                BTFSS           _Z                      ; test if rounding caused carryout
                GOTO            DIV24OK
                RRF             AARGB2,F
                RRF             AARGB3,F
                INCF            EXP,F
                BTFSC           _Z                      ; test for overflow
                GOTO            SETFOV24

DIV24OK         BTFSS           SIGN,MSB
                BCF             AARGB2,MSB      ; clear explicit MSB if positive

                MOVF            AARGB2,W
                MOVWF           AARGB0          ; move result to AARG
                MOVF            AARGB3,W
                MOVWF           AARGB1
```

```
                RETLW           0

SETFUN24        BSF             FPFLAGS,FUN             ; set floating point underflag
                BTFSS           FPFLAGS,SAT             ; test for saturation
                RETLW           0xFF                    ; return error code in WREG

                MOVLW           0x01                    ; saturate to smallest floating
                MOVWF           AEXP                    ; point number = 0x 01 00 00
                CLRF            AARGB0          ; modulo the appropriate sign bit
                CLRF            AARGB1
                RLF             SIGN,F
                RRF             AARGB0,F
                RETLW           0xFF                    ; return error code in WREG

SETFDZ24        BSF             FPFLAGS,FDZ             ; set divide by zero flag
                RETLW           0xFF

;*****************************************************************************************
;*****************************************************************************************

;       Floating Point Subtract

;       Input:  24 bit floating point number in AEXP, AARGB0, AARGB1
;               24 bit floating point number in BEXP, BARGB0, BARGB1

;       Use:    CALL FPS24

;       Output: 24 bit floating point sum in AEXP, AARGB0, AARGB1

;       Result: AARG  <--  AARG - BARG

;       Max Timing:     2+197 = 199 clks                RND = 0
;                       2+208 = 210 clks                RND = 1, SAT = 0
;                       2+213 = 215 clks                RND = 1, SAT = 1

;       Min Timing:     2+12 = 14 clks

;       PM: 2+112 = 114                         DM: 11

;-----------------------------------------------------------------------------------------

FPS24           MOVLW           0x80
                XORWF           BARGB0,F

;*****************************************************************************************

;       Floating Point Add

;       Input:  24 bit floating point number in AEXP, AARGB0, AARGB1
;               24 bit floating point number in BEXP, BARGB0, BARGB1

;       Use:    CALL FPA24

;       Output: 24 bit floating point sum in AEXP, AARGB0, AARGB1

;       Result: AARG  <--  AARG - BARG

;       Max Timing:     25+28+6*6+5+31+72 = 197 clks            RND = 0
;                       25+28+6*6+5+42+72 = 208 clks            RND = 1, SAT = 0
;                       25+28+6*6+5+42+77 = 213 clks            RND = 1, SAT = 1

;       Min Timing:     8+4 = 12 clks

;       PM: 112                                                 DM: 11

;-----------------------------------------------------------------------------------------

FPA24           MOVF            AARGB0,W        ; exclusive or of signs in TEMP
                XORWF           BARGB0,W
                MOVWF           TEMP

        CLRF            AARGB2          ; clear extended byte
        CLRF            BARGB2

                MOVF            AEXP,W                  ; use AARG if AEXP >= BEXP
                SUBWF           BEXP,W
                BTFSS           _C
                GOTO            USEA24

                MOVF            BEXP,W                  ; use BARG if AEXP < BEXP
                MOVWF           AARGB4          ; therefore, swap AARG and BARG
```

```
                MOVF            AEXP,W
                MOVWF           BEXP
                MOVF            AARGB4,W
                MOVWF           AEXP

                MOVF            BARGB0,W
                MOVWF           AARGB4
                MOVF            AARGB0,W
                MOVWF           BARGB0
                MOVF            AARGB4,W
                MOVWF           AARGB0

                MOVF            BARGB1,W
                MOVWF           AARGB4
                MOVF            AARGB1,W
                MOVWF           BARGB1
                MOVF            AARGB4,W
                MOVWF           AARGB1

USEA24          MOVF            BEXP,W                  ; return AARG if BARG = 0
                BTFSC           _Z
                RETLW           0x00

                MOVF            AARGB0,W
                MOVWF           SIGN                    ; save sign in SIGN
                BSF             AARGB0,MSB      ; make MSB's explicit
                BSF             BARGB0,MSB

                MOVF            BEXP,W                  ; compute shift count in BEXP
                SUBWF           AEXP,W
                MOVWF           BEXP
                BTFSC           _Z
                GOTO            ALIGNED24

                MOVLW           8
                SUBWF           BEXP,W
                BTFSS           _C                      ; if BEXP >= 8, do byte shift
                GOTO            ALIGNB24
                MOVWF           BEXP
                MOVF            BARGB1,W        ; keep for postnormalization
        MOVWF           BARGB2
                MOVF            BARGB0,W
        MOVWF           BARGB1
                CLRF            BARGB0

                MOVLW           8
                SUBWF           BEXP,W
                BTFSS           _C                      ; if BEXP >= 8, BARG = 0 relative to AARG
                GOTO            ALIGNB24
                MOVF            SIGN,W
                MOVWF           AARGB0
                RETLW           0x00

ALIGNB24        MOVF            BEXP,W                  ; already aligned if BEXP = 0
                BTFSC           _Z
                GOTO            ALIGNED24

ALOOPB24        BCF             _C                      ; right shift by BEXP
                RRF             BARGB0,F
                RRF             BARGB1,F
        RRF             BARGB2,F
                DECFSZ          BEXP,F
                GOTO            ALOOPB24

ALIGNED24       BTFSS           TEMP,MSB                ; negate if signs opposite
                GOTO            AOK24
        COMF            BARGB2,F
                COMF            BARGB1,F
                COMF            BARGB0,F
                INCF            BARGB2,F
                BTFSC           _Z
                INCF            BARGB1,F
        BTFSC           _Z
        INCF            BARGB0,F

AOK24
                MOVF      BARGB2,W
                ADDWF     AARGB2,F
                MOVF            BARGB1,W
                BTFSC           _C
                INCFSZ          BARGB1,W
```

```
                ADDWF           AARGB1,F
                MOVF            BARGB0,W
                BTFSC           _C
                INCFSZ          BARGB0,W
                ADDWF           AARGB0,F

                BTFSC           TEMP,MSB
                GOTO            ACOMP24
                BTFSS           _C
                GOTO            NRMRND3224

                RRF             AARGB0,F        ; shift right and increment EXP
                RRF             AARGB1,F
        RRF             AARGB2,F
                INCFSZ          AEXP,F
                GOTO            NRMRND3224
                GOTO            SETFOV24

ACOMP24         BTFSC           _C
                GOTO            NRM3224             ; normalize and fix sign

        COMF            AARGB2,F
                COMF            AARGB1,F        ; negate, toggle sign bit and
                COMF            AARGB0,F        ; then normalize
                INCF            AARGB2,F
                BTFSC           _Z
                INCF            AARGB1,F
                BTFSC           _Z
                INCF            AARGB0,F

                MOVLW           0x80
                XORWF           SIGN,F
                GOTO            NRM24

;*****************************************************************************************
;*****************************************************************************************
;

;   Evaluate floor(x)

;   Input:      24 bit floating point number in AEXP, AARGB0, AARGB1

;   Use:        CALL FLOOR24

;   Output:     24 bit floating point number in AEXP, AARGB0, AARGB1

;   Result:     AARG  <--  FLOOR( AARG )

;   Testing on [-MAXNUM,MAXNUM] from 100000 trials:

;       min     max  mean
;   Timing:     37   55   42.7 clks

;       min     max  mean rms
;   Error:      0x00 0x00 0.0  0.0  nsb

;-----------------------------------------------------------------------------------------

;   floor(x) evaluates the largest integer, as a float, not greater than x.

FLOOR24
        CLRF            AARGB2          ; test for zero argument
        MOVF            AEXP,W
        BTFSC           _Z
        RETLW           0x00

        MOVF            AARGB0,W
        MOVWF           AARGB3          ; save mantissa
        MOVF            AARGB1,W
        MOVWF           AARGB4

        MOVLW           EXPBIAS         ; computed unbiased exponent
        SUBWF           AEXP,W
        MOVWF           TEMPB1
        BTFSC           TEMPB1,MSB
        GOTO            FLOOR24ZERO

        SUBLW           0x10-1
        MOVWF           TEMPB0          ; save number of zero bits in TEMPB0
        MOVWF           TEMPB1

        BTFSC           TEMPB1,LSB+3    ; divide by eight
```

```
        GOTO          FLOOR24MASKH

FLOOR24MASKL
        MOVLW         0x07         ; get remainder for mask pointer
        ANDWF         TEMPB0,F
        MOVLW         LOW FLOOR24MASKTABLE
        ADDWF         TEMPB0,F
        MOVLW         HIGH FLOOR24MASKTABLE
        BTFSC         _C
        ADDLW         0x01
        MOVWF         PCLATH
        INCF          TEMPB0,W

        CALL          FLOOR24MASKTABLE ; access table for mask

        ANDWF         AARGB1,F
        BTFSS         AARGB0,MSB       ; if negative, round down
        RETLW         0x00

        MOVWF         AARGB7
        MOVF          AARGB4,W
        SUBWF         AARGB1,W
        BTFSS         _Z
        GOTO          FLOOR24RNDL
        RETLW         0x00

FLOOR24RNDL
        COMF          AARGB7,W
        MOVWF         TEMPB1
        INCF          TEMPB1,W
        ADDWF         AARGB1,F
                BTFSC           _Z
                INCF            AARGB0, F
        BTFSS         _Z           ; has rounding caused carryout?
        RETLW         0x00
        RRF           AARGB0,F
        RRF           AARGB1,F
        INCFSZ        AEXP,F           ; check for overflow
        RETLW         0x00
        GOTO          SETFOV24

FLOOR24MASKH
        MOVLW         0x07         ; get remainder for mask pointer
        ANDWF         TEMPB0,F
        MOVLW         LOW FLOOR24MASKTABLE
        ADDWF         TEMPB0,F
        MOVLW         HIGH FLOOR24MASKTABLE
        BTFSC         _C
        ADDLW         0x01
        MOVWF         PCLATH
        INCF          TEMPB0,W

        CALL          FLOOR24MASKTABLE ; access table for mask

        ANDWF         AARGB0,F
        CLRF          AARGB1
        BTFSS         AARGB0,MSB       ; if negative, round down
        RETLW         0x00

        MOVWF         AARGB7
        MOVF          AARGB4,W
        SUBWF         AARGB1,W
        BTFSS         _Z
        GOTO          FLOOR24RNDH
        MOVF          AARGB3,W
        SUBWF         AARGB0,W
        BTFSS         _Z
        GOTO          FLOOR24RNDH
        RETLW         0x00

FLOOR24RNDH
        COMF          AARGB7,W
        MOVWF         TEMPB1
        INCF          TEMPB1,W
        ADDWF         AARGB0,F
        BTFSS         _C           ; has rounding caused carryout?
        RETLW         0x00
        RRF           AARGB0,F
        RRF           AARGB1,F
        INCFSZ        AEXP,F
        RETLW         0x00
```

```
        GOTO          SETFOV24     ; check for overflow

FLOOR24ZERO
        BTFSC         AARGB0,MSB
        GOTO          FLOOR24MINUSONE
        CLRF          AEXP
        CLRF          AARGB0
        CLRF          AARGB1
        RETLW         0x00

FLOOR24MINUSONE
        MOVLW         0x7F
        MOVWF         AEXP
        MOVLW         0x80
        MOVWF         AARGB0
        CLRF          AARGB1
        RETLW         0x00

;-----------------------------------------------------------------------------------------

;   table for least significant byte requiring masking, using pointer from
;   the remainder of the number of zero bits divided by eight.

FLOOR24MASKTABLE
        MOVWF         PCL
        RETLW         0xFF
        RETLW         0xFE
        RETLW         0xFC
        RETLW         0xF8
        RETLW         0xF0
        RETLW         0xE0
        RETLW         0xC0
        RETLW         0x80
        RETLW         0x00

;*****************************************************************************************
;*****************************************************************************************

;   Floating Point Relation   A < B

;   Input:       24 bit floating point number in AEXP, AARGB0, AARGB1
;       24 bit floating point number in BEXP, BARGB0, BARGB1

;   Use:         CALL TALTB24

;   Output:      logical result in W

;   Result:      if A < B TRUE,    W = 0x01
;       if A < B FALSE, W = 0x00

;   Testing on [-MAXNUM,MAXNUM] from 100000 trials:

;       min      max  mean
;   Timing:      9    28   14.6 clks

TALTB24         MOVF     AARGB0,W     ; test if signs opposite
        XORWF         BARGB0,W
        MOVWF         TEMPB0
        BTFSC         TEMPB0,MSB
        GOTO          TALTB24O

        BTFSC         AARGB0,MSB
        GOTO          TALTB24N

TALTB24P        MOVF     AEXP,W           ; compare positive arguments
        SUBWF         BEXP,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          AARGB0,W
        SUBWF         BARGB0,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          AARGB1,W
        SUBWF         BARGB1,W
        BTFSS         _C
```

```
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01
        RETLW         0x00

TALTB24N        MOVF     BEXP,W           ; compare negative arguments
        SUBWF         AEXP,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          BARGB0,W
        SUBWF         AARGB0,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          BARGB1,W
        SUBWF         AARGB1,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01
        RETLW         0x00

TALTB24O        BTFSS        BARGB0,MSB
        RETLW         0x01
        RETLW         0x00

;*****************************************************************************************
;*****************************************************************************************

;   Floating Point Relation   A <= B

;   Input:       24 bit floating point number in AEXP, AARGB0, AARGB1
;       24 bit floating point number in BEXP, BARGB0, BARGB1

;   Use:         CALL TALEB24

;   Output:      logical result in W

;   Result:      if A <= B TRUE,    W = 0x01
;       if A <= B FALSE, W = 0x00

;   Testing on [-MAXNUM,MAXNUM] from 100000 trials:

;       min       max  mean
;   Timing:      9    26   14.4 clks

TALEB24         MOVF     AARGB0,W     ; test if signs opposite
        XORWF         BARGB0,W
        MOVWF         TEMPB0
        BTFSC         TEMPB0,MSB
        GOTO          TALEB24O

        BTFSC         AARGB0,MSB
        GOTO          TALEB24N

TALEB24P        MOVF     AEXP,W           ; compare positive arguments
        SUBWF         BEXP,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          AARGB0,W
        SUBWF         BARGB0,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          AARGB1,W
        SUBWF         BARGB1,W
        BTFSS         _C
        RETLW         0x00
        RETLW         0x01

TALEB24N        MOVF     BEXP,W           ; compare negative arguments
```

```
        SUBWF         AEXP,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          BARGB0,W
        SUBWF         AARGB0,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          BARGB1,W
        SUBWF         AARGB1,W
        BTFSS         _C
        RETLW         0x00
        RETLW         0x01

TALEB24O         BTFSS        BARGB0,MSB
        RETLW         0x01
        RETLW         0x00

;*****************************************************************************************
;*****************************************************************************************

;   Floating Point Relation   A > B

;   Input:       24 bit floating point number in AEXP, AARGB0, AARGB1
;       24 bit floating point number in BEXP, BARGB0, BARGB1

;   Use:         CALL TAGTB24

;   Output:      logical result in W

;   Result:      if A > B TRUE,    W = 0x01
;       if A > B FALSE, W = 0x00

;   Testing on [-MAXNUM,MAXNUM] from 100000 trials:

;       min      max  mean
;   Timing:      9    28   14.6 clks

TAGTB24          MOVF     BARGB0,W     ; test if signs opposite
        XORWF         AARGB0,W
        MOVWF         TEMPB0
        BTFSC         TEMPB0,MSB
        GOTO          TAGTB24O

        BTFSC         BARGB0,MSB
        GOTO          TAGTB24N

TAGTB24P         MOVF     BEXP,W           ; compare positive arguments
        SUBWF         AEXP,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          BARGB0,W
        SUBWF         AARGB0,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          BARGB1,W
        SUBWF         AARGB1,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01
        RETLW         0x00

TAGTB24N         MOVF     AEXP,W           ; compare negative arguments
        SUBWF         BEXP,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01
```

```
        MOVF          AARGB0,W
        SUBWF         BARGB0,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          AARGB1,W
        SUBWF         BARGB1,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01
        RETLW         0x00

TAGTB24O         BTFSS        AARGB0,MSB
        RETLW         0x01
        RETLW         0x00

;*****************************************************************************************
;*****************************************************************************************

;   Floating Point Relation   A >= B

;   Input:       24 bit floating point number in AEXP, AARGB0, AARGB1
;       24 bit floating point number in BEXP, BARGB0, BARGB1

;   Use:         CALL TAGEB24

;   Output:      logical result in W

;   Result:      if A >= B TRUE,   W = 0x01
;       if A >= B FALSE, W = 0x00

;   Testing on [-MAXNUM,MAXNUM] from 100000 trials:

;       min       max  mean
;   Timing:      9    26   14.4 clks

TAGEB24          MOVF     BARGB0,W     ; test if signs opposite
        XORWF         AARGB0,W
        MOVWF         TEMPB0
        BTFSC         TEMPB0,MSB
        GOTO          TAGEB24O

        BTFSC         BARGB0,MSB
        GOTO          TAGEB24N

TAGEB24P         MOVF     BEXP,W           ; compare positive arguments
        SUBWF         AEXP,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          BARGB0,W
        SUBWF         AARGB0,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          BARGB1,W
        SUBWF         AARGB1,W
        BTFSS         _C
        RETLW         0x00
        RETLW         0x01

TAGEB24N         MOVF     AEXP,W           ; compare negative arguments
        SUBWF         BEXP,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01

        MOVF          AARGB0,W
        SUBWF         BARGB0,W
        BTFSS         _C
        RETLW         0x00
        BTFSS         _Z
        RETLW         0x01
```

```
        MOVF          AARGB1,W
        SUBWF         BARGB1,W
        BTFSS         _C
        RETLW         0x00
        RETLW         0x01

TAGEB24O         BTFSS        AARGB0,MSB
        RETLW         0x01
        RETLW         0x00

;*********************************************************************************************
;*********************************************************************************************

;   Floating Point Relation   A == B

;   Input:       24 bit floating point number in AEXP, AARGB0, AARGB1
;       24 bit floating point number in BEXP, BARGB0, BARGB1

;   Use:         CALL TAEQB24

;   Output:      logical result in W

;   Result:      if A == B TRUE,    W = 0x01
;       if A == B FALSE, W = 0x00

;   Testing on [-MAXNUM,MAXNUM] from 100000 trials:

;       min       max  mean
;   Timing:      5    14   6.9 clks

TAEQB24          MOVF     BEXP,W
        SUBWF         AEXP,W
        BTFSS         _Z
        RETLW         0x00

        MOVF          BARGB0,W
        SUBWF         AARGB0,W
        BTFSS         _Z
        RETLW         0x00

        MOVF          BARGB1,W
        SUBWF         AARGB1,W
        BTFSS         _Z
        RETLW         0x00
        RETLW         0x01


;*********************************************************************************************
;*********************************************************************************************

;   Floating Point Relation   A =! B

;   Input:       24 bit floating point number in AEXP, AARGB0, AARGB1

;       24 bit floating point number in BEXP, BARGB0, BARGB1

;   Use:         CALL TANEB24

;   Output:      logical result in W

;   Result:      if A =! B TRUE,    W = 0x01
;       if A =! B FALSE, W = 0x00

;   Testing on [-MAXNUM,MAXNUM] from 100000 trials:

;       min       max  mean
;   Timing:      5    14   6.9 clks

TANEB24          MOVF     BEXP,W
        SUBWF         AEXP,W
        BTFSS         _Z
        RETLW         0x01

        MOVF          BARGB0,W
        SUBWF         AARGB0,W
        BTFSS         _Z
        RETLW         0x01

        MOVF          BARGB1,W
        SUBWF         AARGB1,W
```

```
        BTFSS         _Z
        RETLW         0x01
        RETLW         0x00


    org          0x1000



;*****************************************************
; Lookup Table for Forward (Channel 0) Power
;   Calibration dBm = .10063*ADC + -40.83


FLowByteTable0
     retlw      0x8B            ; Low Byte, ADRES = 0, Forward Power = 82.60 nW
     retlw      0x15            ; Low Byte, ADRES = 4, Forward Power = 90.63 nW
     retlw      0xAC            ; Low Byte, ADRES = 8, Forward Power = 99.43 nW
     retlw      0x52            ; Low Byte, ADRES = 12, Forward Power = 109.1 nW
     retlw      0x04            ; Low Byte, ADRES = 16, Forward Power = 119.7 nW
     retlw      0x68            ; Low Byte, ADRES = 20, Forward Power = 131.3 nW
     retlw      0xD5            ; Low Byte, ADRES = 24, Forward Power = 144.0 nW
     retlw      0x4E            ; Low Byte, ADRES = 28, Forward Power = 158.0 nW
     retlw      0xD1            ; Low Byte, ADRES = 32, Forward Power = 173.4 nW
     retlw      0x62            ; Low Byte, ADRES = 36, Forward Power = 190.2 nW
     retlw      0x01            ; Low Byte, ADRES = 40, Forward Power = 208.7 nW
     retlw      0xAF            ; Low Byte, ADRES = 44, Forward Power = 229.0 nW
     retlw      0x37            ; Low Byte, ADRES = 48, Forward Power = 251.2 nW
     retlw      0xA0            ; Low Byte, ADRES = 52, Forward Power = 275.6 nW
     retlw      0x13            ; Low Byte, ADRES = 56, Forward Power = 302.4 nW
     retlw      0x91            ; Low Byte, ADRES = 60, Forward Power = 331.7 nW
     retlw      0x1B            ; Low Byte, ADRES = 64, Forward Power = 363.9 nW
     retlw      0xB3            ; Low Byte, ADRES = 68, Forward Power = 399.3 nW
     retlw      0x59            ; Low Byte, ADRES = 72, Forward Power = 438.1 nW
     retlw      0x08            ; Low Byte, ADRES = 76, Forward Power = 480.6 nW
     retlw      0x6C            ; Low Byte, ADRES = 80, Forward Power = 527.3 nW
     retlw      0xDA            ; Low Byte, ADRES = 84, Forward Power = 578.5 nW
     retlw      0x53            ; Low Byte, ADRES = 88, Forward Power = 634.7 nW
     retlw      0xD7            ; Low Byte, ADRES = 92, Forward Power = 696.3 nW
     retlw      0x69            ; Low Byte, ADRES = 96, Forward Power = 763.9 nW
     retlw      0x08            ; Low Byte, ADRES = 100, Forward Power = 838.1 nW
     retlw      0xB7            ; Low Byte, ADRES = 104, Forward Power = 919.5 nW
     retlw      0x3B            ; Low Byte, ADRES = 108, Forward Power = 1.009 uW
     retlw      0xA4            ; Low Byte, ADRES = 112, Forward Power = 1.107 uW
     retlw      0x18            ; Low Byte, ADRES = 116, Forward Power = 1.214 uW
     retlw      0x96            ; Low Byte, ADRES = 120, Forward Power = 1.332 uW
     retlw      0x21            ; Low Byte, ADRES = 124, Forward Power = 1.462 uW
     retlw      0xBA            ; Low Byte, ADRES = 128, Forward Power = 1.603 uW
     retlw      0x61            ; Low Byte, ADRES = 132, Forward Power = 1.759 uW
     retlw      0x0C            ; Low Byte, ADRES = 136, Forward Power = 1.930 uW
     retlw      0x71            ; Low Byte, ADRES = 140, Forward Power = 2.117 uW
     retlw      0xDF            ; Low Byte, ADRES = 144, Forward Power = 2.323 uW
     retlw      0x58            ; Low Byte, ADRES = 148, Forward Power = 2.549 uW
     retlw      0xDD            ; Low Byte, ADRES = 152, Forward Power = 2.796 uW
     retlw      0x6F            ; Low Byte, ADRES = 156, Forward Power = 3.068 uW
     retlw      0x0F            ; Low Byte, ADRES = 160, Forward Power = 3.366 uW
     retlw      0xBE            ; Low Byte, ADRES = 164, Forward Power = 3.693 uW
     retlw      0x40            ; Low Byte, ADRES = 168, Forward Power = 4.051 uW
     retlw      0xA9            ; Low Byte, ADRES = 172, Forward Power = 4.445 uW
     retlw      0x1D            ; Low Byte, ADRES = 176, Forward Power = 4.876 uW
     retlw      0x9C            ; Low Byte, ADRES = 180, Forward Power = 5.350 uW
     retlw      0x28            ; Low Byte, ADRES = 184, Forward Power = 5.869 uW
     retlw      0xC1            ; Low Byte, ADRES = 188, Forward Power = 6.439 uW
     retlw      0x68            ; Low Byte, ADRES = 192, Forward Power = 7.065 uW
     retlw      0x10            ; Low Byte, ADRES = 196, Forward Power = 7.751 uW
     retlw      0x75            ; Low Byte, ADRES = 200, Forward Power = 8.504 uW
     retlw      0xE4            ; Low Byte, ADRES = 204, Forward Power = 9.329 uW
     retlw      0x5E            ; Low Byte, ADRES = 208, Forward Power = 10.24 uW
     retlw      0xE3            ; Low Byte, ADRES = 212, Forward Power = 11.23 uW
     retlw      0x76            ; Low Byte, ADRES = 216, Forward Power = 12.32 uW
     retlw      0x16            ; Low Byte, ADRES = 220, Forward Power = 13.52 uW
     retlw      0xC6            ; Low Byte, ADRES = 224, Forward Power = 14.83 uW
     retlw      0x44            ; Low Byte, ADRES = 228, Forward Power = 16.27 uW
     retlw      0xAE            ; Low Byte, ADRES = 232, Forward Power = 17.85 uW
     retlw      0x22            ; Low Byte, ADRES = 236, Forward Power = 19.58 uW
     retlw      0xA2            ; Low Byte, ADRES = 240, Forward Power = 21.48 uW
     retlw      0x2E            ; Low Byte, ADRES = 244, Forward Power = 23.57 uW
     retlw      0xC7            ; Low Byte, ADRES = 248, Forward Power = 25.86 uW
     retlw      0x70            ; Low Byte, ADRES = 252, Forward Power = 28.37 uW
```

```
retlw      0x14            ; Low Byte, ADRES = 256, Forward Power = 31.13 uW
retlw      0x7A            ; Low Byte, ADRES = 260, Forward Power = 34.15 uW
retlw      0xE9            ; Low Byte, ADRES = 264, Forward Power = 37.47 uW
retlw      0x63            ; Low Byte, ADRES = 268, Forward Power = 41.10 uW
retlw      0xE9            ; Low Byte, ADRES = 272, Forward Power = 45.10 uW
retlw      0x7C            ; Low Byte, ADRES = 276, Forward Power = 49.48 uW
retlw      0x1D            ; Low Byte, ADRES = 280, Forward Power = 54.28 uW
retlw      0xCE            ; Low Byte, ADRES = 284, Forward Power = 59.55 uW
retlw      0x48            ; Low Byte, ADRES = 288, Forward Power = 65.33 uW
retlw      0xB3            ; Low Byte, ADRES = 292, Forward Power = 71.68 uW
retlw      0x27            ; Low Byte, ADRES = 296, Forward Power = 78.64 uW
retlw      0xA8            ; Low Byte, ADRES = 300, Forward Power = 86.28 uW
retlw      0x34            ; Low Byte, ADRES = 304, Forward Power = 94.66 uW
retlw      0xCE            ; Low Byte, ADRES = 308, Forward Power = 103.8 uW
retlw      0x78            ; Low Byte, ADRES = 312, Forward Power = 113.9 uW
retlw      0x19            ; Low Byte, ADRES = 316, Forward Power = 125.0 uW
retlw      0x7E            ; Low Byte, ADRES = 320, Forward Power = 137.1 uW
retlw      0xEE            ; Low Byte, ADRES = 324, Forward Power = 150.5 uW
retlw      0x69            ; Low Byte, ADRES = 328, Forward Power = 165.1 uW
retlw      0xEF            ; Low Byte, ADRES = 332, Forward Power = 181.1 uW
retlw      0x83            ; Low Byte, ADRES = 336, Forward Power = 198.7 uW
retlw      0x25            ; Low Byte, ADRES = 340, Forward Power = 218.0 uW
retlw      0xD6            ; Low Byte, ADRES = 344, Forward Power = 239.2 uW
retlw      0x4D            ; Low Byte, ADRES = 348, Forward Power = 262.4 uW
retlw      0xB7            ; Low Byte, ADRES = 352, Forward Power = 287.9 uW
retlw      0x2D            ; Low Byte, ADRES = 356, Forward Power = 315.8 uW
retlw      0xAD            ; Low Byte, ADRES = 360, Forward Power = 346.5 uW
retlw      0x3A            ; Low Byte, ADRES = 364, Forward Power = 380.1 uW
retlw      0xD5            ; Low Byte, ADRES = 368, Forward Power = 417.0 uW
retlw      0x7F            ; Low Byte, ADRES = 372, Forward Power = 457.5 uW
retlw      0x1D            ; Low Byte, ADRES = 376, Forward Power = 502.0 uW
retlw      0x83            ; Low Byte, ADRES = 380, Forward Power = 550.7 uW
retlw      0xF3            ; Low Byte, ADRES = 384, Forward Power = 604.2 uW
retlw      0x6E            ; Low Byte, ADRES = 388, Forward Power = 662.9 uW
retlw      0xF5            ; Low Byte, ADRES = 392, Forward Power = 727.3 uW
retlw      0x89            ; Low Byte, ADRES = 396, Forward Power = 797.9 uW
retlw      0x2C            ; Low Byte, ADRES = 400, Forward Power = 875.4 uW
retlw      0xDE            ; Low Byte, ADRES = 404, Forward Power = 960.4 uW
retlw      0x51            ; Low Byte, ADRES = 408, Forward Power = 1.054 mW
retlw      0xBC            ; Low Byte, ADRES = 412, Forward Power = 1.156 mW
retlw      0x32            ; Low Byte, ADRES = 416, Forward Power = 1.268 mW
retlw      0xB3            ; Low Byte, ADRES = 420, Forward Power = 1.391 mW
retlw      0x41            ; Low Byte, ADRES = 424, Forward Power = 1.527 mW
retlw      0xDC            ; Low Byte, ADRES = 428, Forward Power = 1.675 mW
retlw      0x87            ; Low Byte, ADRES = 432, Forward Power = 1.837 mW
retlw      0x21            ; Low Byte, ADRES = 436, Forward Power = 2.016 mW
retlw      0x88            ; Low Byte, ADRES = 440, Forward Power = 2.212 mW
retlw      0xF8            ; Low Byte, ADRES = 444, Forward Power = 2.426 mW
retlw      0x74            ; Low Byte, ADRES = 448, Forward Power = 2.662 mW
retlw      0xFB            ; Low Byte, ADRES = 452, Forward Power = 2.921 mW
retlw      0x90            ; Low Byte, ADRES = 456, Forward Power = 3.204 mW
retlw      0x33            ; Low Byte, ADRES = 460, Forward Power = 3.515 mW
retlw      0xE6            ; Low Byte, ADRES = 464, Forward Power = 3.857 mW
retlw      0x55            ; Low Byte, ADRES = 468, Forward Power = 4.231 mW
retlw      0xC1            ; Low Byte, ADRES = 472, Forward Power = 4.642 mW
retlw      0x37            ; Low Byte, ADRES = 476, Forward Power = 5.093 mW
retlw      0xB9            ; Low Byte, ADRES = 480, Forward Power = 5.588 mW
retlw      0x47            ; Low Byte, ADRES = 484, Forward Power = 6.130 mW
retlw      0xE3            ; Low Byte, ADRES = 488, Forward Power = 6.726 mW
retlw      0x8E            ; Low Byte, ADRES = 492, Forward Power = 7.379 mW
retlw      0x25            ; Low Byte, ADRES = 496, Forward Power = 8.096 mW
retlw      0x8C            ; Low Byte, ADRES = 500, Forward Power = 8.882 mW
retlw      0xFD            ; Low Byte, ADRES = 504, Forward Power = 9.744 mW
retlw      0x79            ; Low Byte, ADRES = 508, Forward Power = 10.69 mW
retlw      0x01            ; Low Byte, ADRES = 512, Forward Power = 11.73 mW
retlw      0x97            ; Low Byte, ADRES = 516, Forward Power = 12.87 mW
retlw      0x3A            ; Low Byte, ADRES = 520, Forward Power = 14.12 mW
retlw      0xEE            ; Low Byte, ADRES = 524, Forward Power = 15.49 mW
retlw      0x5A            ; Low Byte, ADRES = 528, Forward Power = 16.99 mW
retlw      0xC6            ; Low Byte, ADRES = 532, Forward Power = 18.64 mW
retlw      0x3C            ; Low Byte, ADRES = 536, Forward Power = 20.45 mW
retlw      0xBF            ; Low Byte, ADRES = 540, Forward Power = 22.44 mW
retlw      0x4D            ; Low Byte, ADRES = 544, Forward Power = 24.62 mW
retlw      0xEA            ; Low Byte, ADRES = 548, Forward Power = 27.01 mW
retlw      0x96            ; Low Byte, ADRES = 552, Forward Power = 29.63 mW
retlw      0x29            ; Low Byte, ADRES = 556, Forward Power = 32.51 mW
retlw      0x91            ; Low Byte, ADRES = 560, Forward Power = 35.67 mW
retlw      0x02            ; Low Byte, ADRES = 564, Forward Power = 39.13 mW
retlw      0x7F            ; Low Byte, ADRES = 568, Forward Power = 42.93 mW
retlw      0x07            ; Low Byte, ADRES = 572, Forward Power = 47.10 mW
retlw      0x9D            ; Low Byte, ADRES = 576, Forward Power = 51.68 mW
```

```
retlw      0x42            ; Low Byte, ADRES = 580, Forward Power = 56.69 mW
retlw      0xF6            ; Low Byte, ADRES = 584, Forward Power = 62.20 mW
retlw      0x5E            ; Low Byte, ADRES = 588, Forward Power = 68.24 mW
retlw      0xCB            ; Low Byte, ADRES = 592, Forward Power = 74.87 mW
retlw      0x42            ; Low Byte, ADRES = 596, Forward Power = 82.14 mW
retlw      0xC4            ; Low Byte, ADRES = 600, Forward Power = 90.12 mW
retlw      0x54            ; Low Byte, ADRES = 604, Forward Power = 98.87 mW
retlw      0xF1            ; Low Byte, ADRES = 608, Forward Power = 108.5 mW
retlw      0x9E            ; Low Byte, ADRES = 612, Forward Power = 119.0 mW
retlw      0x2E            ; Low Byte, ADRES = 616, Forward Power = 130.6 mW
retlw      0x95            ; Low Byte, ADRES = 620, Forward Power = 143.2 mW
retlw      0x07            ; Low Byte, ADRES = 624, Forward Power = 157.1 mW
retlw      0x84            ; Low Byte, ADRES = 628, Forward Power = 172.4 mW
retlw      0x0E            ; Low Byte, ADRES = 632, Forward Power = 189.2 mW
retlw      0xA4            ; Low Byte, ADRES = 636, Forward Power = 207.5 mW
retlw      0x49            ; Low Byte, ADRES = 640, Forward Power = 227.7 mW
retlw      0xFE            ; Low Byte, ADRES = 644, Forward Power = 249.8 mW
retlw      0x63            ; Low Byte, ADRES = 648, Forward Power = 274.0 mW
retlw      0xD0            ; Low Byte, ADRES = 652, Forward Power = 300.7 mW
retlw      0x47            ; Low Byte, ADRES = 656, Forward Power = 329.9 mW
retlw      0xCA            ; Low Byte, ADRES = 660, Forward Power = 361.9 mW
retlw      0x5A            ; Low Byte, ADRES = 664, Forward Power = 397.0 mW
retlw      0xF8            ; Low Byte, ADRES = 668, Forward Power = 435.6 mW
retlw      0xA5            ; Low Byte, ADRES = 672, Forward Power = 477.9 mW
retlw      0x32            ; Low Byte, ADRES = 676, Forward Power = 524.3 mW
retlw      0x9A            ; Low Byte, ADRES = 680, Forward Power = 575.2 mW
retlw      0x0C            ; Low Byte, ADRES = 684, Forward Power = 631.1 mW
retlw      0x8A            ; Low Byte, ADRES = 688, Forward Power = 692.4 mW
retlw      0x14            ; Low Byte, ADRES = 692, Forward Power = 759.6 mW
retlw      0xAB            ; Low Byte, ADRES = 696, Forward Power = 833.4 mW
retlw      0x51            ; Low Byte, ADRES = 700, Forward Power = 914.3 mW
retlw      0x03            ; Low Byte, ADRES = 704, Forward Power = 1.003 W
retlw      0x67            ; Low Byte, ADRES = 708, Forward Power = 1.101 W
retlw      0xD4            ; Low Byte, ADRES = 712, Forward Power = 1.207 W
retlw      0x4C            ; Low Byte, ADRES = 716, Forward Power = 1.325 W
retlw      0xD0            ; Low Byte, ADRES = 720, Forward Power = 1.453 W
retlw      0x61            ; Low Byte, ADRES = 724, Forward Power = 1.594 W
retlw      0xFF            ; Low Byte, ADRES = 728, Forward Power = 1.749 W
retlw      0xAD            ; Low Byte, ADRES = 732, Forward Power = 1.919 W
retlw      0x36            ; Low Byte, ADRES = 736, Forward Power = 2.106 W
retlw      0x9F            ; Low Byte, ADRES = 740, Forward Power = 2.310 W
retlw      0x12            ; Low Byte, ADRES = 744, Forward Power = 2.534 W
retlw      0x90            ; Low Byte, ADRES = 748, Forward Power = 2.781 W
retlw      0x1A            ; Low Byte, ADRES = 752, Forward Power = 3.051 W
retlw      0xB2            ; Low Byte, ADRES = 756, Forward Power = 3.347 W
retlw      0x58            ; Low Byte, ADRES = 760, Forward Power = 3.672 W
retlw      0x07            ; Low Byte, ADRES = 764, Forward Power = 4.028 W
retlw      0x6B            ; Low Byte, ADRES = 768, Forward Power = 4.420 W
retlw      0xD9            ; Low Byte, ADRES = 772, Forward Power = 4.849 W
retlw      0x52            ; Low Byte, ADRES = 776, Forward Power = 5.320 W
retlw      0xD6            ; Low Byte, ADRES = 780, Forward Power = 5.836 W
retlw      0x67            ; Low Byte, ADRES = 784, Forward Power = 6.403 W
retlw      0x06            ; Low Byte, ADRES = 788, Forward Power = 7.025 W
retlw      0xB5            ; Low Byte, ADRES = 792, Forward Power = 7.707 W
retlw      0x3A            ; Low Byte, ADRES = 796, Forward Power = 8.456 W
retlw      0xA3            ; Low Byte, ADRES = 800, Forward Power = 9.277 W
retlw      0x17            ; Low Byte, ADRES = 804, Forward Power = 10.18 W
retlw      0x95            ; Low Byte, ADRES = 808, Forward Power = 11.17 W
retlw      0x20            ; Low Byte, ADRES = 812, Forward Power = 12.25 W
retlw      0xB8            ; Low Byte, ADRES = 816, Forward Power = 13.44 W
retlw      0x5F            ; Low Byte, ADRES = 820, Forward Power = 14.75 W
retlw      0x0B            ; Low Byte, ADRES = 824, Forward Power = 16.18 W
retlw      0x70            ; Low Byte, ADRES = 828, Forward Power = 17.75 W
retlw      0xDE            ; Low Byte, ADRES = 832, Forward Power = 19.47 W
retlw      0x57            ; Low Byte, ADRES = 836, Forward Power = 21.36 W
retlw      0xDC            ; Low Byte, ADRES = 840, Forward Power = 23.44 W
retlw      0x6E            ; Low Byte, ADRES = 844, Forward Power = 25.71 W
retlw      0x0E            ; Low Byte, ADRES = 848, Forward Power = 28.21 W
retlw      0xBD            ; Low Byte, ADRES = 852, Forward Power = 30.95 W
retlw      0x3F            ; Low Byte, ADRES = 856, Forward Power = 33.96 W
retlw      0xA8            ; Low Byte, ADRES = 860, Forward Power = 37.25 W
retlw      0x1C            ; Low Byte, ADRES = 864, Forward Power = 40.87 W
retlw      0x9B            ; Low Byte, ADRES = 868, Forward Power = 44.84 W
retlw      0x26            ; Low Byte, ADRES = 872, Forward Power = 49.20 W
retlw      0xBF            ; Low Byte, ADRES = 876, Forward Power = 53.97 W
retlw      0x67            ; Low Byte, ADRES = 880, Forward Power = 59.22 W
retlw      0x0F            ; Low Byte, ADRES = 884, Forward Power = 64.97 W
retlw      0x74            ; Low Byte, ADRES = 888, Forward Power = 71.28 W
retlw      0xE3            ; Low Byte, ADRES = 892, Forward Power = 78.20 W
retlw      0x5D            ; Low Byte, ADRES = 896, Forward Power = 85.79 W
retlw      0xE2            ; Low Byte, ADRES = 900, Forward Power = 94.12 W
```

```
        retlw   0x74            ; Low Byte, ADRES = 904, Forward Power = 103.3 W
        retlw   0x15            ; Low Byte, ADRES = 908, Forward Power = 113.3 W
        retlw   0xC5            ; Low Byte, ADRES = 912, Forward Power = 124.3 W
        retlw   0x43            ; Low Byte, ADRES = 916, Forward Power = 136.4 W
        retlw   0xAD            ; Low Byte, ADRES = 920, Forward Power = 149.6 W
        retlw   0x21            ; Low Byte, ADRES = 924, Forward Power = 164.1 W
        retlw   0xA1            ; Low Byte, ADRES = 928, Forward Power = 180.1 W
        retlw   0x2D            ; Low Byte, ADRES = 932, Forward Power = 197.6 W
        retlw   0xC6            ; Low Byte, ADRES = 936, Forward Power = 216.8 W
        retlw   0x6E            ; Low Byte, ADRES = 940, Forward Power = 237.8 W
        retlw   0x14            ; Low Byte, ADRES = 944, Forward Power = 260.9 W
        retlw   0x79            ; Low Byte, ADRES = 948, Forward Power = 286.2 W
        retlw   0xE8            ; Low Byte, ADRES = 952, Forward Power = 314.0 W
        retlw   0x62            ; Low Byte, ADRES = 956, Forward Power = 344.5 W
        retlw   0xE8            ; Low Byte, ADRES = 960, Forward Power = 378.0 W
        retlw   0x7B            ; Low Byte, ADRES = 964, Forward Power = 414.7 W
        retlw   0x1C            ; Low Byte, ADRES = 968, Forward Power = 455.0 W
        retlw   0xCD            ; Low Byte, ADRES = 972, Forward Power = 499.2 W
        retlw   0x47            ; Low Byte, ADRES = 976, Forward Power = 547.6 W
        retlw   0xB2            ; Low Byte, ADRES = 980, Forward Power = 600.8 W
        retlw   0x26            ; Low Byte, ADRES = 984, Forward Power = 659.2 W
        retlw   0xA6            ; Low Byte, ADRES = 988, Forward Power = 723.2 W
        retlw   0x33            ; Low Byte, ADRES = 992, Forward Power = 793.4 W
        retlw   0xCD            ; Low Byte, ADRES = 996, Forward Power = 870.5 W
        retlw   0x76            ; Low Byte, ADRES = 1000, Forward Power = 955.0 W
        retlw   0x18            ; Low Byte, ADRES = 1004, Forward Power = 1,047.7 W
        retlw   0x7E            ; Low Byte, ADRES = 1008, Forward Power = 1,149.5 W
        retlw   0xED            ; Low Byte, ADRES = 1012, Forward Power = 1,261.1 W
        retlw   0x68            ; Low Byte, ADRES = 1016, Forward Power = 1,383.6 W
        retlw   0xEE            ; Low Byte, ADRES = 1020, Forward Power = 1,518.0 W

FLowByteTable1
        retlw   0xAC            ; Low Byte, ADRES = 1, Forward Power = 84.54 nW
        retlw   0x39            ; Low Byte, ADRES = 5, Forward Power = 92.75 nW
        retlw   0xD4            ; Low Byte, ADRES = 9, Forward Power = 101.8 nW
        retlw   0x7E            ; Low Byte, ADRES = 13, Forward Power = 111.6 nW
        retlw   0x1C            ; Low Byte, ADRES = 17, Forward Power = 122.5 nW
        retlw   0x82            ; Low Byte, ADRES = 21, Forward Power = 134.4 nW
        retlw   0xF2            ; Low Byte, ADRES = 25, Forward Power = 147.4 nW
        retlw   0x6D            ; Low Byte, ADRES = 29, Forward Power = 161.7 nW
        retlw   0xF4            ; Low Byte, ADRES = 33, Forward Power = 177.5 nW
        retlw   0x88            ; Low Byte, ADRES = 37, Forward Power = 194.7 nW
        retlw   0x2B            ; Low Byte, ADRES = 41, Forward Power = 213.6 nW
        retlw   0xDD            ; Low Byte, ADRES = 45, Forward Power = 234.3 nW
        retlw   0x50            ; Low Byte, ADRES = 49, Forward Power = 257.1 nW
        retlw   0xBB            ; Low Byte, ADRES = 53, Forward Power = 282.1 nW
        retlw   0x31            ; Low Byte, ADRES = 57, Forward Power = 309.5 nW
        retlw   0xB2            ; Low Byte, ADRES = 61, Forward Power = 339.5 nW
        retlw   0x40            ; Low Byte, ADRES = 65, Forward Power = 372.5 nW
        retlw   0xDB            ; Low Byte, ADRES = 69, Forward Power = 408.6 nW
        retlw   0x86            ; Low Byte, ADRES = 73, Forward Power = 448.3 nW
        retlw   0x20            ; Low Byte, ADRES = 77, Forward Power = 491.9 nW
        retlw   0x87            ; Low Byte, ADRES = 81, Forward Power = 539.6 nW
        retlw   0xF7            ; Low Byte, ADRES = 85, Forward Power = 592.0 nW
        retlw   0x73            ; Low Byte, ADRES = 89, Forward Power = 649.5 nW
        retlw   0xFA            ; Low Byte, ADRES = 93, Forward Power = 712.6 nW
        retlw   0x8F            ; Low Byte, ADRES = 97, Forward Power = 781.8 nW
        retlw   0x32            ; Low Byte, ADRES = 101, Forward Power = 857.8 nW
        retlw   0xE5            ; Low Byte, ADRES = 105, Forward Power = 941.1 nW
        retlw   0x55            ; Low Byte, ADRES = 109, Forward Power = 1.032 uW
        retlw   0xC0            ; Low Byte, ADRES = 113, Forward Power = 1.133 uW
        retlw   0x36            ; Low Byte, ADRES = 117, Forward Power = 1.243 uW
        retlw   0xB8            ; Low Byte, ADRES = 121, Forward Power = 1.363 uW
        retlw   0x46            ; Low Byte, ADRES = 125, Forward Power = 1.496 uW
        retlw   0xE2            ; Low Byte, ADRES = 129, Forward Power = 1.641 uW
        retlw   0x8D            ; Low Byte, ADRES = 133, Forward Power = 1.800 uW
        retlw   0x24            ; Low Byte, ADRES = 137, Forward Power = 1.975 uW
        retlw   0x8B            ; Low Byte, ADRES = 141, Forward Power = 2.167 uW
        retlw   0xFC            ; Low Byte, ADRES = 145, Forward Power = 2.378 uW
        retlw   0x78            ; Low Byte, ADRES = 149, Forward Power = 2.608 uW
        retlw   0x00            ; Low Byte, ADRES = 153, Forward Power = 2.862 uW
        retlw   0x96            ; Low Byte, ADRES = 157, Forward Power = 3.140 uW
        retlw   0x39            ; Low Byte, ADRES = 161, Forward Power = 3.445 uW
        retlw   0xED            ; Low Byte, ADRES = 165, Forward Power = 3.779 uW
        retlw   0x59            ; Low Byte, ADRES = 169, Forward Power = 4.146 uW
        retlw   0xC5            ; Low Byte, ADRES = 173, Forward Power = 4.549 uW
        retlw   0x3C            ; Low Byte, ADRES = 177, Forward Power = 4.991 uW
        retlw   0xBE            ; Low Byte, ADRES = 181, Forward Power = 5.475 uW
        retlw   0x4C            ; Low Byte, ADRES = 185, Forward Power = 6.007 uW
        retlw   0xE9            ; Low Byte, ADRES = 189, Forward Power = 6.590 uW
        retlw   0x95            ; Low Byte, ADRES = 193, Forward Power = 7.230 uW
```

```
retlw      0x29            ; Low Byte, ADRES = 197, Forward Power = 7.933 uW
retlw      0x90            ; Low Byte, ADRES = 201, Forward Power = 8.703 uW
retlw      0x02            ; Low Byte, ADRES = 205, Forward Power = 9.548 uW
retlw      0x7E            ; Low Byte, ADRES = 209, Forward Power = 10.48 uW
retlw      0x07            ; Low Byte, ADRES = 213, Forward Power = 11.49 uW
retlw      0x9C            ; Low Byte, ADRES = 217, Forward Power = 12.61 uW
retlw      0x41            ; Low Byte, ADRES = 221, Forward Power = 13.83 uW
retlw      0xF5            ; Low Byte, ADRES = 225, Forward Power = 15.18 uW
retlw      0x5D            ; Low Byte, ADRES = 229, Forward Power = 16.65 uW
retlw      0xCA            ; Low Byte, ADRES = 233, Forward Power = 18.27 uW
retlw      0x41            ; Low Byte, ADRES = 237, Forward Power = 20.04 uW
retlw      0xC4            ; Low Byte, ADRES = 241, Forward Power = 21.99 uW
retlw      0x53            ; Low Byte, ADRES = 245, Forward Power = 24.12 uW
retlw      0xF0            ; Low Byte, ADRES = 249, Forward Power = 26.47 uW
retlw      0x9D            ; Low Byte, ADRES = 253, Forward Power = 29.04 uW
retlw      0x2D            ; Low Byte, ADRES = 257, Forward Power = 31.86 uW
retlw      0x95            ; Low Byte, ADRES = 261, Forward Power = 34.95 uW
retlw      0x07            ; Low Byte, ADRES = 265, Forward Power = 38.34 uW
retlw      0x84            ; Low Byte, ADRES = 269, Forward Power = 42.07 uW
retlw      0x0D            ; Low Byte, ADRES = 273, Forward Power = 46.15 uW
retlw      0xA3            ; Low Byte, ADRES = 277, Forward Power = 50.64 uW
retlw      0x48            ; Low Byte, ADRES = 281, Forward Power = 55.55 uW
retlw      0xFD            ; Low Byte, ADRES = 285, Forward Power = 60.95 uW
retlw      0x62            ; Low Byte, ADRES = 289, Forward Power = 66.87 uW
retlw      0xCF            ; Low Byte, ADRES = 293, Forward Power = 73.36 uW
retlw      0x46            ; Low Byte, ADRES = 297, Forward Power = 80.48 uW
retlw      0xC9            ; Low Byte, ADRES = 301, Forward Power = 88.30 uW
retlw      0x59            ; Low Byte, ADRES = 305, Forward Power = 96.88 uW
retlw      0xF7            ; Low Byte, ADRES = 309, Forward Power = 106.3 uW
retlw      0xA4            ; Low Byte, ADRES = 313, Forward Power = 116.6 uW
retlw      0x31            ; Low Byte, ADRES = 317, Forward Power = 127.9 uW
retlw      0x99            ; Low Byte, ADRES = 321, Forward Power = 140.4 uW
retlw      0x0C            ; Low Byte, ADRES = 325, Forward Power = 154.0 uW
retlw      0x89            ; Low Byte, ADRES = 329, Forward Power = 168.9 uW
retlw      0x13            ; Low Byte, ADRES = 333, Forward Power = 185.3 uW
retlw      0xAA            ; Low Byte, ADRES = 337, Forward Power = 203.3 uW
retlw      0x4F            ; Low Byte, ADRES = 341, Forward Power = 223.1 uW
retlw      0x03            ; Low Byte, ADRES = 345, Forward Power = 244.8 uW
retlw      0x66            ; Low Byte, ADRES = 349, Forward Power = 268.5 uW
retlw      0xD4            ; Low Byte, ADRES = 353, Forward Power = 294.6 uW
retlw      0x4C            ; Low Byte, ADRES = 357, Forward Power = 323.2 uW
retlw      0xCF            ; Low Byte, ADRES = 361, Forward Power = 354.6 uW
retlw      0x60            ; Low Byte, ADRES = 365, Forward Power = 389.0 uW
retlw      0xFE            ; Low Byte, ADRES = 369, Forward Power = 426.8 uW
retlw      0xAC            ; Low Byte, ADRES = 373, Forward Power = 468.3 uW
retlw      0x35            ; Low Byte, ADRES = 377, Forward Power = 513.7 uW
retlw      0x9E            ; Low Byte, ADRES = 381, Forward Power = 563.6 uW
retlw      0x11            ; Low Byte, ADRES = 385, Forward Power = 618.4 uW
retlw      0x8F            ; Low Byte, ADRES = 389, Forward Power = 678.4 uW
retlw      0x19            ; Low Byte, ADRES = 393, Forward Power = 744.3 uW
retlw      0xB1            ; Low Byte, ADRES = 397, Forward Power = 816.6 uW
retlw      0x57            ; Low Byte, ADRES = 401, Forward Power = 895.9 uW
retlw      0x07            ; Low Byte, ADRES = 405, Forward Power = 982.9 uW
retlw      0x6B            ; Low Byte, ADRES = 409, Forward Power = 1.078 mW
retlw      0xD9            ; Low Byte, ADRES = 413, Forward Power = 1.183 mW
retlw      0x51            ; Low Byte, ADRES = 417, Forward Power = 1.298 mW
retlw      0xD5            ; Low Byte, ADRES = 421, Forward Power = 1.424 mW
retlw      0x66            ; Low Byte, ADRES = 425, Forward Power = 1.562 mW
retlw      0x05            ; Low Byte, ADRES = 429, Forward Power = 1.714 mW
retlw      0xB4            ; Low Byte, ADRES = 433, Forward Power = 1.881 mW
retlw      0x3A            ; Low Byte, ADRES = 437, Forward Power = 2.063 mW
retlw      0xA3            ; Low Byte, ADRES = 441, Forward Power = 2.264 mW
retlw      0x16            ; Low Byte, ADRES = 445, Forward Power = 2.483 mW
retlw      0x94            ; Low Byte, ADRES = 449, Forward Power = 2.725 mW
retlw      0x1F            ; Low Byte, ADRES = 453, Forward Power = 2.989 mW
retlw      0xB7            ; Low Byte, ADRES = 457, Forward Power = 3.279 mW
retlw      0x5E            ; Low Byte, ADRES = 461, Forward Power = 3.598 mW
retlw      0x0B            ; Low Byte, ADRES = 465, Forward Power = 3.947 mW
retlw      0x6F            ; Low Byte, ADRES = 469, Forward Power = 4.331 mW
retlw      0xDE            ; Low Byte, ADRES = 473, Forward Power = 4.751 mW
retlw      0x56            ; Low Byte, ADRES = 477, Forward Power = 5.213 mW
retlw      0xDB            ; Low Byte, ADRES = 481, Forward Power = 5.719 mW
retlw      0x6D            ; Low Byte, ADRES = 485, Forward Power = 6.274 mW
retlw      0x0C            ; Low Byte, ADRES = 489, Forward Power = 6.883 mW
retlw      0xBC            ; Low Byte, ADRES = 493, Forward Power = 7.552 mW
retlw      0x3E            ; Low Byte, ADRES = 497, Forward Power = 8.285 mW
retlw      0xA7            ; Low Byte, ADRES = 501, Forward Power = 9.090 mW
retlw      0x1B            ; Low Byte, ADRES = 505, Forward Power = 9.973 mW
retlw      0x9A            ; Low Byte, ADRES = 509, Forward Power = 10.94 mW
retlw      0x25            ; Low Byte, ADRES = 513, Forward Power = 12.00 mW
retlw      0xBE            ; Low Byte, ADRES = 517, Forward Power = 13.17 mW
```

```
retlw      0x66            ; Low Byte, ADRES = 521, Forward Power = 14.45 mW
retlw      0x0F            ; Low Byte, ADRES = 525, Forward Power = 15.85 mW
retlw      0x74            ; Low Byte, ADRES = 529, Forward Power = 17.39 mW
retlw      0xE2            ; Low Byte, ADRES = 533, Forward Power = 19.08 mW
retlw      0x5C            ; Low Byte, ADRES = 537, Forward Power = 20.93 mW
retlw      0xE1            ; Low Byte, ADRES = 541, Forward Power = 22.97 mW
retlw      0x73            ; Low Byte, ADRES = 545, Forward Power = 25.20 mW
retlw      0x14            ; Low Byte, ADRES = 549, Forward Power = 27.64 mW
retlw      0xC4            ; Low Byte, ADRES = 553, Forward Power = 30.33 mW
retlw      0x42            ; Low Byte, ADRES = 557, Forward Power = 33.27 mW
retlw      0xAC            ; Low Byte, ADRES = 561, Forward Power = 36.50 mW
retlw      0x20            ; Low Byte, ADRES = 565, Forward Power = 40.05 mW
retlw      0xA0            ; Low Byte, ADRES = 569, Forward Power = 43.94 mW
retlw      0x2C            ; Low Byte, ADRES = 573, Forward Power = 48.21 mW
retlw      0xC5            ; Low Byte, ADRES = 577, Forward Power = 52.89 mW
retlw      0x6D            ; Low Byte, ADRES = 581, Forward Power = 58.02 mW
retlw      0x13            ; Low Byte, ADRES = 585, Forward Power = 63.66 mW
retlw      0x78            ; Low Byte, ADRES = 589, Forward Power = 69.84 mW
retlw      0xE7            ; Low Byte, ADRES = 593, Forward Power = 76.62 mW
retlw      0x61            ; Low Byte, ADRES = 597, Forward Power = 84.06 mW
retlw      0xE7            ; Low Byte, ADRES = 601, Forward Power = 92.23 mW
retlw      0x7A            ; Low Byte, ADRES = 605, Forward Power = 101.2 mW
retlw      0x1B            ; Low Byte, ADRES = 609, Forward Power = 111.0 mW
retlw      0xCB            ; Low Byte, ADRES = 613, Forward Power = 121.8 mW
retlw      0x47            ; Low Byte, ADRES = 617, Forward Power = 133.6 mW
retlw      0xB1            ; Low Byte, ADRES = 621, Forward Power = 146.6 mW
retlw      0x26            ; Low Byte, ADRES = 625, Forward Power = 160.8 mW
retlw      0xA6            ; Low Byte, ADRES = 629, Forward Power = 176.5 mW
retlw      0x32            ; Low Byte, ADRES = 633, Forward Power = 193.6 mW
retlw      0xCC            ; Low Byte, ADRES = 637, Forward Power = 212.4 mW
retlw      0x75            ; Low Byte, ADRES = 641, Forward Power = 233.0 mW
retlw      0x17            ; Low Byte, ADRES = 645, Forward Power = 255.6 mW
retlw      0x7D            ; Low Byte, ADRES = 649, Forward Power = 280.5 mW
retlw      0xEC            ; Low Byte, ADRES = 653, Forward Power = 307.7 mW
retlw      0x67            ; Low Byte, ADRES = 657, Forward Power = 337.6 mW
retlw      0xED            ; Low Byte, ADRES = 661, Forward Power = 370.4 mW
retlw      0x80            ; Low Byte, ADRES = 665, Forward Power = 406.3 mW
retlw      0x22            ; Low Byte, ADRES = 669, Forward Power = 445.8 mW
retlw      0xD3            ; Low Byte, ADRES = 673, Forward Power = 489.1 mW
retlw      0x4B            ; Low Byte, ADRES = 677, Forward Power = 536.6 mW
retlw      0xB6            ; Low Byte, ADRES = 681, Forward Power = 588.7 mW
retlw      0x2B            ; Low Byte, ADRES = 685, Forward Power = 645.9 mW
retlw      0xAB            ; Low Byte, ADRES = 689, Forward Power = 708.6 mW
retlw      0x38            ; Low Byte, ADRES = 693, Forward Power = 777.4 mW
retlw      0xD3            ; Low Byte, ADRES = 697, Forward Power = 852.9 mW
retlw      0x7C            ; Low Byte, ADRES = 701, Forward Power = 935.8 mW
retlw      0x1B            ; Low Byte, ADRES = 705, Forward Power = 1.027 W
retlw      0x81            ; Low Byte, ADRES = 709, Forward Power = 1.126 W
retlw      0xF1            ; Low Byte, ADRES = 713, Forward Power = 1.236 W
retlw      0x6C            ; Low Byte, ADRES = 717, Forward Power = 1.356 W
retlw      0xF3            ; Low Byte, ADRES = 721, Forward Power = 1.487 W
retlw      0x87            ; Low Byte, ADRES = 725, Forward Power = 1.632 W
retlw      0x29            ; Low Byte, ADRES = 729, Forward Power = 1.790 W
retlw      0xDB            ; Low Byte, ADRES = 733, Forward Power = 1.964 W
retlw      0x4F            ; Low Byte, ADRES = 737, Forward Power = 2.155 W
retlw      0xBA            ; Low Byte, ADRES = 741, Forward Power = 2.364 W
retlw      0x30            ; Low Byte, ADRES = 745, Forward Power = 2.594 W
retlw      0xB1            ; Low Byte, ADRES = 749, Forward Power = 2.846 W
retlw      0x3F            ; Low Byte, ADRES = 753, Forward Power = 3.122 W
retlw      0xDA            ; Low Byte, ADRES = 757, Forward Power = 3.425 W
retlw      0x84            ; Low Byte, ADRES = 761, Forward Power = 3.758 W
retlw      0x1F            ; Low Byte, ADRES = 765, Forward Power = 4.123 W
retlw      0x86            ; Low Byte, ADRES = 769, Forward Power = 4.523 W
retlw      0xF6            ; Low Byte, ADRES = 773, Forward Power = 4.962 W
retlw      0x72            ; Low Byte, ADRES = 777, Forward Power = 5.444 W
retlw      0xF9            ; Low Byte, ADRES = 781, Forward Power = 5.973 W
retlw      0x8E            ; Low Byte, ADRES = 785, Forward Power = 6.553 W
retlw      0x31            ; Low Byte, ADRES = 789, Forward Power = 7.190 W
retlw      0xE3            ; Low Byte, ADRES = 793, Forward Power = 7.888 W
retlw      0x54            ; Low Byte, ADRES = 797, Forward Power = 8.654 W
retlw      0xBF            ; Low Byte, ADRES = 801, Forward Power = 9.494 W
retlw      0x35            ; Low Byte, ADRES = 805, Forward Power = 10.42 W
retlw      0xB7            ; Low Byte, ADRES = 809, Forward Power = 11.43 W
retlw      0x45            ; Low Byte, ADRES = 813, Forward Power = 12.54 W
retlw      0xE1            ; Low Byte, ADRES = 817, Forward Power = 13.76 W
retlw      0x8C            ; Low Byte, ADRES = 821, Forward Power = 15.09 W
retlw      0x24            ; Low Byte, ADRES = 825, Forward Power = 16.56 W
retlw      0x8B            ; Low Byte, ADRES = 829, Forward Power = 18.16 W
retlw      0xFB            ; Low Byte, ADRES = 833, Forward Power = 19.93 W
retlw      0x77            ; Low Byte, ADRES = 837, Forward Power = 21.86 W
retlw      0xFF            ; Low Byte, ADRES = 841, Forward Power = 23.99 W
```

```
        retlw   0x94            ; Low Byte, ADRES = 845, Forward Power = 26.32 W
        retlw   0x38            ; Low Byte, ADRES = 849, Forward Power = 28.87 W
        retlw   0xEB            ; Low Byte, ADRES = 853, Forward Power = 31.68 W
        retlw   0x58            ; Low Byte, ADRES = 857, Forward Power = 34.75 W
        retlw   0xC4            ; Low Byte, ADRES = 861, Forward Power = 38.13 W
        retlw   0x3B            ; Low Byte, ADRES = 865, Forward Power = 41.83 W
        retlw   0xBD            ; Low Byte, ADRES = 869, Forward Power = 45.89 W
        retlw   0x4B            ; Low Byte, ADRES = 873, Forward Power = 50.35 W
        retlw   0xE8            ; Low Byte, ADRES = 877, Forward Power = 55.24 W
        retlw   0x93            ; Low Byte, ADRES = 881, Forward Power = 60.60 W
        retlw   0x28            ; Low Byte, ADRES = 885, Forward Power = 66.49 W
        retlw   0x8F            ; Low Byte, ADRES = 889, Forward Power = 72.95 W
        retlw   0x01            ; Low Byte, ADRES = 893, Forward Power = 80.03 W
        retlw   0x7D            ; Low Byte, ADRES = 897, Forward Power = 87.80 W
        retlw   0x05            ; Low Byte, ADRES = 901, Forward Power = 96.33 W
        retlw   0x9B            ; Low Byte, ADRES = 905, Forward Power = 105.7 W
        retlw   0x3F            ; Low Byte, ADRES = 909, Forward Power = 115.9 W
        retlw   0xF3            ; Low Byte, ADRES = 913, Forward Power = 127.2 W
        retlw   0x5D            ; Low Byte, ADRES = 917, Forward Power = 139.6 W
        retlw   0xC9            ; Low Byte, ADRES = 921, Forward Power = 153.1 W
        retlw   0x40            ; Low Byte, ADRES = 925, Forward Power = 168.0 W
        retlw   0xC2            ; Low Byte, ADRES = 929, Forward Power = 184.3 W
        retlw   0x52            ; Low Byte, ADRES = 933, Forward Power = 202.2 W
        retlw   0xEF            ; Low Byte, ADRES = 937, Forward Power = 221.8 W
        retlw   0x9B            ; Low Byte, ADRES = 941, Forward Power = 243.4 W
        retlw   0x2C            ; Low Byte, ADRES = 945, Forward Power = 267.0 W
        retlw   0x94            ; Low Byte, ADRES = 949, Forward Power = 292.9 W
        retlw   0x06            ; Low Byte, ADRES = 953, Forward Power = 321.4 W
        retlw   0x82            ; Low Byte, ADRES = 957, Forward Power = 352.6 W
        retlw   0x0B            ; Low Byte, ADRES = 961, Forward Power = 386.9 W
        retlw   0xA2            ; Low Byte, ADRES = 965, Forward Power = 424.4 W
        retlw   0x47            ; Low Byte, ADRES = 969, Forward Power = 465.6 W
        retlw   0xFB            ; Low Byte, ADRES = 973, Forward Power = 510.9 W
        retlw   0x61            ; Low Byte, ADRES = 977, Forward Power = 560.5 W
        retlw   0xCE            ; Low Byte, ADRES = 981, Forward Power = 614.9 W
        retlw   0x45            ; Low Byte, ADRES = 985, Forward Power = 674.6 W
        retlw   0xC8            ; Low Byte, ADRES = 989, Forward Power = 740.1 W
        retlw   0x58            ; Low Byte, ADRES = 993, Forward Power = 812.0 W
        retlw   0xF6            ; Low Byte, ADRES = 997, Forward Power = 890.9 W
        retlw   0xA3            ; Low Byte, ADRES = 1001, Forward Power = 977.4 W
        retlw   0x30            ; Low Byte, ADRES = 1005, Forward Power = 1,072.3 W
        retlw   0x98            ; Low Byte, ADRES = 1009, Forward Power = 1,176.4 W
        retlw   0x0B            ; Low Byte, ADRES = 1013, Forward Power = 1,290.7 W
        retlw   0x88            ; Low Byte, ADRES = 1017, Forward Power = 1,416.0 W
        retlw   0x12            ; Low Byte, ADRES = 1021, Forward Power = 1,553.5 W

FLowByteTable2
        retlw   0xCE            ; Low Byte, ADRES = 2, Forward Power = 86.52 nW
        retlw   0x5F            ; Low Byte, ADRES = 6, Forward Power = 94.92 nW
        retlw   0xFD            ; Low Byte, ADRES = 10, Forward Power = 104.1 nW
        retlw   0xAB            ; Low Byte, ADRES = 14, Forward Power = 114.3 nW
        retlw   0x35            ; Low Byte, ADRES = 18, Forward Power = 125.4 nW
        retlw   0x9D            ; Low Byte, ADRES = 22, Forward Power = 137.5 nW
        retlw   0x10            ; Low Byte, ADRES = 26, Forward Power = 150.9 nW
        retlw   0x8E            ; Low Byte, ADRES = 30, Forward Power = 165.5 nW
        retlw   0x18            ; Low Byte, ADRES = 34, Forward Power = 181.6 nW
        retlw   0xB0            ; Low Byte, ADRES = 38, Forward Power = 199.2 nW
        retlw   0x56            ; Low Byte, ADRES = 42, Forward Power = 218.6 nW
        retlw   0x06            ; Low Byte, ADRES = 46, Forward Power = 239.8 nW
        retlw   0x6A            ; Low Byte, ADRES = 50, Forward Power = 263.1 nW
        retlw   0xD8            ; Low Byte, ADRES = 54, Forward Power = 288.7 nW
        retlw   0x50            ; Low Byte, ADRES = 58, Forward Power = 316.7 nW
        retlw   0xD4            ; Low Byte, ADRES = 62, Forward Power = 347.5 nW
        retlw   0x65            ; Low Byte, ADRES = 66, Forward Power = 381.2 nW
        retlw   0x04            ; Low Byte, ADRES = 70, Forward Power = 418.2 nW
        retlw   0xB3            ; Low Byte, ADRES = 74, Forward Power = 458.8 nW
        retlw   0x39            ; Low Byte, ADRES = 78, Forward Power = 503.4 nW
        retlw   0xA2            ; Low Byte, ADRES = 82, Forward Power = 552.3 nW
        retlw   0x15            ; Low Byte, ADRES = 86, Forward Power = 605.9 nW
        retlw   0x94            ; Low Byte, ADRES = 90, Forward Power = 664.8 nW
        retlw   0x1E            ; Low Byte, ADRES = 94, Forward Power = 729.3 nW
        retlw   0xB6            ; Low Byte, ADRES = 98, Forward Power = 800.2 nW
        retlw   0x5D            ; Low Byte, ADRES = 102, Forward Power = 877.9 nW
        retlw   0x0A            ; Low Byte, ADRES = 106, Forward Power = 963.1 nW
        retlw   0x6F            ; Low Byte, ADRES = 110, Forward Power = 1.057 uW
        retlw   0xDD            ; Low Byte, ADRES = 114, Forward Power = 1.159 uW
        retlw   0x56            ; Low Byte, ADRES = 118, Forward Power = 1.272 uW
        retlw   0xDA            ; Low Byte, ADRES = 122, Forward Power = 1.395 uW
        retlw   0x6C            ; Low Byte, ADRES = 126, Forward Power = 1.531 uW
        retlw   0x0B            ; Low Byte, ADRES = 130, Forward Power = 1.680 uW
        retlw   0xBB            ; Low Byte, ADRES = 134, Forward Power = 1.843 uW
```

```
retlw      0x3D            ; Low Byte, ADRES = 138, Forward Power = 2.022 uW
retlw      0xA7            ; Low Byte, ADRES = 142, Forward Power = 2.218 uW
retlw      0x1A            ; Low Byte, ADRES = 146, Forward Power = 2.433 uW
retlw      0x99            ; Low Byte, ADRES = 150, Forward Power = 2.670 uW
retlw      0x24            ; Low Byte, ADRES = 154, Forward Power = 2.929 uW
retlw      0xBD            ; Low Byte, ADRES = 158, Forward Power = 3.213 uW
retlw      0x65            ; Low Byte, ADRES = 162, Forward Power = 3.525 uW
retlw      0x0E            ; Low Byte, ADRES = 166, Forward Power = 3.868 uW
retlw      0x73            ; Low Byte, ADRES = 170, Forward Power = 4.243 uW
retlw      0xE2            ; Low Byte, ADRES = 174, Forward Power = 4.655 uW
retlw      0x5B            ; Low Byte, ADRES = 178, Forward Power = 5.108 uW
retlw      0xE0            ; Low Byte, ADRES = 182, Forward Power = 5.604 uW
retlw      0x72            ; Low Byte, ADRES = 186, Forward Power = 6.148 uW
retlw      0x13            ; Low Byte, ADRES = 190, Forward Power = 6.745 uW
retlw      0xC2            ; Low Byte, ADRES = 194, Forward Power = 7.400 uW
retlw      0x42            ; Low Byte, ADRES = 198, Forward Power = 8.118 uW
retlw      0xAB            ; Low Byte, ADRES = 202, Forward Power = 8.907 uW
retlw      0x20            ; Low Byte, ADRES = 206, Forward Power = 9.772 uW
retlw      0x9F            ; Low Byte, ADRES = 210, Forward Power = 10.72 uW
retlw      0x2B            ; Low Byte, ADRES = 214, Forward Power = 11.76 uW
retlw      0xC4            ; Low Byte, ADRES = 218, Forward Power = 12.90 uW
retlw      0x6C            ; Low Byte, ADRES = 222, Forward Power = 14.16 uW
retlw      0x12            ; Low Byte, ADRES = 226, Forward Power = 15.53 uW
retlw      0x78            ; Low Byte, ADRES = 230, Forward Power = 17.04 uW
retlw      0xE7            ; Low Byte, ADRES = 234, Forward Power = 18.70 uW
retlw      0x61            ; Low Byte, ADRES = 238, Forward Power = 20.51 uW
retlw      0xE6            ; Low Byte, ADRES = 242, Forward Power = 22.50 uW
retlw      0x79            ; Low Byte, ADRES = 246, Forward Power = 24.69 uW
retlw      0x1A            ; Low Byte, ADRES = 250, Forward Power = 27.09 uW
retlw      0xCA            ; Low Byte, ADRES = 254, Forward Power = 29.72 uW
retlw      0x46            ; Low Byte, ADRES = 258, Forward Power = 32.60 uW
retlw      0xB0            ; Low Byte, ADRES = 262, Forward Power = 35.77 uW
retlw      0x25            ; Low Byte, ADRES = 266, Forward Power = 39.24 uW
retlw      0xA5            ; Low Byte, ADRES = 270, Forward Power = 43.05 uW
retlw      0x31            ; Low Byte, ADRES = 274, Forward Power = 47.23 uW
retlw      0xCB            ; Low Byte, ADRES = 278, Forward Power = 51.82 uW
retlw      0x74            ; Low Byte, ADRES = 282, Forward Power = 56.85 uW
retlw      0x17            ; Low Byte, ADRES = 286, Forward Power = 62.38 uW
retlw      0x7C            ; Low Byte, ADRES = 290, Forward Power = 68.43 uW
retlw      0xEC            ; Low Byte, ADRES = 294, Forward Power = 75.08 uW
retlw      0x66            ; Low Byte, ADRES = 298, Forward Power = 82.37 uW
retlw      0xEC            ; Low Byte, ADRES = 302, Forward Power = 90.37 uW
retlw      0x7F            ; Low Byte, ADRES = 306, Forward Power = 99.15 uW
retlw      0x21            ; Low Byte, ADRES = 310, Forward Power = 108.8 uW
retlw      0xD2            ; Low Byte, ADRES = 314, Forward Power = 119.3 uW
retlw      0x4A            ; Low Byte, ADRES = 318, Forward Power = 130.9 uW
retlw      0xB5            ; Low Byte, ADRES = 322, Forward Power = 143.6 uW
retlw      0x2A            ; Low Byte, ADRES = 326, Forward Power = 157.6 uW
retlw      0xAA            ; Low Byte, ADRES = 330, Forward Power = 172.9 uW
retlw      0x37            ; Low Byte, ADRES = 334, Forward Power = 189.7 uW
retlw      0xD2            ; Low Byte, ADRES = 338, Forward Power = 208.1 uW
retlw      0x7B            ; Low Byte, ADRES = 342, Forward Power = 228.3 uW
retlw      0x1B            ; Low Byte, ADRES = 346, Forward Power = 250.5 uW
retlw      0x81            ; Low Byte, ADRES = 350, Forward Power = 274.8 uW
retlw      0xF1            ; Low Byte, ADRES = 354, Forward Power = 301.5 uW
retlw      0x6B            ; Low Byte, ADRES = 358, Forward Power = 330.8 uW
retlw      0xF2            ; Low Byte, ADRES = 362, Forward Power = 362.9 uW
retlw      0x86            ; Low Byte, ADRES = 366, Forward Power = 398.2 uW
retlw      0x28            ; Low Byte, ADRES = 370, Forward Power = 436.8 uW
retlw      0xDA            ; Low Byte, ADRES = 374, Forward Power = 479.2 uW
retlw      0x4F            ; Low Byte, ADRES = 378, Forward Power = 525.8 uW
retlw      0xBA            ; Low Byte, ADRES = 382, Forward Power = 576.9 uW
retlw      0x2F            ; Low Byte, ADRES = 386, Forward Power = 632.9 uW
retlw      0xB0            ; Low Byte, ADRES = 390, Forward Power = 694.3 uW
retlw      0x3E            ; Low Byte, ADRES = 394, Forward Power = 761.8 uW
retlw      0xD9            ; Low Byte, ADRES = 398, Forward Power = 835.7 uW
retlw      0x83            ; Low Byte, ADRES = 402, Forward Power = 916.9 uW
retlw      0x1F            ; Low Byte, ADRES = 406, Forward Power = 1.006 mW
retlw      0x85            ; Low Byte, ADRES = 410, Forward Power = 1.104 mW
retlw      0xF6            ; Low Byte, ADRES = 414, Forward Power = 1.211 mW
retlw      0x71            ; Low Byte, ADRES = 418, Forward Power = 1.328 mW
retlw      0xF8            ; Low Byte, ADRES = 422, Forward Power = 1.457 mW
retlw      0x8D            ; Low Byte, ADRES = 426, Forward Power = 1.599 mW
retlw      0x2F            ; Low Byte, ADRES = 430, Forward Power = 1.754 mW
retlw      0xE2            ; Low Byte, ADRES = 434, Forward Power = 1.925 mW
retlw      0x53            ; Low Byte, ADRES = 438, Forward Power = 2.112 mW
retlw      0xBF            ; Low Byte, ADRES = 442, Forward Power = 2.317 mW
retlw      0x35            ; Low Byte, ADRES = 446, Forward Power = 2.542 mW
retlw      0xB6            ; Low Byte, ADRES = 450, Forward Power = 2.788 mW
retlw      0x44            ; Low Byte, ADRES = 454, Forward Power = 3.059 mW
retlw      0xE0            ; Low Byte, ADRES = 458, Forward Power = 3.356 mW
```

```
retlw      0x8B            ; Low Byte, ADRES = 462, Forward Power = 3.682 mW
retlw      0x23            ; Low Byte, ADRES = 466, Forward Power = 4.040 mW
retlw      0x8A            ; Low Byte, ADRES = 470, Forward Power = 4.432 mW
retlw      0xFB            ; Low Byte, ADRES = 474, Forward Power = 4.863 mW
retlw      0x76            ; Low Byte, ADRES = 478, Forward Power = 5.335 mW
retlw      0xFE            ; Low Byte, ADRES = 482, Forward Power = 5.853 mW
retlw      0x93            ; Low Byte, ADRES = 486, Forward Power = 6.421 mW
retlw      0x37            ; Low Byte, ADRES = 490, Forward Power = 7.045 mW
retlw      0xEA            ; Low Byte, ADRES = 494, Forward Power = 7.729 mW
retlw      0x57            ; Low Byte, ADRES = 498, Forward Power = 8.480 mW
retlw      0xC3            ; Low Byte, ADRES = 502, Forward Power = 9.303 mW
retlw      0x3A            ; Low Byte, ADRES = 506, Forward Power = 10.21 mW
retlw      0xBC            ; Low Byte, ADRES = 510, Forward Power = 11.20 mW
retlw      0x4A            ; Low Byte, ADRES = 514, Forward Power = 12.29 mW
retlw      0xE7            ; Low Byte, ADRES = 518, Forward Power = 13.48 mW
retlw      0x92            ; Low Byte, ADRES = 522, Forward Power = 14.79 mW
retlw      0x27            ; Low Byte, ADRES = 526, Forward Power = 16.22 mW
retlw      0x8E            ; Low Byte, ADRES = 530, Forward Power = 17.80 mW
retlw      0x00            ; Low Byte, ADRES = 534, Forward Power = 19.53 mW
retlw      0x7C            ; Low Byte, ADRES = 538, Forward Power = 21.42 mW
retlw      0x04            ; Low Byte, ADRES = 542, Forward Power = 23.50 mW
retlw      0x9A            ; Low Byte, ADRES = 546, Forward Power = 25.79 mW
retlw      0x3E            ; Low Byte, ADRES = 550, Forward Power = 28.29 mW
retlw      0xF2            ; Low Byte, ADRES = 554, Forward Power = 31.04 mW
retlw      0x5C            ; Low Byte, ADRES = 558, Forward Power = 34.05 mW
retlw      0xC8            ; Low Byte, ADRES = 562, Forward Power = 37.36 mW
retlw      0x3F            ; Low Byte, ADRES = 566, Forward Power = 40.99 mW
retlw      0xC2            ; Low Byte, ADRES = 570, Forward Power = 44.97 mW
retlw      0x51            ; Low Byte, ADRES = 574, Forward Power = 49.34 mW
retlw      0xEE            ; Low Byte, ADRES = 578, Forward Power = 54.13 mW
retlw      0x9A            ; Low Byte, ADRES = 582, Forward Power = 59.38 mW
retlw      0x2B            ; Low Byte, ADRES = 586, Forward Power = 65.15 mW
retlw      0x93            ; Low Byte, ADRES = 590, Forward Power = 71.48 mW
retlw      0x05            ; Low Byte, ADRES = 594, Forward Power = 78.42 mW
retlw      0x82            ; Low Byte, ADRES = 598, Forward Power = 86.03 mW
retlw      0x0B            ; Low Byte, ADRES = 602, Forward Power = 94.39 mW
retlw      0xA1            ; Low Byte, ADRES = 606, Forward Power = 103.6 mW
retlw      0x45            ; Low Byte, ADRES = 610, Forward Power = 113.6 mW
retlw      0xFA            ; Low Byte, ADRES = 614, Forward Power = 124.6 mW
retlw      0x60            ; Low Byte, ADRES = 618, Forward Power = 136.8 mW
retlw      0xCD            ; Low Byte, ADRES = 622, Forward Power = 150.0 mW
retlw      0x44            ; Low Byte, ADRES = 626, Forward Power = 164.6 mW
retlw      0xC7            ; Low Byte, ADRES = 630, Forward Power = 180.6 mW
retlw      0x57            ; Low Byte, ADRES = 634, Forward Power = 198.1 mW
retlw      0xF5            ; Low Byte, ADRES = 638, Forward Power = 217.4 mW
retlw      0xA2            ; Low Byte, ADRES = 642, Forward Power = 238.5 mW
retlw      0x30            ; Low Byte, ADRES = 646, Forward Power = 261.6 mW
retlw      0x98            ; Low Byte, ADRES = 650, Forward Power = 287.0 mW
retlw      0x0A            ; Low Byte, ADRES = 654, Forward Power = 314.9 mW
retlw      0x87            ; Low Byte, ADRES = 658, Forward Power = 345.5 mW
retlw      0x11            ; Low Byte, ADRES = 662, Forward Power = 379.1 mW
retlw      0xA7            ; Low Byte, ADRES = 666, Forward Power = 415.9 mW
retlw      0x4D            ; Low Byte, ADRES = 670, Forward Power = 456.3 mW
retlw      0x01            ; Low Byte, ADRES = 674, Forward Power = 500.6 mW
retlw      0x65            ; Low Byte, ADRES = 678, Forward Power = 549.2 mW
retlw      0xD2            ; Low Byte, ADRES = 682, Forward Power = 602.5 mW
retlw      0x4A            ; Low Byte, ADRES = 686, Forward Power = 661.0 mW
retlw      0xCD            ; Low Byte, ADRES = 690, Forward Power = 725.2 mW
retlw      0x5E            ; Low Byte, ADRES = 694, Forward Power = 795.6 mW
retlw      0xFC            ; Low Byte, ADRES = 698, Forward Power = 872.9 mW
retlw      0xA9            ; Low Byte, ADRES = 702, Forward Power = 957.7 mW
retlw      0x34            ; Low Byte, ADRES = 706, Forward Power = 1.051 W
retlw      0x9C            ; Low Byte, ADRES = 710, Forward Power = 1.153 W
retlw      0x0F            ; Low Byte, ADRES = 714, Forward Power = 1.265 W
retlw      0x8D            ; Low Byte, ADRES = 718, Forward Power = 1.388 W
retlw      0x17            ; Low Byte, ADRES = 722, Forward Power = 1.522 W
retlw      0xAE            ; Low Byte, ADRES = 726, Forward Power = 1.670 W
retlw      0x54            ; Low Byte, ADRES = 730, Forward Power = 1.832 W
retlw      0x05            ; Low Byte, ADRES = 734, Forward Power = 2.010 W
retlw      0x69            ; Low Byte, ADRES = 738, Forward Power = 2.205 W
retlw      0xD7            ; Low Byte, ADRES = 742, Forward Power = 2.420 W
retlw      0x4F            ; Low Byte, ADRES = 746, Forward Power = 2.655 W
retlw      0xD3            ; Low Byte, ADRES = 750, Forward Power = 2.912 W
retlw      0x64            ; Low Byte, ADRES = 754, Forward Power = 3.195 W
retlw      0x03            ; Low Byte, ADRES = 758, Forward Power = 3.506 W
retlw      0xB1            ; Low Byte, ADRES = 762, Forward Power = 3.846 W
retlw      0x38            ; Low Byte, ADRES = 766, Forward Power = 4.219 W
retlw      0xA1            ; Low Byte, ADRES = 770, Forward Power = 4.629 W
retlw      0x14            ; Low Byte, ADRES = 774, Forward Power = 5.079 W
retlw      0x92            ; Low Byte, ADRES = 778, Forward Power = 5.572 W
retlw      0x1D            ; Low Byte, ADRES = 782, Forward Power = 6.113 W
```

```
      retlw      0xB5            ; Low Byte, ADRES = 786, Forward Power = 6.707 W
      retlw      0x5C            ; Low Byte, ADRES = 790, Forward Power = 7.358 W
      retlw      0x09            ; Low Byte, ADRES = 794, Forward Power = 8.073 W
      retlw      0x6E            ; Low Byte, ADRES = 798, Forward Power = 8.857 W
      retlw      0xDC            ; Low Byte, ADRES = 802, Forward Power = 9.717 W
      retlw      0x55            ; Low Byte, ADRES = 806, Forward Power = 10.66 W
      retlw      0xD9            ; Low Byte, ADRES = 810, Forward Power = 11.70 W
      retlw      0x6A            ; Low Byte, ADRES = 814, Forward Power = 12.83 W
      retlw      0x0A            ; Low Byte, ADRES = 818, Forward Power = 14.08 W
      retlw      0xB9            ; Low Byte, ADRES = 822, Forward Power = 15.44 W
      retlw      0x3C            ; Low Byte, ADRES = 826, Forward Power = 16.94 W
      retlw      0xA6            ; Low Byte, ADRES = 830, Forward Power = 18.59 W
      retlw      0x19            ; Low Byte, ADRES = 834, Forward Power = 20.40 W
      retlw      0x98            ; Low Byte, ADRES = 838, Forward Power = 22.38 W
      retlw      0x23            ; Low Byte, ADRES = 842, Forward Power = 24.55 W
      retlw      0xBC            ; Low Byte, ADRES = 846, Forward Power = 26.93 W
      retlw      0x63            ; Low Byte, ADRES = 850, Forward Power = 29.55 W
      retlw      0x0D            ; Low Byte, ADRES = 854, Forward Power = 32.42 W
      retlw      0x72            ; Low Byte, ADRES = 858, Forward Power = 35.57 W
      retlw      0xE1            ; Low Byte, ADRES = 862, Forward Power = 39.02 W
      retlw      0x5A            ; Low Byte, ADRES = 866, Forward Power = 42.81 W
      retlw      0xDF            ; Low Byte, ADRES = 870, Forward Power = 46.97 W
      retlw      0x71            ; Low Byte, ADRES = 874, Forward Power = 51.53 W
      retlw      0x11            ; Low Byte, ADRES = 878, Forward Power = 56.53 W
      retlw      0xC1            ; Low Byte, ADRES = 882, Forward Power = 62.02 W
      retlw      0x41            ; Low Byte, ADRES = 886, Forward Power = 68.05 W
      retlw      0xAB            ; Low Byte, ADRES = 890, Forward Power = 74.66 W
      retlw      0x1F            ; Low Byte, ADRES = 894, Forward Power = 81.91 W
      retlw      0x9E            ; Low Byte, ADRES = 898, Forward Power = 89.86 W
      retlw      0x29            ; Low Byte, ADRES = 902, Forward Power = 98.59 W
      retlw      0xC3            ; Low Byte, ADRES = 906, Forward Power = 108.2 W
      retlw      0x6B            ; Low Byte, ADRES = 910, Forward Power = 118.7 W
      retlw      0x12            ; Low Byte, ADRES = 914, Forward Power = 130.2 W
      retlw      0x77            ; Low Byte, ADRES = 918, Forward Power = 142.8 W
      retlw      0xE6            ; Low Byte, ADRES = 922, Forward Power = 156.7 W
      retlw      0x5F            ; Low Byte, ADRES = 926, Forward Power = 171.9 W
      retlw      0xE5            ; Low Byte, ADRES = 930, Forward Power = 188.6 W
      retlw      0x78            ; Low Byte, ADRES = 934, Forward Power = 206.9 W
      retlw      0x18            ; Low Byte, ADRES = 938, Forward Power = 227.0 W
      retlw      0xC9            ; Low Byte, ADRES = 942, Forward Power = 249.1 W
      retlw      0x45            ; Low Byte, ADRES = 946, Forward Power = 273.3 W
      retlw      0xAF            ; Low Byte, ADRES = 950, Forward Power = 299.8 W
      retlw      0x24            ; Low Byte, ADRES = 954, Forward Power = 328.9 W
      retlw      0xA4            ; Low Byte, ADRES = 958, Forward Power = 360.9 W
      retlw      0x30            ; Low Byte, ADRES = 962, Forward Power = 395.9 W
      retlw      0xC9            ; Low Byte, ADRES = 966, Forward Power = 434.4 W
      retlw      0x72            ; Low Byte, ADRES = 970, Forward Power = 476.6 W
      retlw      0x16            ; Low Byte, ADRES = 974, Forward Power = 522.8 W
      retlw      0x7B            ; Low Byte, ADRES = 978, Forward Power = 573.6 W
      retlw      0xEB            ; Low Byte, ADRES = 982, Forward Power = 629.3 W
      retlw      0x65            ; Low Byte, ADRES = 986, Forward Power = 690.4 W
      retlw      0xEB            ; Low Byte, ADRES = 990, Forward Power = 757.5 W
      retlw      0x7E            ; Low Byte, ADRES = 994, Forward Power = 831.0 W
      retlw      0x20            ; Low Byte, ADRES = 998, Forward Power = 911.7 W
      retlw      0xD1            ; Low Byte, ADRES = 1002, Forward Power = 1,000.3 W
      retlw      0x49            ; Low Byte, ADRES = 1006, Forward Power = 1,097.4 W
      retlw      0xB4            ; Low Byte, ADRES = 1010, Forward Power = 1,204.0 W
      retlw      0x29            ; Low Byte, ADRES = 1014, Forward Power = 1,320.9 W
      retlw      0xA9            ; Low Byte, ADRES = 1018, Forward Power = 1,449.2 W
      retlw      0x36            ; Low Byte, ADRES = 1022, Forward Power = 1,590.0 W

FLowByteTable3
      retlw      0xF1            ; Low Byte, ADRES = 3, Forward Power = 88.55 nW
      retlw      0x85            ; Low Byte, ADRES = 7, Forward Power = 97.15 nW
      retlw      0x27            ; Low Byte, ADRES = 11, Forward Power = 106.6 nW
      retlw      0xD9            ; Low Byte, ADRES = 15, Forward Power = 116.9 nW
      retlw      0x4E            ; Low Byte, ADRES = 19, Forward Power = 128.3 nW
      retlw      0xB9            ; Low Byte, ADRES = 23, Forward Power = 140.8 nW
      retlw      0x2E            ; Low Byte, ADRES = 27, Forward Power = 154.4 nW
      retlw      0xAF            ; Low Byte, ADRES = 31, Forward Power = 169.4 nW
      retlw      0x3D            ; Low Byte, ADRES = 35, Forward Power = 185.9 nW
      retlw      0xD8            ; Low Byte, ADRES = 39, Forward Power = 203.9 nW
      retlw      0x82            ; Low Byte, ADRES = 43, Forward Power = 223.7 nW
      retlw      0x1E            ; Low Byte, ADRES = 47, Forward Power = 245.4 nW
      retlw      0x85            ; Low Byte, ADRES = 51, Forward Power = 269.3 nW
      retlw      0xF5            ; Low Byte, ADRES = 55, Forward Power = 295.4 nW
      retlw      0x70            ; Low Byte, ADRES = 59, Forward Power = 324.1 nW
      retlw      0xF7            ; Low Byte, ADRES = 63, Forward Power = 355.6 nW
      retlw      0x8C            ; Low Byte, ADRES = 67, Forward Power = 390.1 nW
      retlw      0x2E            ; Low Byte, ADRES = 71, Forward Power = 428.0 nW
      retlw      0xE1            ; Low Byte, ADRES = 75, Forward Power = 469.6 nW
```

```
retlw      0x52            ; Low Byte, ADRES = 79, Forward Power = 515.2 nW
retlw      0xBE            ; Low Byte, ADRES = 83, Forward Power = 565.2 nW
retlw      0x34            ; Low Byte, ADRES = 87, Forward Power = 620.1 nW
retlw      0xB5            ; Low Byte, ADRES = 91, Forward Power = 680.4 nW
retlw      0x43            ; Low Byte, ADRES = 95, Forward Power = 746.4 nW
retlw      0xDF            ; Low Byte, ADRES = 99, Forward Power = 818.9 nW
retlw      0x89            ; Low Byte, ADRES = 103, Forward Power = 898.4 nW
retlw      0x22            ; Low Byte, ADRES = 107, Forward Power = 985.7 nW
retlw      0x89            ; Low Byte, ADRES = 111, Forward Power = 1.081 uW
retlw      0xFA            ; Low Byte, ADRES = 115, Forward Power = 1.186 uW
retlw      0x76            ; Low Byte, ADRES = 119, Forward Power = 1.302 uW
retlw      0xFD            ; Low Byte, ADRES = 123, Forward Power = 1.428 uW
retlw      0x92            ; Low Byte, ADRES = 127, Forward Power = 1.567 uW
retlw      0x36            ; Low Byte, ADRES = 131, Forward Power = 1.719 uW
retlw      0xE9            ; Low Byte, ADRES = 135, Forward Power = 1.886 uW
retlw      0x57            ; Low Byte, ADRES = 139, Forward Power = 2.069 uW
retlw      0xC3            ; Low Byte, ADRES = 143, Forward Power = 2.270 uW
retlw      0x39            ; Low Byte, ADRES = 147, Forward Power = 2.490 uW
retlw      0xBB            ; Low Byte, ADRES = 151, Forward Power = 2.732 uW
retlw      0x49            ; Low Byte, ADRES = 155, Forward Power = 2.998 uW
retlw      0xE6            ; Low Byte, ADRES = 159, Forward Power = 3.289 uW
retlw      0x91            ; Low Byte, ADRES = 163, Forward Power = 3.608 uW
retlw      0x27            ; Low Byte, ADRES = 167, Forward Power = 3.958 uW
retlw      0x8E            ; Low Byte, ADRES = 171, Forward Power = 4.343 uW
retlw      0xFF            ; Low Byte, ADRES = 175, Forward Power = 4.765 uW
retlw      0x7B            ; Low Byte, ADRES = 179, Forward Power = 5.227 uW
retlw      0x03            ; Low Byte, ADRES = 183, Forward Power = 5.735 uW
retlw      0x99            ; Low Byte, ADRES = 187, Forward Power = 6.292 uW
retlw      0x3D            ; Low Byte, ADRES = 191, Forward Power = 6.903 uW
retlw      0xF1            ; Low Byte, ADRES = 195, Forward Power = 7.573 uW
retlw      0x5B            ; Low Byte, ADRES = 199, Forward Power = 8.309 uW
retlw      0xC8            ; Low Byte, ADRES = 203, Forward Power = 9.116 uW
retlw      0x3E            ; Low Byte, ADRES = 207, Forward Power = 10.00 uW
retlw      0xC1            ; Low Byte, ADRES = 211, Forward Power = 10.97 uW
retlw      0x50            ; Low Byte, ADRES = 215, Forward Power = 12.04 uW
retlw      0xED            ; Low Byte, ADRES = 219, Forward Power = 13.21 uW
retlw      0x99            ; Low Byte, ADRES = 223, Forward Power = 14.49 uW
retlw      0x2B            ; Low Byte, ADRES = 227, Forward Power = 15.90 uW
retlw      0x92            ; Low Byte, ADRES = 231, Forward Power = 17.44 uW
retlw      0x04            ; Low Byte, ADRES = 235, Forward Power = 19.13 uW
retlw      0x81            ; Low Byte, ADRES = 239, Forward Power = 20.99 uW
retlw      0x0A            ; Low Byte, ADRES = 243, Forward Power = 23.03 uW
retlw      0xA0            ; Low Byte, ADRES = 247, Forward Power = 25.27 uW
retlw      0x44            ; Low Byte, ADRES = 251, Forward Power = 27.72 uW
retlw      0xF9            ; Low Byte, ADRES = 255, Forward Power = 30.41 uW
retlw      0x60            ; Low Byte, ADRES = 259, Forward Power = 33.37 uW
retlw      0xCC            ; Low Byte, ADRES = 263, Forward Power = 36.61 uW
retlw      0x44            ; Low Byte, ADRES = 267, Forward Power = 40.16 uW
retlw      0xC7            ; Low Byte, ADRES = 271, Forward Power = 44.06 uW
retlw      0x56            ; Low Byte, ADRES = 275, Forward Power = 48.34 uW
retlw      0xF4            ; Low Byte, ADRES = 279, Forward Power = 53.04 uW
retlw      0xA0            ; Low Byte, ADRES = 283, Forward Power = 58.19 uW
retlw      0x2F            ; Low Byte, ADRES = 287, Forward Power = 63.84 uW
retlw      0x97            ; Low Byte, ADRES = 291, Forward Power = 70.04 uW
retlw      0x09            ; Low Byte, ADRES = 295, Forward Power = 76.84 uW
retlw      0x86            ; Low Byte, ADRES = 299, Forward Power = 84.30 uW
retlw      0x10            ; Low Byte, ADRES = 303, Forward Power = 92.49 uW
retlw      0xA6            ; Low Byte, ADRES = 307, Forward Power = 101.5 uW
retlw      0x4C            ; Low Byte, ADRES = 311, Forward Power = 111.3 uW
retlw      0x01            ; Low Byte, ADRES = 315, Forward Power = 122.1 uW
retlw      0x64            ; Low Byte, ADRES = 319, Forward Power = 134.0 uW
retlw      0xD1            ; Low Byte, ADRES = 323, Forward Power = 147.0 uW
retlw      0x49            ; Low Byte, ADRES = 327, Forward Power = 161.3 uW
retlw      0xCC            ; Low Byte, ADRES = 331, Forward Power = 177.0 uW
retlw      0x5D            ; Low Byte, ADRES = 335, Forward Power = 194.1 uW
retlw      0xFB            ; Low Byte, ADRES = 339, Forward Power = 213.0 uW
retlw      0xA8            ; Low Byte, ADRES = 343, Forward Power = 233.7 uW
retlw      0x33            ; Low Byte, ADRES = 347, Forward Power = 256.4 uW
retlw      0x9C            ; Low Byte, ADRES = 351, Forward Power = 281.3 uW
retlw      0x0E            ; Low Byte, ADRES = 355, Forward Power = 308.6 uW
retlw      0x8C            ; Low Byte, ADRES = 359, Forward Power = 338.5 uW
retlw      0x16            ; Low Byte, ADRES = 363, Forward Power = 371.4 uW
retlw      0xAD            ; Low Byte, ADRES = 367, Forward Power = 407.5 uW
retlw      0x53            ; Low Byte, ADRES = 371, Forward Power = 447.1 uW
retlw      0x05            ; Low Byte, ADRES = 375, Forward Power = 490.5 uW
retlw      0x69            ; Low Byte, ADRES = 379, Forward Power = 538.1 uW
retlw      0xD6            ; Low Byte, ADRES = 383, Forward Power = 590.4 uW
retlw      0x4E            ; Low Byte, ADRES = 387, Forward Power = 647.7 uW
retlw      0xD2            ; Low Byte, ADRES = 391, Forward Power = 710.6 uW
retlw      0x63            ; Low Byte, ADRES = 395, Forward Power = 779.6 uW
retlw      0x02            ; Low Byte, ADRES = 399, Forward Power = 855.3 uW
```

```
retlw      0xB0            ; Low Byte, ADRES = 403, Forward Power = 938.4 uW
retlw      0x38            ; Low Byte, ADRES = 407, Forward Power = 1.030 mW
retlw      0xA0            ; Low Byte, ADRES = 411, Forward Power = 1.130 mW
retlw      0x13            ; Low Byte, ADRES = 415, Forward Power = 1.239 mW
retlw      0x92            ; Low Byte, ADRES = 419, Forward Power = 1.360 mW
retlw      0x1C            ; Low Byte, ADRES = 423, Forward Power = 1.492 mW
retlw      0xB4            ; Low Byte, ADRES = 427, Forward Power = 1.636 mW
retlw      0x5B            ; Low Byte, ADRES = 431, Forward Power = 1.795 mW
retlw      0x09            ; Low Byte, ADRES = 435, Forward Power = 1.970 mW
retlw      0x6D            ; Low Byte, ADRES = 439, Forward Power = 2.161 mW
retlw      0xDB            ; Low Byte, ADRES = 443, Forward Power = 2.371 mW
retlw      0x54            ; Low Byte, ADRES = 447, Forward Power = 2.601 mW
retlw      0xD8            ; Low Byte, ADRES = 451, Forward Power = 2.854 mW
retlw      0x69            ; Low Byte, ADRES = 455, Forward Power = 3.131 mW
retlw      0x09            ; Low Byte, ADRES = 459, Forward Power = 3.435 mW
retlw      0xB8            ; Low Byte, ADRES = 463, Forward Power = 3.769 mW
retlw      0x3C            ; Low Byte, ADRES = 467, Forward Power = 4.134 mW
retlw      0xA5            ; Low Byte, ADRES = 471, Forward Power = 4.536 mW
retlw      0x19            ; Low Byte, ADRES = 475, Forward Power = 4.977 mW
retlw      0x97            ; Low Byte, ADRES = 479, Forward Power = 5.460 mW
retlw      0x22            ; Low Byte, ADRES = 483, Forward Power = 5.990 mW
retlw      0xBB            ; Low Byte, ADRES = 487, Forward Power = 6.572 mW
retlw      0x62            ; Low Byte, ADRES = 491, Forward Power = 7.210 mW
retlw      0x0D            ; Low Byte, ADRES = 495, Forward Power = 7.910 mW
retlw      0x72            ; Low Byte, ADRES = 499, Forward Power = 8.678 mW
retlw      0xE0            ; Low Byte, ADRES = 503, Forward Power = 9.521 mW
retlw      0x59            ; Low Byte, ADRES = 507, Forward Power = 10.45 mW
retlw      0xDE            ; Low Byte, ADRES = 511, Forward Power = 11.46 mW
retlw      0x70            ; Low Byte, ADRES = 515, Forward Power = 12.57 mW
retlw      0x10            ; Low Byte, ADRES = 519, Forward Power = 13.79 mW
retlw      0xC0            ; Low Byte, ADRES = 523, Forward Power = 15.13 mW
retlw      0x40            ; Low Byte, ADRES = 527, Forward Power = 16.60 mW
retlw      0xAA            ; Low Byte, ADRES = 531, Forward Power = 18.22 mW
retlw      0x1E            ; Low Byte, ADRES = 535, Forward Power = 19.99 mW
retlw      0x9D            ; Low Byte, ADRES = 539, Forward Power = 21.93 mW
retlw      0x28            ; Low Byte, ADRES = 543, Forward Power = 24.06 mW
retlw      0xC2            ; Low Byte, ADRES = 547, Forward Power = 26.39 mW
retlw      0x6A            ; Low Byte, ADRES = 551, Forward Power = 28.95 mW
retlw      0x11            ; Low Byte, ADRES = 555, Forward Power = 31.77 mW
retlw      0x76            ; Low Byte, ADRES = 559, Forward Power = 34.85 mW
retlw      0xE5            ; Low Byte, ADRES = 563, Forward Power = 38.24 mW
retlw      0x5F            ; Low Byte, ADRES = 567, Forward Power = 41.95 mW
retlw      0xE4            ; Low Byte, ADRES = 571, Forward Power = 46.02 mW
retlw      0x77            ; Low Byte, ADRES = 575, Forward Power = 50.49 mW
retlw      0x17            ; Low Byte, ADRES = 579, Forward Power = 55.40 mW
retlw      0xC8            ; Low Byte, ADRES = 583, Forward Power = 60.78 mW
retlw      0x44            ; Low Byte, ADRES = 587, Forward Power = 66.68 mW
retlw      0xAF            ; Low Byte, ADRES = 591, Forward Power = 73.15 mW
retlw      0x23            ; Low Byte, ADRES = 595, Forward Power = 80.26 mW
retlw      0xA3            ; Low Byte, ADRES = 599, Forward Power = 88.05 mW
retlw      0x2F            ; Low Byte, ADRES = 603, Forward Power = 96.60 mW
retlw      0xC8            ; Low Byte, ADRES = 607, Forward Power = 106.0 mW
retlw      0x71            ; Low Byte, ADRES = 611, Forward Power = 116.3 mW
retlw      0x15            ; Low Byte, ADRES = 615, Forward Power = 127.6 mW
retlw      0x7B            ; Low Byte, ADRES = 619, Forward Power = 140.0 mW
retlw      0xEA            ; Low Byte, ADRES = 623, Forward Power = 153.5 mW
retlw      0x64            ; Low Byte, ADRES = 627, Forward Power = 168.5 mW
retlw      0xEA            ; Low Byte, ADRES = 631, Forward Power = 184.8 mW
retlw      0x7D            ; Low Byte, ADRES = 635, Forward Power = 202.8 mW
retlw      0x1E            ; Low Byte, ADRES = 639, Forward Power = 222.5 mW
retlw      0xCF            ; Low Byte, ADRES = 643, Forward Power = 244.1 mW
retlw      0x49            ; Low Byte, ADRES = 647, Forward Power = 267.8 mW
retlw      0xB3            ; Low Byte, ADRES = 651, Forward Power = 293.8 mW
retlw      0x28            ; Low Byte, ADRES = 655, Forward Power = 322.3 mW
retlw      0xA8            ; Low Byte, ADRES = 659, Forward Power = 353.6 mW
retlw      0x35            ; Low Byte, ADRES = 663, Forward Power = 387.9 mW
retlw      0xCF            ; Low Byte, ADRES = 667, Forward Power = 425.6 mW
retlw      0x79            ; Low Byte, ADRES = 671, Forward Power = 467.0 mW
retlw      0x19            ; Low Byte, ADRES = 675, Forward Power = 512.3 mW
retlw      0x7F            ; Low Byte, ADRES = 679, Forward Power = 562.1 mW
retlw      0xEF            ; Low Byte, ADRES = 683, Forward Power = 616.6 mW
retlw      0x6A            ; Low Byte, ADRES = 687, Forward Power = 676.5 mW
retlw      0xF0            ; Low Byte, ADRES = 691, Forward Power = 742.2 mW
retlw      0x84            ; Low Byte, ADRES = 695, Forward Power = 814.3 mW
retlw      0x26            ; Low Byte, ADRES = 699, Forward Power = 893.4 mW
retlw      0xD7            ; Low Byte, ADRES = 703, Forward Power = 980.1 mW
retlw      0x4D            ; Low Byte, ADRES = 707, Forward Power = 1.075 W
retlw      0xB8            ; Low Byte, ADRES = 711, Forward Power = 1.180 W
retlw      0x2D            ; Low Byte, ADRES = 715, Forward Power = 1.294 W
retlw      0xAE            ; Low Byte, ADRES = 719, Forward Power = 1.420 W
retlw      0x3B            ; Low Byte, ADRES = 723, Forward Power = 1.558 W
```

```
      retlw      0xD6            ; Low Byte, ADRES = 727, Forward Power = 1.709 W
      retlw      0x80            ; Low Byte, ADRES = 731, Forward Power = 1.875 W
      retlw      0x1D            ; Low Byte, ADRES = 735, Forward Power = 2.057 W
      retlw      0x84            ; Low Byte, ADRES = 739, Forward Power = 2.257 W
      retlw      0xF4            ; Low Byte, ADRES = 743, Forward Power = 2.476 W
      retlw      0x6F            ; Low Byte, ADRES = 747, Forward Power = 2.717 W
      retlw      0xF6            ; Low Byte, ADRES = 751, Forward Power = 2.981 W
      retlw      0x8A            ; Low Byte, ADRES = 755, Forward Power = 3.270 W
      retlw      0x2D            ; Low Byte, ADRES = 759, Forward Power = 3.588 W
      retlw      0xDF            ; Low Byte, ADRES = 763, Forward Power = 3.936 W
      retlw      0x52            ; Low Byte, ADRES = 767, Forward Power = 4.318 W
      retlw      0xBD            ; Low Byte, ADRES = 771, Forward Power = 4.738 W
      retlw      0x33            ; Low Byte, ADRES = 775, Forward Power = 5.198 W
      retlw      0xB4            ; Low Byte, ADRES = 779, Forward Power = 5.703 W
      retlw      0x42            ; Low Byte, ADRES = 783, Forward Power = 6.256 W
      retlw      0xDD            ; Low Byte, ADRES = 787, Forward Power = 6.864 W
      retlw      0x88            ; Low Byte, ADRES = 791, Forward Power = 7.531 W
      retlw      0x22            ; Low Byte, ADRES = 795, Forward Power = 8.262 W
      retlw      0x88            ; Low Byte, ADRES = 799, Forward Power = 9.064 W
      retlw      0xF9            ; Low Byte, ADRES = 803, Forward Power = 9.945 W
      retlw      0x75            ; Low Byte, ADRES = 807, Forward Power = 10.91 W
      retlw      0xFC            ; Low Byte, ADRES = 811, Forward Power = 11.97 W
      retlw      0x91            ; Low Byte, ADRES = 815, Forward Power = 13.13 W
      retlw      0x34            ; Low Byte, ADRES = 819, Forward Power = 14.41 W
      retlw      0xE7            ; Low Byte, ADRES = 823, Forward Power = 15.81 W
      retlw      0x56            ; Low Byte, ADRES = 827, Forward Power = 17.34 W
      retlw      0xC2            ; Low Byte, ADRES = 831, Forward Power = 19.03 W
      retlw      0x38            ; Low Byte, ADRES = 835, Forward Power = 20.87 W
      retlw      0xBA            ; Low Byte, ADRES = 839, Forward Power = 22.90 W
      retlw      0x48            ; Low Byte, ADRES = 843, Forward Power = 25.13 W
      retlw      0xE4            ; Low Byte, ADRES = 847, Forward Power = 27.57 W
      retlw      0x90            ; Low Byte, ADRES = 851, Forward Power = 30.24 W
      retlw      0x26            ; Low Byte, ADRES = 855, Forward Power = 33.18 W
      retlw      0x8D            ; Low Byte, ADRES = 859, Forward Power = 36.40 W
      retlw      0xFE            ; Low Byte, ADRES = 863, Forward Power = 39.94 W
      retlw      0x7A            ; Low Byte, ADRES = 867, Forward Power = 43.81 W
      retlw      0x02            ; Low Byte, ADRES = 871, Forward Power = 48.07 W
      retlw      0x98            ; Low Byte, ADRES = 875, Forward Power = 52.74 W
      retlw      0x3C            ; Low Byte, ADRES = 879, Forward Power = 57.86 W
      retlw      0xEF            ; Low Byte, ADRES = 883, Forward Power = 63.48 W
      retlw      0x5A            ; Low Byte, ADRES = 887, Forward Power = 69.64 W
      retlw      0xC7            ; Low Byte, ADRES = 891, Forward Power = 76.41 W
      retlw      0x3D            ; Low Byte, ADRES = 895, Forward Power = 83.83 W
      retlw      0xC0            ; Low Byte, ADRES = 899, Forward Power = 91.97 W
      retlw      0x4E            ; Low Byte, ADRES = 903, Forward Power = 100.9 W
      retlw      0xEB            ; Low Byte, ADRES = 907, Forward Power = 110.7 W
      retlw      0x97            ; Low Byte, ADRES = 911, Forward Power = 121.4 W
      retlw      0x2A            ; Low Byte, ADRES = 915, Forward Power = 133.2 W
      retlw      0x91            ; Low Byte, ADRES = 919, Forward Power = 146.2 W
      retlw      0x03            ; Low Byte, ADRES = 923, Forward Power = 160.4 W
      retlw      0x80            ; Low Byte, ADRES = 927, Forward Power = 176.0 W
      retlw      0x08            ; Low Byte, ADRES = 931, Forward Power = 193.0 W
      retlw      0x9E            ; Low Byte, ADRES = 935, Forward Power = 211.8 W
      retlw      0x43            ; Low Byte, ADRES = 939, Forward Power = 232.4 W
      retlw      0xF7            ; Low Byte, ADRES = 943, Forward Power = 254.9 W
      retlw      0x5F            ; Low Byte, ADRES = 947, Forward Power = 279.7 W
      retlw      0xCB            ; Low Byte, ADRES = 951, Forward Power = 306.8 W
      retlw      0x43            ; Low Byte, ADRES = 955, Forward Power = 336.6 W
      retlw      0xC5            ; Low Byte, ADRES = 959, Forward Power = 369.3 W
      retlw      0x55            ; Low Byte, ADRES = 963, Forward Power = 405.2 W
      retlw      0xF2            ; Low Byte, ADRES = 967, Forward Power = 444.6 W
      retlw      0x9F            ; Low Byte, ADRES = 971, Forward Power = 487.7 W
      retlw      0x2E            ; Low Byte, ADRES = 975, Forward Power = 535.1 W
      retlw      0x96            ; Low Byte, ADRES = 979, Forward Power = 587.1 W
      retlw      0x08            ; Low Byte, ADRES = 983, Forward Power = 644.1 W
      retlw      0x85            ; Low Byte, ADRES = 987, Forward Power = 706.6 W
      retlw      0x0E            ; Low Byte, ADRES = 991, Forward Power = 775.2 W
      retlw      0xA5            ; Low Byte, ADRES = 995, Forward Power = 850.5 W
      retlw      0x4A            ; Low Byte, ADRES = 999, Forward Power = 933.1 W
      retlw      0xFF            ; Low Byte, ADRES = 1003, Forward Power = 1,023.7 W
      retlw      0x63            ; Low Byte, ADRES = 1007, Forward Power = 1,123.2 W
      retlw      0xD0            ; Low Byte, ADRES = 1011, Forward Power = 1,232.2 W
      retlw      0x48            ; Low Byte, ADRES = 1015, Forward Power = 1,351.9 W
      retlw      0xCB            ; Low Byte, ADRES = 1019, Forward Power = 1,483.2 W
      retlw      0x5B            ; Low Byte, ADRES = 1023, Forward Power = 1,627.2 W

FHighByteTable0
      retlw      0x0D            ; High Byte, ADRES = 0, Forward Power = 82.60 nW
      retlw      0x0E            ; High Byte, ADRES = 4, Forward Power = 90.63 nW
      retlw      0x0E            ; High Byte, ADRES = 8, Forward Power = 99.43 nW
      retlw      0x0F            ; High Byte, ADRES = 12, Forward Power = 109.1 nW
```

```
retlw      0x10            ; High Byte, ADRES = 16, Forward Power = 119.7 nW
retlw      0x10            ; High Byte, ADRES = 20, Forward Power = 131.3 nW
retlw      0x10            ; High Byte, ADRES = 24, Forward Power = 144.0 nW
retlw      0x11            ; High Byte, ADRES = 28, Forward Power = 158.0 nW
retlw      0x11            ; High Byte, ADRES = 32, Forward Power = 173.4 nW
retlw      0x12            ; High Byte, ADRES = 36, Forward Power = 190.2 nW
retlw      0x13            ; High Byte, ADRES = 40, Forward Power = 208.7 nW
retlw      0x13            ; High Byte, ADRES = 44, Forward Power = 229.0 nW
retlw      0x14            ; High Byte, ADRES = 48, Forward Power = 251.2 nW
retlw      0x14            ; High Byte, ADRES = 52, Forward Power = 275.6 nW
retlw      0x15            ; High Byte, ADRES = 56, Forward Power = 302.4 nW
retlw      0x15            ; High Byte, ADRES = 60, Forward Power = 331.7 nW
retlw      0x16            ; High Byte, ADRES = 64, Forward Power = 363.9 nW
retlw      0x16            ; High Byte, ADRES = 68, Forward Power = 399.3 nW
retlw      0x17            ; High Byte, ADRES = 72, Forward Power = 438.1 nW
retlw      0x18            ; High Byte, ADRES = 76, Forward Power = 480.6 nW
retlw      0x18            ; High Byte, ADRES = 80, Forward Power = 527.3 nW
retlw      0x18            ; High Byte, ADRES = 84, Forward Power = 578.5 nW
retlw      0x19            ; High Byte, ADRES = 88, Forward Power = 634.7 nW
retlw      0x19            ; High Byte, ADRES = 92, Forward Power = 696.3 nW
retlw      0x1A            ; High Byte, ADRES = 96, Forward Power = 763.9 nW
retlw      0x1B            ; High Byte, ADRES = 100, Forward Power = 838.1 nW
retlw      0x1B            ; High Byte, ADRES = 104, Forward Power = 919.5 nW
retlw      0x1C            ; High Byte, ADRES = 108, Forward Power = 1.009 uW
retlw      0x1C            ; High Byte, ADRES = 112, Forward Power = 1.107 uW
retlw      0x1D            ; High Byte, ADRES = 116, Forward Power = 1.214 uW
retlw      0x1D            ; High Byte, ADRES = 120, Forward Power = 1.332 uW
retlw      0x1E            ; High Byte, ADRES = 124, Forward Power = 1.462 uW
retlw      0x1E            ; High Byte, ADRES = 128, Forward Power = 1.603 uW
retlw      0x1F            ; High Byte, ADRES = 132, Forward Power = 1.759 uW
retlw      0x20            ; High Byte, ADRES = 136, Forward Power = 1.930 uW
retlw      0x20            ; High Byte, ADRES = 140, Forward Power = 2.117 uW
retlw      0x20            ; High Byte, ADRES = 144, Forward Power = 2.323 uW
retlw      0x21            ; High Byte, ADRES = 148, Forward Power = 2.549 uW
retlw      0x21            ; High Byte, ADRES = 152, Forward Power = 2.796 uW
retlw      0x22            ; High Byte, ADRES = 156, Forward Power = 3.068 uW
retlw      0x23            ; High Byte, ADRES = 160, Forward Power = 3.366 uW
retlw      0x23            ; High Byte, ADRES = 164, Forward Power = 3.693 uW
retlw      0x24            ; High Byte, ADRES = 168, Forward Power = 4.051 uW
retlw      0x24            ; High Byte, ADRES = 172, Forward Power = 4.445 uW
retlw      0x25            ; High Byte, ADRES = 176, Forward Power = 4.876 uW
retlw      0x25            ; High Byte, ADRES = 180, Forward Power = 5.350 uW
retlw      0x26            ; High Byte, ADRES = 184, Forward Power = 5.869 uW
retlw      0x26            ; High Byte, ADRES = 188, Forward Power = 6.439 uW
retlw      0x27            ; High Byte, ADRES = 192, Forward Power = 7.065 uW
retlw      0x28            ; High Byte, ADRES = 196, Forward Power = 7.751 uW
retlw      0x28            ; High Byte, ADRES = 200, Forward Power = 8.504 uW
retlw      0x28            ; High Byte, ADRES = 204, Forward Power = 9.329 uW
retlw      0x29            ; High Byte, ADRES = 208, Forward Power = 10.24 uW
retlw      0x29            ; High Byte, ADRES = 212, Forward Power = 11.23 uW
retlw      0x2A            ; High Byte, ADRES = 216, Forward Power = 12.32 uW
retlw      0x2B            ; High Byte, ADRES = 220, Forward Power = 13.52 uW
retlw      0x2B            ; High Byte, ADRES = 224, Forward Power = 14.83 uW
retlw      0x2C            ; High Byte, ADRES = 228, Forward Power = 16.27 uW
retlw      0x2C            ; High Byte, ADRES = 232, Forward Power = 17.85 uW
retlw      0x2D            ; High Byte, ADRES = 236, Forward Power = 19.58 uW
retlw      0x2D            ; High Byte, ADRES = 240, Forward Power = 21.48 uW
retlw      0x2E            ; High Byte, ADRES = 244, Forward Power = 23.57 uW
retlw      0x2E            ; High Byte, ADRES = 248, Forward Power = 25.86 uW
retlw      0x2F            ; High Byte, ADRES = 252, Forward Power = 28.37 uW
retlw      0x30            ; High Byte, ADRES = 256, Forward Power = 31.13 uW
retlw      0x30            ; High Byte, ADRES = 260, Forward Power = 34.15 uW
retlw      0x30            ; High Byte, ADRES = 264, Forward Power = 37.47 uW
retlw      0x31            ; High Byte, ADRES = 268, Forward Power = 41.10 uW
retlw      0x31            ; High Byte, ADRES = 272, Forward Power = 45.10 uW
retlw      0x32            ; High Byte, ADRES = 276, Forward Power = 49.48 uW
retlw      0x33            ; High Byte, ADRES = 280, Forward Power = 54.28 uW
retlw      0x33            ; High Byte, ADRES = 284, Forward Power = 59.55 uW
retlw      0x34            ; High Byte, ADRES = 288, Forward Power = 65.33 uW
retlw      0x34            ; High Byte, ADRES = 292, Forward Power = 71.68 uW
retlw      0x35            ; High Byte, ADRES = 296, Forward Power = 78.64 uW
retlw      0x35            ; High Byte, ADRES = 300, Forward Power = 86.28 uW
retlw      0x36            ; High Byte, ADRES = 304, Forward Power = 94.66 uW
retlw      0x36            ; High Byte, ADRES = 308, Forward Power = 103.8 uW
retlw      0x37            ; High Byte, ADRES = 312, Forward Power = 113.9 uW
retlw      0x38            ; High Byte, ADRES = 316, Forward Power = 125.0 uW
retlw      0x38            ; High Byte, ADRES = 320, Forward Power = 137.1 uW
retlw      0x38            ; High Byte, ADRES = 324, Forward Power = 150.5 uW
retlw      0x39            ; High Byte, ADRES = 328, Forward Power = 165.1 uW
retlw      0x39            ; High Byte, ADRES = 332, Forward Power = 181.1 uW
retlw      0x3A            ; High Byte, ADRES = 336, Forward Power = 198.7 uW
```

```
retlw      0x3B            ; High Byte, ADRES = 340, Forward Power = 218.0 uW
retlw      0x3B            ; High Byte, ADRES = 344, Forward Power = 239.2 uW
retlw      0x3C            ; High Byte, ADRES = 348, Forward Power = 262.4 uW
retlw      0x3C            ; High Byte, ADRES = 352, Forward Power = 287.9 uW
retlw      0x3D            ; High Byte, ADRES = 356, Forward Power = 315.8 uW
retlw      0x3D            ; High Byte, ADRES = 360, Forward Power = 346.5 uW
retlw      0x3E            ; High Byte, ADRES = 364, Forward Power = 380.1 uW
retlw      0x3E            ; High Byte, ADRES = 368, Forward Power = 417.0 uW
retlw      0x3F            ; High Byte, ADRES = 372, Forward Power = 457.5 uW
retlw      0x40            ; High Byte, ADRES = 376, Forward Power = 502.0 uW
retlw      0x40            ; High Byte, ADRES = 380, Forward Power = 550.7 uW
retlw      0x40            ; High Byte, ADRES = 384, Forward Power = 604.2 uW
retlw      0x41            ; High Byte, ADRES = 388, Forward Power = 662.9 uW
retlw      0x41            ; High Byte, ADRES = 392, Forward Power = 727.3 uW
retlw      0x42            ; High Byte, ADRES = 396, Forward Power = 797.9 uW
retlw      0x43            ; High Byte, ADRES = 400, Forward Power = 875.4 uW
retlw      0x43            ; High Byte, ADRES = 404, Forward Power = 960.4 uW
retlw      0x44            ; High Byte, ADRES = 408, Forward Power = 1.054 mW
retlw      0x44            ; High Byte, ADRES = 412, Forward Power = 1.156 mW
retlw      0x45            ; High Byte, ADRES = 416, Forward Power = 1.268 mW
retlw      0x45            ; High Byte, ADRES = 420, Forward Power = 1.391 mW
retlw      0x46            ; High Byte, ADRES = 424, Forward Power = 1.527 mW
retlw      0x46            ; High Byte, ADRES = 428, Forward Power = 1.675 mW
retlw      0x47            ; High Byte, ADRES = 432, Forward Power = 1.837 mW
retlw      0x48            ; High Byte, ADRES = 436, Forward Power = 2.016 mW
retlw      0x48            ; High Byte, ADRES = 440, Forward Power = 2.212 mW
retlw      0x48            ; High Byte, ADRES = 444, Forward Power = 2.426 mW
retlw      0x49            ; High Byte, ADRES = 448, Forward Power = 2.662 mW
retlw      0x49            ; High Byte, ADRES = 452, Forward Power = 2.921 mW
retlw      0x4A            ; High Byte, ADRES = 456, Forward Power = 3.204 mW
retlw      0x4B            ; High Byte, ADRES = 460, Forward Power = 3.515 mW
retlw      0x4B            ; High Byte, ADRES = 464, Forward Power = 3.857 mW
retlw      0x4C            ; High Byte, ADRES = 468, Forward Power = 4.231 mW
retlw      0x4C            ; High Byte, ADRES = 472, Forward Power = 4.642 mW
retlw      0x4D            ; High Byte, ADRES = 476, Forward Power = 5.093 mW
retlw      0x4D            ; High Byte, ADRES = 480, Forward Power = 5.588 mW
retlw      0x4E            ; High Byte, ADRES = 484, Forward Power = 6.130 mW
retlw      0x4E            ; High Byte, ADRES = 488, Forward Power = 6.726 mW
retlw      0x4F            ; High Byte, ADRES = 492, Forward Power = 7.379 mW
retlw      0x50            ; High Byte, ADRES = 496, Forward Power = 8.096 mW
retlw      0x50            ; High Byte, ADRES = 500, Forward Power = 8.882 mW
retlw      0x50            ; High Byte, ADRES = 504, Forward Power = 9.744 mW
retlw      0x51            ; High Byte, ADRES = 508, Forward Power = 10.69 mW
retlw      0x52            ; High Byte, ADRES = 512, Forward Power = 11.73 mW
retlw      0x52            ; High Byte, ADRES = 516, Forward Power = 12.87 mW
retlw      0x53            ; High Byte, ADRES = 520, Forward Power = 14.12 mW
retlw      0x53            ; High Byte, ADRES = 524, Forward Power = 15.49 mW
retlw      0x54            ; High Byte, ADRES = 528, Forward Power = 16.99 mW
retlw      0x54            ; High Byte, ADRES = 532, Forward Power = 18.64 mW
retlw      0x55            ; High Byte, ADRES = 536, Forward Power = 20.45 mW
retlw      0x55            ; High Byte, ADRES = 540, Forward Power = 22.44 mW
retlw      0x56            ; High Byte, ADRES = 544, Forward Power = 24.62 mW
retlw      0x56            ; High Byte, ADRES = 548, Forward Power = 27.01 mW
retlw      0x57            ; High Byte, ADRES = 552, Forward Power = 29.63 mW
retlw      0x58            ; High Byte, ADRES = 556, Forward Power = 32.51 mW
retlw      0x58            ; High Byte, ADRES = 560, Forward Power = 35.67 mW
retlw      0x59            ; High Byte, ADRES = 564, Forward Power = 39.13 mW
retlw      0x59            ; High Byte, ADRES = 568, Forward Power = 42.93 mW
retlw      0x5A            ; High Byte, ADRES = 572, Forward Power = 47.10 mW
retlw      0x5A            ; High Byte, ADRES = 576, Forward Power = 51.68 mW
retlw      0x5B            ; High Byte, ADRES = 580, Forward Power = 56.69 mW
retlw      0x5B            ; High Byte, ADRES = 584, Forward Power = 62.20 mW
retlw      0x5C            ; High Byte, ADRES = 588, Forward Power = 68.24 mW
retlw      0x5C            ; High Byte, ADRES = 592, Forward Power = 74.87 mW
retlw      0x5D            ; High Byte, ADRES = 596, Forward Power = 82.14 mW
retlw      0x5D            ; High Byte, ADRES = 600, Forward Power = 90.12 mW
retlw      0x5E            ; High Byte, ADRES = 604, Forward Power = 98.87 mW
retlw      0x5E            ; High Byte, ADRES = 608, Forward Power = 108.5 mW
retlw      0x5F            ; High Byte, ADRES = 612, Forward Power = 119.0 mW
retlw      0x60            ; High Byte, ADRES = 616, Forward Power = 130.6 mW
retlw      0x60            ; High Byte, ADRES = 620, Forward Power = 143.2 mW
retlw      0x61            ; High Byte, ADRES = 624, Forward Power = 157.1 mW
retlw      0x61            ; High Byte, ADRES = 628, Forward Power = 172.4 mW
retlw      0x62            ; High Byte, ADRES = 632, Forward Power = 189.2 mW
retlw      0x62            ; High Byte, ADRES = 636, Forward Power = 207.5 mW
retlw      0x63            ; High Byte, ADRES = 640, Forward Power = 227.7 mW
retlw      0x63            ; High Byte, ADRES = 644, Forward Power = 249.8 mW
retlw      0x64            ; High Byte, ADRES = 648, Forward Power = 274.0 mW
retlw      0x64            ; High Byte, ADRES = 652, Forward Power = 300.7 mW
retlw      0x65            ; High Byte, ADRES = 656, Forward Power = 329.9 mW
retlw      0x65            ; High Byte, ADRES = 660, Forward Power = 361.9 mW
```

```
retlw      0x66            ; High Byte, ADRES = 664, Forward Power = 397.0 mW
retlw      0x66            ; High Byte, ADRES = 668, Forward Power = 435.6 mW
retlw      0x67            ; High Byte, ADRES = 672, Forward Power = 477.9 mW
retlw      0x68            ; High Byte, ADRES = 676, Forward Power = 524.3 mW
retlw      0x68            ; High Byte, ADRES = 680, Forward Power = 575.2 mW
retlw      0x69            ; High Byte, ADRES = 684, Forward Power = 631.1 mW
retlw      0x69            ; High Byte, ADRES = 688, Forward Power = 692.4 mW
retlw      0x6A            ; High Byte, ADRES = 692, Forward Power = 759.6 mW
retlw      0x6A            ; High Byte, ADRES = 696, Forward Power = 833.4 mW
retlw      0x6B            ; High Byte, ADRES = 700, Forward Power = 914.3 mW
retlw      0x6C            ; High Byte, ADRES = 704, Forward Power = 1.003 W
retlw      0x6C            ; High Byte, ADRES = 708, Forward Power = 1.101 W
retlw      0x6C            ; High Byte, ADRES = 712, Forward Power = 1.207 W
retlw      0x6D            ; High Byte, ADRES = 716, Forward Power = 1.325 W
retlw      0x6D            ; High Byte, ADRES = 720, Forward Power = 1.453 W
retlw      0x6E            ; High Byte, ADRES = 724, Forward Power = 1.594 W
retlw      0x6E            ; High Byte, ADRES = 728, Forward Power = 1.749 W
retlw      0x6F            ; High Byte, ADRES = 732, Forward Power = 1.919 W
retlw      0x70            ; High Byte, ADRES = 736, Forward Power = 2.106 W
retlw      0x70            ; High Byte, ADRES = 740, Forward Power = 2.310 W
retlw      0x71            ; High Byte, ADRES = 744, Forward Power = 2.534 W
retlw      0x71            ; High Byte, ADRES = 748, Forward Power = 2.781 W
retlw      0x72            ; High Byte, ADRES = 752, Forward Power = 3.051 W
retlw      0x72            ; High Byte, ADRES = 756, Forward Power = 3.347 W
retlw      0x73            ; High Byte, ADRES = 760, Forward Power = 3.672 W
retlw      0x74            ; High Byte, ADRES = 764, Forward Power = 4.028 W
retlw      0x74            ; High Byte, ADRES = 768, Forward Power = 4.420 W
retlw      0x74            ; High Byte, ADRES = 772, Forward Power = 4.849 W
retlw      0x75            ; High Byte, ADRES = 776, Forward Power = 5.320 W
retlw      0x75            ; High Byte, ADRES = 780, Forward Power = 5.836 W
retlw      0x76            ; High Byte, ADRES = 784, Forward Power = 6.403 W
retlw      0x77            ; High Byte, ADRES = 788, Forward Power = 7.025 W
retlw      0x77            ; High Byte, ADRES = 792, Forward Power = 7.707 W
retlw      0x78            ; High Byte, ADRES = 796, Forward Power = 8.456 W
retlw      0x78            ; High Byte, ADRES = 800, Forward Power = 9.277 W
retlw      0x79            ; High Byte, ADRES = 804, Forward Power = 10.18 W
retlw      0x79            ; High Byte, ADRES = 808, Forward Power = 11.17 W
retlw      0x7A            ; High Byte, ADRES = 812, Forward Power = 12.25 W
retlw      0x7A            ; High Byte, ADRES = 816, Forward Power = 13.44 W
retlw      0x7B            ; High Byte, ADRES = 820, Forward Power = 14.75 W
retlw      0x7C            ; High Byte, ADRES = 824, Forward Power = 16.18 W
retlw      0x7C            ; High Byte, ADRES = 828, Forward Power = 17.75 W
retlw      0x7C            ; High Byte, ADRES = 832, Forward Power = 19.47 W
retlw      0x7D            ; High Byte, ADRES = 836, Forward Power = 21.36 W
retlw      0x7D            ; High Byte, ADRES = 840, Forward Power = 23.44 W
retlw      0x7E            ; High Byte, ADRES = 844, Forward Power = 25.71 W
retlw      0x7F            ; High Byte, ADRES = 848, Forward Power = 28.21 W
retlw      0x7F            ; High Byte, ADRES = 852, Forward Power = 30.95 W
retlw      0x80            ; High Byte, ADRES = 856, Forward Power = 33.96 W
retlw      0x80            ; High Byte, ADRES = 860, Forward Power = 37.25 W
retlw      0x81            ; High Byte, ADRES = 864, Forward Power = 40.87 W
retlw      0x81            ; High Byte, ADRES = 868, Forward Power = 44.84 W
retlw      0x82            ; High Byte, ADRES = 872, Forward Power = 49.20 W
retlw      0x82            ; High Byte, ADRES = 876, Forward Power = 53.97 W
retlw      0x83            ; High Byte, ADRES = 880, Forward Power = 59.22 W
retlw      0x84            ; High Byte, ADRES = 884, Forward Power = 64.97 W
retlw      0x84            ; High Byte, ADRES = 888, Forward Power = 71.28 W
retlw      0x84            ; High Byte, ADRES = 892, Forward Power = 78.20 W
retlw      0x85            ; High Byte, ADRES = 896, Forward Power = 85.79 W
retlw      0x85            ; High Byte, ADRES = 900, Forward Power = 94.12 W
retlw      0x86            ; High Byte, ADRES = 904, Forward Power = 103.3 W
retlw      0x87            ; High Byte, ADRES = 908, Forward Power = 113.3 W
retlw      0x87            ; High Byte, ADRES = 912, Forward Power = 124.3 W
retlw      0x88            ; High Byte, ADRES = 916, Forward Power = 136.4 W
retlw      0x88            ; High Byte, ADRES = 920, Forward Power = 149.6 W
retlw      0x89            ; High Byte, ADRES = 924, Forward Power = 164.1 W
retlw      0x89            ; High Byte, ADRES = 928, Forward Power = 180.1 W
retlw      0x8A            ; High Byte, ADRES = 932, Forward Power = 197.6 W
retlw      0x8A            ; High Byte, ADRES = 936, Forward Power = 216.8 W
retlw      0x8B            ; High Byte, ADRES = 940, Forward Power = 237.8 W
retlw      0x8C            ; High Byte, ADRES = 944, Forward Power = 260.9 W
retlw      0x8C            ; High Byte, ADRES = 948, Forward Power = 286.2 W
retlw      0x8C            ; High Byte, ADRES = 952, Forward Power = 314.0 W
retlw      0x8D            ; High Byte, ADRES = 956, Forward Power = 344.5 W
retlw      0x8D            ; High Byte, ADRES = 960, Forward Power = 378.0 W
retlw      0x8E            ; High Byte, ADRES = 964, Forward Power = 414.7 W
retlw      0x8F            ; High Byte, ADRES = 968, Forward Power = 455.0 W
retlw      0x8F            ; High Byte, ADRES = 972, Forward Power = 499.2 W
retlw      0x90            ; High Byte, ADRES = 976, Forward Power = 547.6 W
retlw      0x90            ; High Byte, ADRES = 980, Forward Power = 600.8 W
retlw      0x91            ; High Byte, ADRES = 984, Forward Power = 659.2 W
```

```
        retlw   0x91            ; High Byte, ADRES = 988, Forward Power = 723.2 W
        retlw   0x92            ; High Byte, ADRES = 992, Forward Power = 793.4 W
        retlw   0x92            ; High Byte, ADRES = 996, Forward Power = 870.5 W
        retlw   0x93            ; High Byte, ADRES = 1000, Forward Power = 955.0 W
        retlw   0x94            ; High Byte, ADRES = 1004, Forward Power = 1,047.7 W
        retlw   0x94            ; High Byte, ADRES = 1008, Forward Power = 1,149.5 W
        retlw   0x94            ; High Byte, ADRES = 1012, Forward Power = 1,261.1 W
        retlw   0x95            ; High Byte, ADRES = 1016, Forward Power = 1,383.6 W
        retlw   0x95            ; High Byte, ADRES = 1020, Forward Power = 1,518.0 W

FHighByteTable1
        retlw   0x0D            ; High Byte, ADRES = 1, Forward Power = 84.54 nW
        retlw   0x0E            ; High Byte, ADRES = 5, Forward Power = 92.75 nW
        retlw   0x0E            ; High Byte, ADRES = 9, Forward Power = 101.8 nW
        retlw   0x0F            ; High Byte, ADRES = 13, Forward Power = 111.6 nW
        retlw   0x10            ; High Byte, ADRES = 17, Forward Power = 122.5 nW
        retlw   0x10            ; High Byte, ADRES = 21, Forward Power = 134.4 nW
        retlw   0x10            ; High Byte, ADRES = 25, Forward Power = 147.4 nW
        retlw   0x11            ; High Byte, ADRES = 29, Forward Power = 161.7 nW
        retlw   0x11            ; High Byte, ADRES = 33, Forward Power = 177.5 nW
        retlw   0x12            ; High Byte, ADRES = 37, Forward Power = 194.7 nW
        retlw   0x13            ; High Byte, ADRES = 41, Forward Power = 213.6 nW
        retlw   0x13            ; High Byte, ADRES = 45, Forward Power = 234.3 nW
        retlw   0x14            ; High Byte, ADRES = 49, Forward Power = 257.1 nW
        retlw   0x14            ; High Byte, ADRES = 53, Forward Power = 282.1 nW
        retlw   0x15            ; High Byte, ADRES = 57, Forward Power = 309.5 nW
        retlw   0x15            ; High Byte, ADRES = 61, Forward Power = 339.5 nW
        retlw   0x16            ; High Byte, ADRES = 65, Forward Power = 372.5 nW
        retlw   0x16            ; High Byte, ADRES = 69, Forward Power = 408.6 nW
        retlw   0x17            ; High Byte, ADRES = 73, Forward Power = 448.3 nW
        retlw   0x18            ; High Byte, ADRES = 77, Forward Power = 491.9 nW
        retlw   0x18            ; High Byte, ADRES = 81, Forward Power = 539.6 nW
        retlw   0x18            ; High Byte, ADRES = 85, Forward Power = 592.0 nW
        retlw   0x19            ; High Byte, ADRES = 89, Forward Power = 649.5 nW
        retlw   0x19            ; High Byte, ADRES = 93, Forward Power = 712.6 nW
        retlw   0x1A            ; High Byte, ADRES = 97, Forward Power = 781.8 nW
        retlw   0x1B            ; High Byte, ADRES = 101, Forward Power = 857.8 nW
        retlw   0x1B            ; High Byte, ADRES = 105, Forward Power = 941.1 nW
        retlw   0x1C            ; High Byte, ADRES = 109, Forward Power = 1.032 uW
        retlw   0x1C            ; High Byte, ADRES = 113, Forward Power = 1.133 uW
        retlw   0x1D            ; High Byte, ADRES = 117, Forward Power = 1.243 uW
        retlw   0x1D            ; High Byte, ADRES = 121, Forward Power = 1.363 uW
        retlw   0x1E            ; High Byte, ADRES = 125, Forward Power = 1.496 uW
        retlw   0x1E            ; High Byte, ADRES = 129, Forward Power = 1.641 uW
        retlw   0x1F            ; High Byte, ADRES = 133, Forward Power = 1.800 uW
        retlw   0x20            ; High Byte, ADRES = 137, Forward Power = 1.975 uW
        retlw   0x20            ; High Byte, ADRES = 141, Forward Power = 2.167 uW
        retlw   0x20            ; High Byte, ADRES = 145, Forward Power = 2.378 uW
        retlw   0x21            ; High Byte, ADRES = 149, Forward Power = 2.608 uW
        retlw   0x22            ; High Byte, ADRES = 153, Forward Power = 2.862 uW
        retlw   0x22            ; High Byte, ADRES = 157, Forward Power = 3.140 uW
        retlw   0x23            ; High Byte, ADRES = 161, Forward Power = 3.445 uW
        retlw   0x23            ; High Byte, ADRES = 165, Forward Power = 3.779 uW
        retlw   0x24            ; High Byte, ADRES = 169, Forward Power = 4.146 uW
        retlw   0x24            ; High Byte, ADRES = 173, Forward Power = 4.549 uW
        retlw   0x25            ; High Byte, ADRES = 177, Forward Power = 4.991 uW
        retlw   0x25            ; High Byte, ADRES = 181, Forward Power = 5.475 uW
        retlw   0x26            ; High Byte, ADRES = 185, Forward Power = 6.007 uW
        retlw   0x26            ; High Byte, ADRES = 189, Forward Power = 6.590 uW
        retlw   0x27            ; High Byte, ADRES = 193, Forward Power = 7.230 uW
        retlw   0x28            ; High Byte, ADRES = 197, Forward Power = 7.933 uW
        retlw   0x28            ; High Byte, ADRES = 201, Forward Power = 8.703 uW
        retlw   0x29            ; High Byte, ADRES = 205, Forward Power = 9.548 uW
        retlw   0x29            ; High Byte, ADRES = 209, Forward Power = 10.48 uW
        retlw   0x2A            ; High Byte, ADRES = 213, Forward Power = 11.49 uW
        retlw   0x2A            ; High Byte, ADRES = 217, Forward Power = 12.61 uW
        retlw   0x2B            ; High Byte, ADRES = 221, Forward Power = 13.83 uW
        retlw   0x2B            ; High Byte, ADRES = 225, Forward Power = 15.18 uW
        retlw   0x2C            ; High Byte, ADRES = 229, Forward Power = 16.65 uW
        retlw   0x2C            ; High Byte, ADRES = 233, Forward Power = 18.27 uW
        retlw   0x2D            ; High Byte, ADRES = 237, Forward Power = 20.04 uW
        retlw   0x2D            ; High Byte, ADRES = 241, Forward Power = 21.99 uW
        retlw   0x2E            ; High Byte, ADRES = 245, Forward Power = 24.12 uW
        retlw   0x2E            ; High Byte, ADRES = 249, Forward Power = 26.47 uW
        retlw   0x2F            ; High Byte, ADRES = 253, Forward Power = 29.04 uW
        retlw   0x30            ; High Byte, ADRES = 257, Forward Power = 31.86 uW
        retlw   0x30            ; High Byte, ADRES = 261, Forward Power = 34.95 uW
        retlw   0x31            ; High Byte, ADRES = 265, Forward Power = 38.34 uW
        retlw   0x31            ; High Byte, ADRES = 269, Forward Power = 42.07 uW
        retlw   0x32            ; High Byte, ADRES = 273, Forward Power = 46.15 uW
        retlw   0x32            ; High Byte, ADRES = 277, Forward Power = 50.64 uW
```

```
retlw      0x33            ; High Byte, ADRES = 281, Forward Power = 55.55 uW
retlw      0x33            ; High Byte, ADRES = 285, Forward Power = 60.95 uW
retlw      0x34            ; High Byte, ADRES = 289, Forward Power = 66.87 uW
retlw      0x34            ; High Byte, ADRES = 293, Forward Power = 73.36 uW
retlw      0x35            ; High Byte, ADRES = 297, Forward Power = 80.48 uW
retlw      0x35            ; High Byte, ADRES = 301, Forward Power = 88.30 uW
retlw      0x36            ; High Byte, ADRES = 305, Forward Power = 96.88 uW
retlw      0x36            ; High Byte, ADRES = 309, Forward Power = 106.3 uW
retlw      0x37            ; High Byte, ADRES = 313, Forward Power = 116.6 uW
retlw      0x38            ; High Byte, ADRES = 317, Forward Power = 127.9 uW
retlw      0x38            ; High Byte, ADRES = 321, Forward Power = 140.4 uW
retlw      0x39            ; High Byte, ADRES = 325, Forward Power = 154.0 uW
retlw      0x39            ; High Byte, ADRES = 329, Forward Power = 168.9 uW
retlw      0x3A            ; High Byte, ADRES = 333, Forward Power = 185.3 uW
retlw      0x3A            ; High Byte, ADRES = 337, Forward Power = 203.3 uW
retlw      0x3B            ; High Byte, ADRES = 341, Forward Power = 223.1 uW
retlw      0x3C            ; High Byte, ADRES = 345, Forward Power = 244.8 uW
retlw      0x3C            ; High Byte, ADRES = 349, Forward Power = 268.5 uW
retlw      0x3C            ; High Byte, ADRES = 353, Forward Power = 294.6 uW
retlw      0x3D            ; High Byte, ADRES = 357, Forward Power = 323.2 uW
retlw      0x3D            ; High Byte, ADRES = 361, Forward Power = 354.6 uW
retlw      0x3E            ; High Byte, ADRES = 365, Forward Power = 389.0 uW
retlw      0x3E            ; High Byte, ADRES = 369, Forward Power = 426.8 uW
retlw      0x3F            ; High Byte, ADRES = 373, Forward Power = 468.3 uW
retlw      0x40            ; High Byte, ADRES = 377, Forward Power = 513.7 uW
retlw      0x40            ; High Byte, ADRES = 381, Forward Power = 563.6 uW
retlw      0x41            ; High Byte, ADRES = 385, Forward Power = 618.4 uW
retlw      0x41            ; High Byte, ADRES = 389, Forward Power = 678.4 uW
retlw      0x42            ; High Byte, ADRES = 393, Forward Power = 744.3 uW
retlw      0x42            ; High Byte, ADRES = 397, Forward Power = 816.6 uW
retlw      0x43            ; High Byte, ADRES = 401, Forward Power = 895.9 uW
retlw      0x44            ; High Byte, ADRES = 405, Forward Power = 982.9 uW
retlw      0x44            ; High Byte, ADRES = 409, Forward Power = 1.078 mW
retlw      0x44            ; High Byte, ADRES = 413, Forward Power = 1.183 mW
retlw      0x45            ; High Byte, ADRES = 417, Forward Power = 1.298 mW
retlw      0x45            ; High Byte, ADRES = 421, Forward Power = 1.424 mW
retlw      0x46            ; High Byte, ADRES = 425, Forward Power = 1.562 mW
retlw      0x47            ; High Byte, ADRES = 429, Forward Power = 1.714 mW
retlw      0x47            ; High Byte, ADRES = 433, Forward Power = 1.881 mW
retlw      0x48            ; High Byte, ADRES = 437, Forward Power = 2.063 mW
retlw      0x48            ; High Byte, ADRES = 441, Forward Power = 2.264 mW
retlw      0x49            ; High Byte, ADRES = 445, Forward Power = 2.483 mW
retlw      0x49            ; High Byte, ADRES = 449, Forward Power = 2.725 mW
retlw      0x4A            ; High Byte, ADRES = 453, Forward Power = 2.989 mW
retlw      0x4A            ; High Byte, ADRES = 457, Forward Power = 3.279 mW
retlw      0x4B            ; High Byte, ADRES = 461, Forward Power = 3.598 mW
retlw      0x4C            ; High Byte, ADRES = 465, Forward Power = 3.947 mW
retlw      0x4C            ; High Byte, ADRES = 469, Forward Power = 4.331 mW
retlw      0x4C            ; High Byte, ADRES = 473, Forward Power = 4.751 mW
retlw      0x4D            ; High Byte, ADRES = 477, Forward Power = 5.213 mW
retlw      0x4D            ; High Byte, ADRES = 481, Forward Power = 5.719 mW
retlw      0x4E            ; High Byte, ADRES = 485, Forward Power = 6.274 mW
retlw      0x4F            ; High Byte, ADRES = 489, Forward Power = 6.883 mW
retlw      0x4F            ; High Byte, ADRES = 493, Forward Power = 7.552 mW
retlw      0x50            ; High Byte, ADRES = 497, Forward Power = 8.285 mW
retlw      0x50            ; High Byte, ADRES = 501, Forward Power = 9.090 mW
retlw      0x51            ; High Byte, ADRES = 505, Forward Power = 9.973 mW
retlw      0x51            ; High Byte, ADRES = 509, Forward Power = 10.94 mW
retlw      0x52            ; High Byte, ADRES = 513, Forward Power = 12.00 mW
retlw      0x52            ; High Byte, ADRES = 517, Forward Power = 13.17 mW
retlw      0x53            ; High Byte, ADRES = 521, Forward Power = 14.45 mW
retlw      0x54            ; High Byte, ADRES = 525, Forward Power = 15.85 mW
retlw      0x54            ; High Byte, ADRES = 529, Forward Power = 17.39 mW
retlw      0x54            ; High Byte, ADRES = 533, Forward Power = 19.08 mW
retlw      0x55            ; High Byte, ADRES = 537, Forward Power = 20.93 mW
retlw      0x55            ; High Byte, ADRES = 541, Forward Power = 22.97 mW
retlw      0x56            ; High Byte, ADRES = 545, Forward Power = 25.20 mW
retlw      0x57            ; High Byte, ADRES = 549, Forward Power = 27.64 mW
retlw      0x57            ; High Byte, ADRES = 553, Forward Power = 30.33 mW
retlw      0x58            ; High Byte, ADRES = 557, Forward Power = 33.27 mW
retlw      0x58            ; High Byte, ADRES = 561, Forward Power = 36.50 mW
retlw      0x59            ; High Byte, ADRES = 565, Forward Power = 40.05 mW
retlw      0x59            ; High Byte, ADRES = 569, Forward Power = 43.94 mW
retlw      0x5A            ; High Byte, ADRES = 573, Forward Power = 48.21 mW
retlw      0x5A            ; High Byte, ADRES = 577, Forward Power = 52.89 mW
retlw      0x5B            ; High Byte, ADRES = 581, Forward Power = 58.02 mW
retlw      0x5C            ; High Byte, ADRES = 585, Forward Power = 63.66 mW
retlw      0x5C            ; High Byte, ADRES = 589, Forward Power = 69.84 mW
retlw      0x5C            ; High Byte, ADRES = 593, Forward Power = 76.62 mW
retlw      0x5D            ; High Byte, ADRES = 597, Forward Power = 84.06 mW
retlw      0x5D            ; High Byte, ADRES = 601, Forward Power = 92.23 mW
```

```
retlw      0x5E            ; High Byte, ADRES = 605, Forward Power = 101.2 mW
retlw      0x5F            ; High Byte, ADRES = 609, Forward Power = 111.0 mW
retlw      0x5F            ; High Byte, ADRES = 613, Forward Power = 121.8 mW
retlw      0x60            ; High Byte, ADRES = 617, Forward Power = 133.6 mW
retlw      0x60            ; High Byte, ADRES = 621, Forward Power = 146.6 mW
retlw      0x61            ; High Byte, ADRES = 625, Forward Power = 160.8 mW
retlw      0x61            ; High Byte, ADRES = 629, Forward Power = 176.5 mW
retlw      0x62            ; High Byte, ADRES = 633, Forward Power = 193.6 mW
retlw      0x62            ; High Byte, ADRES = 637, Forward Power = 212.4 mW
retlw      0x63            ; High Byte, ADRES = 641, Forward Power = 233.0 mW
retlw      0x64            ; High Byte, ADRES = 645, Forward Power = 255.6 mW
retlw      0x64            ; High Byte, ADRES = 649, Forward Power = 280.5 mW
retlw      0x64            ; High Byte, ADRES = 653, Forward Power = 307.7 mW
retlw      0x65            ; High Byte, ADRES = 657, Forward Power = 337.6 mW
retlw      0x65            ; High Byte, ADRES = 661, Forward Power = 370.4 mW
retlw      0x66            ; High Byte, ADRES = 665, Forward Power = 406.3 mW
retlw      0x67            ; High Byte, ADRES = 669, Forward Power = 445.8 mW
retlw      0x67            ; High Byte, ADRES = 673, Forward Power = 489.1 mW
retlw      0x68            ; High Byte, ADRES = 677, Forward Power = 536.6 mW
retlw      0x68            ; High Byte, ADRES = 681, Forward Power = 588.7 mW
retlw      0x69            ; High Byte, ADRES = 685, Forward Power = 645.9 mW
retlw      0x69            ; High Byte, ADRES = 689, Forward Power = 708.6 mW
retlw      0x6A            ; High Byte, ADRES = 693, Forward Power = 777.4 mW
retlw      0x6A            ; High Byte, ADRES = 697, Forward Power = 852.9 mW
retlw      0x6B            ; High Byte, ADRES = 701, Forward Power = 935.8 mW
retlw      0x6C            ; High Byte, ADRES = 705, Forward Power = 1.027 W
retlw      0x6C            ; High Byte, ADRES = 709, Forward Power = 1.126 W
retlw      0x6C            ; High Byte, ADRES = 713, Forward Power = 1.236 W
retlw      0x6D            ; High Byte, ADRES = 717, Forward Power = 1.356 W
retlw      0x6D            ; High Byte, ADRES = 721, Forward Power = 1.487 W
retlw      0x6E            ; High Byte, ADRES = 725, Forward Power = 1.632 W
retlw      0x6F            ; High Byte, ADRES = 729, Forward Power = 1.790 W
retlw      0x6F            ; High Byte, ADRES = 733, Forward Power = 1.964 W
retlw      0x70            ; High Byte, ADRES = 737, Forward Power = 2.155 W
retlw      0x70            ; High Byte, ADRES = 741, Forward Power = 2.364 W
retlw      0x71            ; High Byte, ADRES = 745, Forward Power = 2.594 W
retlw      0x71            ; High Byte, ADRES = 749, Forward Power = 2.846 W
retlw      0x72            ; High Byte, ADRES = 753, Forward Power = 3.122 W
retlw      0x72            ; High Byte, ADRES = 757, Forward Power = 3.425 W
retlw      0x73            ; High Byte, ADRES = 761, Forward Power = 3.758 W
retlw      0x74            ; High Byte, ADRES = 765, Forward Power = 4.123 W
retlw      0x74            ; High Byte, ADRES = 769, Forward Power = 4.523 W
retlw      0x74            ; High Byte, ADRES = 773, Forward Power = 4.962 W
retlw      0x75            ; High Byte, ADRES = 777, Forward Power = 5.444 W
retlw      0x75            ; High Byte, ADRES = 781, Forward Power = 5.973 W
retlw      0x76            ; High Byte, ADRES = 785, Forward Power = 6.553 W
retlw      0x77            ; High Byte, ADRES = 789, Forward Power = 7.190 W
retlw      0x77            ; High Byte, ADRES = 793, Forward Power = 7.888 W
retlw      0x78            ; High Byte, ADRES = 797, Forward Power = 8.654 W
retlw      0x78            ; High Byte, ADRES = 801, Forward Power = 9.494 W
retlw      0x79            ; High Byte, ADRES = 805, Forward Power = 10.42 W
retlw      0x79            ; High Byte, ADRES = 809, Forward Power = 11.43 W
retlw      0x7A            ; High Byte, ADRES = 813, Forward Power = 12.54 W
retlw      0x7A            ; High Byte, ADRES = 817, Forward Power = 13.76 W
retlw      0x7B            ; High Byte, ADRES = 821, Forward Power = 15.09 W
retlw      0x7C            ; High Byte, ADRES = 825, Forward Power = 16.56 W
retlw      0x7C            ; High Byte, ADRES = 829, Forward Power = 18.16 W
retlw      0x7C            ; High Byte, ADRES = 833, Forward Power = 19.93 W
retlw      0x7D            ; High Byte, ADRES = 837, Forward Power = 21.86 W
retlw      0x7D            ; High Byte, ADRES = 841, Forward Power = 23.99 W
retlw      0x7E            ; High Byte, ADRES = 845, Forward Power = 26.32 W
retlw      0x7F            ; High Byte, ADRES = 849, Forward Power = 28.87 W
retlw      0x7F            ; High Byte, ADRES = 853, Forward Power = 31.68 W
retlw      0x80            ; High Byte, ADRES = 857, Forward Power = 34.75 W
retlw      0x80            ; High Byte, ADRES = 861, Forward Power = 38.13 W
retlw      0x81            ; High Byte, ADRES = 865, Forward Power = 41.83 W
retlw      0x81            ; High Byte, ADRES = 869, Forward Power = 45.89 W
retlw      0x82            ; High Byte, ADRES = 873, Forward Power = 50.35 W
retlw      0x82            ; High Byte, ADRES = 877, Forward Power = 55.24 W
retlw      0x83            ; High Byte, ADRES = 881, Forward Power = 60.60 W
retlw      0x84            ; High Byte, ADRES = 885, Forward Power = 66.49 W
retlw      0x84            ; High Byte, ADRES = 889, Forward Power = 72.95 W
retlw      0x85            ; High Byte, ADRES = 893, Forward Power = 80.03 W
retlw      0x85            ; High Byte, ADRES = 897, Forward Power = 87.80 W
retlw      0x86            ; High Byte, ADRES = 901, Forward Power = 96.33 W
retlw      0x86            ; High Byte, ADRES = 905, Forward Power = 105.7 W
retlw      0x87            ; High Byte, ADRES = 909, Forward Power = 115.9 W
retlw      0x87            ; High Byte, ADRES = 913, Forward Power = 127.2 W
retlw      0x88            ; High Byte, ADRES = 917, Forward Power = 139.6 W
retlw      0x88            ; High Byte, ADRES = 921, Forward Power = 153.1 W
retlw      0x89            ; High Byte, ADRES = 925, Forward Power = 168.0 W
```

```
        retlw   0x89            ; High Byte, ADRES = 929, Forward Power = 184.3 W
        retlw   0x8A            ; High Byte, ADRES = 933, Forward Power = 202.2 W
        retlw   0x8A            ; High Byte, ADRES = 937, Forward Power = 221.8 W
        retlw   0x8B            ; High Byte, ADRES = 941, Forward Power = 243.4 W
        retlw   0x8C            ; High Byte, ADRES = 945, Forward Power = 267.0 W
        retlw   0x8C            ; High Byte, ADRES = 949, Forward Power = 292.9 W
        retlw   0x8D            ; High Byte, ADRES = 953, Forward Power = 321.4 W
        retlw   0x8D            ; High Byte, ADRES = 957, Forward Power = 352.6 W
        retlw   0x8E            ; High Byte, ADRES = 961, Forward Power = 386.9 W
        retlw   0x8E            ; High Byte, ADRES = 965, Forward Power = 424.4 W
        retlw   0x8F            ; High Byte, ADRES = 969, Forward Power = 465.6 W
        retlw   0x8F            ; High Byte, ADRES = 973, Forward Power = 510.9 W
        retlw   0x90            ; High Byte, ADRES = 977, Forward Power = 560.5 W
        retlw   0x90            ; High Byte, ADRES = 981, Forward Power = 614.9 W
        retlw   0x91            ; High Byte, ADRES = 985, Forward Power = 674.6 W
        retlw   0x91            ; High Byte, ADRES = 989, Forward Power = 740.1 W
        retlw   0x92            ; High Byte, ADRES = 993, Forward Power = 812.0 W
        retlw   0x92            ; High Byte, ADRES = 997, Forward Power = 890.9 W
        retlw   0x93            ; High Byte, ADRES = 1001, Forward Power = 977.4 W
        retlw   0x94            ; High Byte, ADRES = 1005, Forward Power = 1,072.3 W
        retlw   0x94            ; High Byte, ADRES = 1009, Forward Power = 1,176.4 W
        retlw   0x95            ; High Byte, ADRES = 1013, Forward Power = 1,290.7 W
        retlw   0x95            ; High Byte, ADRES = 1017, Forward Power = 1,416.0 W
        retlw   0x96            ; High Byte, ADRES = 1021, Forward Power = 1,553.5 W

FHighByteTable2
        retlw   0x0D            ; High Byte, ADRES = 2, Forward Power = 86.52 nW
        retlw   0x0E            ; High Byte, ADRES = 6, Forward Power = 94.92 nW
        retlw   0x0E            ; High Byte, ADRES = 10, Forward Power = 104.1 nW
        retlw   0x0F            ; High Byte, ADRES = 14, Forward Power = 114.3 nW
        retlw   0x10            ; High Byte, ADRES = 18, Forward Power = 125.4 nW
        retlw   0x10            ; High Byte, ADRES = 22, Forward Power = 137.5 nW
        retlw   0x11            ; High Byte, ADRES = 26, Forward Power = 150.9 nW
        retlw   0x11            ; High Byte, ADRES = 30, Forward Power = 165.5 nW
        retlw   0x12            ; High Byte, ADRES = 34, Forward Power = 181.6 nW
        retlw   0x12            ; High Byte, ADRES = 38, Forward Power = 199.2 nW
        retlw   0x13            ; High Byte, ADRES = 42, Forward Power = 218.6 nW
        retlw   0x14            ; High Byte, ADRES = 46, Forward Power = 239.8 nW
        retlw   0x14            ; High Byte, ADRES = 50, Forward Power = 263.1 nW
        retlw   0x14            ; High Byte, ADRES = 54, Forward Power = 288.7 nW
        retlw   0x15            ; High Byte, ADRES = 58, Forward Power = 316.7 nW
        retlw   0x15            ; High Byte, ADRES = 62, Forward Power = 347.5 nW
        retlw   0x16            ; High Byte, ADRES = 66, Forward Power = 381.2 nW
        retlw   0x17            ; High Byte, ADRES = 70, Forward Power = 418.2 nW
        retlw   0x17            ; High Byte, ADRES = 74, Forward Power = 458.8 nW
        retlw   0x18            ; High Byte, ADRES = 78, Forward Power = 503.4 nW
        retlw   0x18            ; High Byte, ADRES = 82, Forward Power = 552.3 nW
        retlw   0x19            ; High Byte, ADRES = 86, Forward Power = 605.9 nW
        retlw   0x19            ; High Byte, ADRES = 90, Forward Power = 664.8 nW
        retlw   0x1A            ; High Byte, ADRES = 94, Forward Power = 729.3 nW
        retlw   0x1A            ; High Byte, ADRES = 98, Forward Power = 800.2 nW
        retlw   0x1B            ; High Byte, ADRES = 102, Forward Power = 877.9 nW
        retlw   0x1C            ; High Byte, ADRES = 106, Forward Power = 963.1 nW
        retlw   0x1C            ; High Byte, ADRES = 110, Forward Power = 1.057 uW
        retlw   0x1C            ; High Byte, ADRES = 114, Forward Power = 1.159 uW
        retlw   0x1D            ; High Byte, ADRES = 118, Forward Power = 1.272 uW
        retlw   0x1D            ; High Byte, ADRES = 122, Forward Power = 1.395 uW
        retlw   0x1E            ; High Byte, ADRES = 126, Forward Power = 1.531 uW
        retlw   0x1F            ; High Byte, ADRES = 130, Forward Power = 1.680 uW
        retlw   0x1F            ; High Byte, ADRES = 134, Forward Power = 1.843 uW
        retlw   0x20            ; High Byte, ADRES = 138, Forward Power = 2.022 uW
        retlw   0x20            ; High Byte, ADRES = 142, Forward Power = 2.218 uW
        retlw   0x21            ; High Byte, ADRES = 146, Forward Power = 2.433 uW
        retlw   0x21            ; High Byte, ADRES = 150, Forward Power = 2.670 uW
        retlw   0x22            ; High Byte, ADRES = 154, Forward Power = 2.929 uW
        retlw   0x22            ; High Byte, ADRES = 158, Forward Power = 3.213 uW
        retlw   0x23            ; High Byte, ADRES = 162, Forward Power = 3.525 uW
        retlw   0x24            ; High Byte, ADRES = 166, Forward Power = 3.868 uW
        retlw   0x24            ; High Byte, ADRES = 170, Forward Power = 4.243 uW
        retlw   0x24            ; High Byte, ADRES = 174, Forward Power = 4.655 uW
        retlw   0x25            ; High Byte, ADRES = 178, Forward Power = 5.108 uW
        retlw   0x25            ; High Byte, ADRES = 182, Forward Power = 5.604 uW
        retlw   0x26            ; High Byte, ADRES = 186, Forward Power = 6.148 uW
        retlw   0x27            ; High Byte, ADRES = 190, Forward Power = 6.745 uW
        retlw   0x27            ; High Byte, ADRES = 194, Forward Power = 7.400 uW
        retlw   0x28            ; High Byte, ADRES = 198, Forward Power = 8.118 uW
        retlw   0x28            ; High Byte, ADRES = 202, Forward Power = 8.907 uW
        retlw   0x29            ; High Byte, ADRES = 206, Forward Power = 9.772 uW
        retlw   0x29            ; High Byte, ADRES = 210, Forward Power = 10.72 uW
        retlw   0x2A            ; High Byte, ADRES = 214, Forward Power = 11.76 uW
        retlw   0x2A            ; High Byte, ADRES = 218, Forward Power = 12.90 uW
```

```
retlw      0x2B            ; High Byte, ADRES = 222, Forward Power = 14.16 uW
retlw      0x2C            ; High Byte, ADRES = 226, Forward Power = 15.53 uW
retlw      0x2C            ; High Byte, ADRES = 230, Forward Power = 17.04 uW
retlw      0x2C            ; High Byte, ADRES = 234, Forward Power = 18.70 uW
retlw      0x2D            ; High Byte, ADRES = 238, Forward Power = 20.51 uW
retlw      0x2D            ; High Byte, ADRES = 242, Forward Power = 22.50 uW
retlw      0x2E            ; High Byte, ADRES = 246, Forward Power = 24.69 uW
retlw      0x2F            ; High Byte, ADRES = 250, Forward Power = 27.09 uW
retlw      0x2F            ; High Byte, ADRES = 254, Forward Power = 29.72 uW
retlw      0x30            ; High Byte, ADRES = 258, Forward Power = 32.60 uW
retlw      0x30            ; High Byte, ADRES = 262, Forward Power = 35.77 uW
retlw      0x31            ; High Byte, ADRES = 266, Forward Power = 39.24 uW
retlw      0x31            ; High Byte, ADRES = 270, Forward Power = 43.05 uW
retlw      0x32            ; High Byte, ADRES = 274, Forward Power = 47.23 uW
retlw      0x32            ; High Byte, ADRES = 278, Forward Power = 51.82 uW
retlw      0x33            ; High Byte, ADRES = 282, Forward Power = 56.85 uW
retlw      0x34            ; High Byte, ADRES = 286, Forward Power = 62.38 uW
retlw      0x34            ; High Byte, ADRES = 290, Forward Power = 68.43 uW
retlw      0x34            ; High Byte, ADRES = 294, Forward Power = 75.08 uW
retlw      0x35            ; High Byte, ADRES = 298, Forward Power = 82.37 uW
retlw      0x35            ; High Byte, ADRES = 302, Forward Power = 90.37 uW
retlw      0x36            ; High Byte, ADRES = 306, Forward Power = 99.15 uW
retlw      0x37            ; High Byte, ADRES = 310, Forward Power = 108.8 uW
retlw      0x37            ; High Byte, ADRES = 314, Forward Power = 119.3 uW
retlw      0x38            ; High Byte, ADRES = 318, Forward Power = 130.9 uW
retlw      0x38            ; High Byte, ADRES = 322, Forward Power = 143.6 uW
retlw      0x39            ; High Byte, ADRES = 326, Forward Power = 157.6 uW
retlw      0x39            ; High Byte, ADRES = 330, Forward Power = 172.9 uW
retlw      0x3A            ; High Byte, ADRES = 334, Forward Power = 189.7 uW
retlw      0x3A            ; High Byte, ADRES = 338, Forward Power = 208.1 uW
retlw      0x3B            ; High Byte, ADRES = 342, Forward Power = 228.3 uW
retlw      0x3C            ; High Byte, ADRES = 346, Forward Power = 250.5 uW
retlw      0x3C            ; High Byte, ADRES = 350, Forward Power = 274.8 uW
retlw      0x3C            ; High Byte, ADRES = 354, Forward Power = 301.5 uW
retlw      0x3D            ; High Byte, ADRES = 358, Forward Power = 330.8 uW
retlw      0x3D            ; High Byte, ADRES = 362, Forward Power = 362.9 uW
retlw      0x3E            ; High Byte, ADRES = 366, Forward Power = 398.2 uW
retlw      0x3F            ; High Byte, ADRES = 370, Forward Power = 436.8 uW
retlw      0x3F            ; High Byte, ADRES = 374, Forward Power = 479.2 uW
retlw      0x40            ; High Byte, ADRES = 378, Forward Power = 525.8 uW
retlw      0x40            ; High Byte, ADRES = 382, Forward Power = 576.9 uW
retlw      0x41            ; High Byte, ADRES = 386, Forward Power = 632.9 uW
retlw      0x41            ; High Byte, ADRES = 390, Forward Power = 694.3 uW
retlw      0x42            ; High Byte, ADRES = 394, Forward Power = 761.8 uW
retlw      0x42            ; High Byte, ADRES = 398, Forward Power = 835.7 uW
retlw      0x43            ; High Byte, ADRES = 402, Forward Power = 916.9 uW
retlw      0x44            ; High Byte, ADRES = 406, Forward Power = 1.006 mW
retlw      0x44            ; High Byte, ADRES = 410, Forward Power = 1.104 mW
retlw      0x44            ; High Byte, ADRES = 414, Forward Power = 1.211 mW
retlw      0x45            ; High Byte, ADRES = 418, Forward Power = 1.328 mW
retlw      0x45            ; High Byte, ADRES = 422, Forward Power = 1.457 mW
retlw      0x46            ; High Byte, ADRES = 426, Forward Power = 1.599 mW
retlw      0x47            ; High Byte, ADRES = 430, Forward Power = 1.754 mW
retlw      0x47            ; High Byte, ADRES = 434, Forward Power = 1.925 mW
retlw      0x48            ; High Byte, ADRES = 438, Forward Power = 2.112 mW
retlw      0x48            ; High Byte, ADRES = 442, Forward Power = 2.317 mW
retlw      0x49            ; High Byte, ADRES = 446, Forward Power = 2.542 mW
retlw      0x49            ; High Byte, ADRES = 450, Forward Power = 2.788 mW
retlw      0x4A            ; High Byte, ADRES = 454, Forward Power = 3.059 mW
retlw      0x4A            ; High Byte, ADRES = 458, Forward Power = 3.356 mW
retlw      0x4B            ; High Byte, ADRES = 462, Forward Power = 3.682 mW
retlw      0x4C            ; High Byte, ADRES = 466, Forward Power = 4.040 mW
retlw      0x4C            ; High Byte, ADRES = 470, Forward Power = 4.432 mW
retlw      0x4C            ; High Byte, ADRES = 474, Forward Power = 4.863 mW
retlw      0x4D            ; High Byte, ADRES = 478, Forward Power = 5.335 mW
retlw      0x4D            ; High Byte, ADRES = 482, Forward Power = 5.853 mW
retlw      0x4E            ; High Byte, ADRES = 486, Forward Power = 6.421 mW
retlw      0x4F            ; High Byte, ADRES = 490, Forward Power = 7.045 mW
retlw      0x4F            ; High Byte, ADRES = 494, Forward Power = 7.729 mW
retlw      0x50            ; High Byte, ADRES = 498, Forward Power = 8.480 mW
retlw      0x50            ; High Byte, ADRES = 502, Forward Power = 9.303 mW
retlw      0x51            ; High Byte, ADRES = 506, Forward Power = 10.21 mW
retlw      0x51            ; High Byte, ADRES = 510, Forward Power = 11.20 mW
retlw      0x52            ; High Byte, ADRES = 514, Forward Power = 12.29 mW
retlw      0x52            ; High Byte, ADRES = 518, Forward Power = 13.48 mW
retlw      0x53            ; High Byte, ADRES = 522, Forward Power = 14.79 mW
retlw      0x54            ; High Byte, ADRES = 526, Forward Power = 16.22 mW
retlw      0x54            ; High Byte, ADRES = 530, Forward Power = 17.80 mW
retlw      0x55            ; High Byte, ADRES = 534, Forward Power = 19.53 mW
retlw      0x55            ; High Byte, ADRES = 538, Forward Power = 21.42 mW
retlw      0x56            ; High Byte, ADRES = 542, Forward Power = 23.50 mW
```

```
retlw      0x56            ; High Byte, ADRES = 546, Forward Power = 25.79 mW
retlw      0x57            ; High Byte, ADRES = 550, Forward Power = 28.29 mW
retlw      0x57            ; High Byte, ADRES = 554, Forward Power = 31.04 mW
retlw      0x58            ; High Byte, ADRES = 558, Forward Power = 34.05 mW
retlw      0x58            ; High Byte, ADRES = 562, Forward Power = 37.36 mW
retlw      0x59            ; High Byte, ADRES = 566, Forward Power = 40.99 mW
retlw      0x59            ; High Byte, ADRES = 570, Forward Power = 44.97 mW
retlw      0x5A            ; High Byte, ADRES = 574, Forward Power = 49.34 mW
retlw      0x5A            ; High Byte, ADRES = 578, Forward Power = 54.13 mW
retlw      0x5B            ; High Byte, ADRES = 582, Forward Power = 59.38 mW
retlw      0x5C            ; High Byte, ADRES = 586, Forward Power = 65.15 mW
retlw      0x5C            ; High Byte, ADRES = 590, Forward Power = 71.48 mW
retlw      0x5D            ; High Byte, ADRES = 594, Forward Power = 78.42 mW
retlw      0x5D            ; High Byte, ADRES = 598, Forward Power = 86.03 mW
retlw      0x5E            ; High Byte, ADRES = 602, Forward Power = 94.39 mW
retlw      0x5E            ; High Byte, ADRES = 606, Forward Power = 103.6 mW
retlw      0x5F            ; High Byte, ADRES = 610, Forward Power = 113.6 mW
retlw      0x5F            ; High Byte, ADRES = 614, Forward Power = 124.6 mW
retlw      0x60            ; High Byte, ADRES = 618, Forward Power = 136.8 mW
retlw      0x60            ; High Byte, ADRES = 622, Forward Power = 150.0 mW
retlw      0x61            ; High Byte, ADRES = 626, Forward Power = 164.6 mW
retlw      0x61            ; High Byte, ADRES = 630, Forward Power = 180.6 mW
retlw      0x62            ; High Byte, ADRES = 634, Forward Power = 198.1 mW
retlw      0x62            ; High Byte, ADRES = 638, Forward Power = 217.4 mW
retlw      0x63            ; High Byte, ADRES = 642, Forward Power = 238.5 mW
retlw      0x64            ; High Byte, ADRES = 646, Forward Power = 261.6 mW
retlw      0x64            ; High Byte, ADRES = 650, Forward Power = 287.0 mW
retlw      0x65            ; High Byte, ADRES = 654, Forward Power = 314.9 mW
retlw      0x65            ; High Byte, ADRES = 658, Forward Power = 345.5 mW
retlw      0x66            ; High Byte, ADRES = 662, Forward Power = 379.1 mW
retlw      0x66            ; High Byte, ADRES = 666, Forward Power = 415.9 mW
retlw      0x67            ; High Byte, ADRES = 670, Forward Power = 456.3 mW
retlw      0x68            ; High Byte, ADRES = 674, Forward Power = 500.6 mW
retlw      0x68            ; High Byte, ADRES = 678, Forward Power = 549.2 mW
retlw      0x68            ; High Byte, ADRES = 682, Forward Power = 602.5 mW
retlw      0x69            ; High Byte, ADRES = 686, Forward Power = 661.0 mW
retlw      0x69            ; High Byte, ADRES = 690, Forward Power = 725.2 mW
retlw      0x6A            ; High Byte, ADRES = 694, Forward Power = 795.6 mW
retlw      0x6A            ; High Byte, ADRES = 698, Forward Power = 872.9 mW
retlw      0x6B            ; High Byte, ADRES = 702, Forward Power = 957.7 mW
retlw      0x6C            ; High Byte, ADRES = 706, Forward Power = 1.051 W
retlw      0x6C            ; High Byte, ADRES = 710, Forward Power = 1.153 W
retlw      0x6D            ; High Byte, ADRES = 714, Forward Power = 1.265 W
retlw      0x6D            ; High Byte, ADRES = 718, Forward Power = 1.388 W
retlw      0x6E            ; High Byte, ADRES = 722, Forward Power = 1.522 W
retlw      0x6E            ; High Byte, ADRES = 726, Forward Power = 1.670 W
retlw      0x6F            ; High Byte, ADRES = 730, Forward Power = 1.832 W
retlw      0x70            ; High Byte, ADRES = 734, Forward Power = 2.010 W
retlw      0x70            ; High Byte, ADRES = 738, Forward Power = 2.205 W
retlw      0x70            ; High Byte, ADRES = 742, Forward Power = 2.420 W
retlw      0x71            ; High Byte, ADRES = 746, Forward Power = 2.655 W
retlw      0x71            ; High Byte, ADRES = 750, Forward Power = 2.912 W
retlw      0x72            ; High Byte, ADRES = 754, Forward Power = 3.195 W
retlw      0x73            ; High Byte, ADRES = 758, Forward Power = 3.506 W
retlw      0x73            ; High Byte, ADRES = 762, Forward Power = 3.846 W
retlw      0x74            ; High Byte, ADRES = 766, Forward Power = 4.219 W
retlw      0x74            ; High Byte, ADRES = 770, Forward Power = 4.629 W
retlw      0x75            ; High Byte, ADRES = 774, Forward Power = 5.079 W
retlw      0x75            ; High Byte, ADRES = 778, Forward Power = 5.572 W
retlw      0x76            ; High Byte, ADRES = 782, Forward Power = 6.113 W
retlw      0x76            ; High Byte, ADRES = 786, Forward Power = 6.707 W
retlw      0x77            ; High Byte, ADRES = 790, Forward Power = 7.358 W
retlw      0x78            ; High Byte, ADRES = 794, Forward Power = 8.073 W
retlw      0x78            ; High Byte, ADRES = 798, Forward Power = 8.857 W
retlw      0x78            ; High Byte, ADRES = 802, Forward Power = 9.717 W
retlw      0x79            ; High Byte, ADRES = 806, Forward Power = 10.66 W
retlw      0x79            ; High Byte, ADRES = 810, Forward Power = 11.70 W
retlw      0x7A            ; High Byte, ADRES = 814, Forward Power = 12.83 W
retlw      0x7B            ; High Byte, ADRES = 818, Forward Power = 14.08 W
retlw      0x7B            ; High Byte, ADRES = 822, Forward Power = 15.44 W
retlw      0x7C            ; High Byte, ADRES = 826, Forward Power = 16.94 W
retlw      0x7C            ; High Byte, ADRES = 830, Forward Power = 18.59 W
retlw      0x7D            ; High Byte, ADRES = 834, Forward Power = 20.40 W
retlw      0x7D            ; High Byte, ADRES = 838, Forward Power = 22.38 W
retlw      0x7E            ; High Byte, ADRES = 842, Forward Power = 24.55 W
retlw      0x7E            ; High Byte, ADRES = 846, Forward Power = 26.93 W
retlw      0x7F            ; High Byte, ADRES = 850, Forward Power = 29.55 W
retlw      0x80            ; High Byte, ADRES = 854, Forward Power = 32.42 W
retlw      0x80            ; High Byte, ADRES = 858, Forward Power = 35.57 W
retlw      0x80            ; High Byte, ADRES = 862, Forward Power = 39.02 W
retlw      0x81            ; High Byte, ADRES = 866, Forward Power = 42.81 W
```

```
      retlw      0x81            ; High Byte, ADRES = 870, Forward Power = 46.97 W
      retlw      0x82            ; High Byte, ADRES = 874, Forward Power = 51.53 W
      retlw      0x83            ; High Byte, ADRES = 878, Forward Power = 56.53 W
      retlw      0x83            ; High Byte, ADRES = 882, Forward Power = 62.02 W
      retlw      0x84            ; High Byte, ADRES = 886, Forward Power = 68.05 W
      retlw      0x84            ; High Byte, ADRES = 890, Forward Power = 74.66 W
      retlw      0x85            ; High Byte, ADRES = 894, Forward Power = 81.91 W
      retlw      0x85            ; High Byte, ADRES = 898, Forward Power = 89.86 W
      retlw      0x86            ; High Byte, ADRES = 902, Forward Power = 98.59 W
      retlw      0x86            ; High Byte, ADRES = 906, Forward Power = 108.2 W
      retlw      0x87            ; High Byte, ADRES = 910, Forward Power = 118.7 W
      retlw      0x88            ; High Byte, ADRES = 914, Forward Power = 130.2 W
      retlw      0x88            ; High Byte, ADRES = 918, Forward Power = 142.8 W
      retlw      0x88            ; High Byte, ADRES = 922, Forward Power = 156.7 W
      retlw      0x89            ; High Byte, ADRES = 926, Forward Power = 171.9 W
      retlw      0x89            ; High Byte, ADRES = 930, Forward Power = 188.6 W
      retlw      0x8A            ; High Byte, ADRES = 934, Forward Power = 206.9 W
      retlw      0x8B            ; High Byte, ADRES = 938, Forward Power = 227.0 W
      retlw      0x8B            ; High Byte, ADRES = 942, Forward Power = 249.1 W
      retlw      0x8C            ; High Byte, ADRES = 946, Forward Power = 273.3 W
      retlw      0x8C            ; High Byte, ADRES = 950, Forward Power = 299.8 W
      retlw      0x8D            ; High Byte, ADRES = 954, Forward Power = 328.9 W
      retlw      0x8D            ; High Byte, ADRES = 958, Forward Power = 360.9 W
      retlw      0x8E            ; High Byte, ADRES = 962, Forward Power = 395.9 W
      retlw      0x8E            ; High Byte, ADRES = 966, Forward Power = 434.4 W
      retlw      0x8F            ; High Byte, ADRES = 970, Forward Power = 476.6 W
      retlw      0x90            ; High Byte, ADRES = 974, Forward Power = 522.8 W
      retlw      0x90            ; High Byte, ADRES = 978, Forward Power = 573.6 W
      retlw      0x90            ; High Byte, ADRES = 982, Forward Power = 629.3 W
      retlw      0x91            ; High Byte, ADRES = 986, Forward Power = 690.4 W
      retlw      0x91            ; High Byte, ADRES = 990, Forward Power = 757.5 W
      retlw      0x92            ; High Byte, ADRES = 994, Forward Power = 831.0 W
      retlw      0x93            ; High Byte, ADRES = 998, Forward Power = 911.7 W
      retlw      0x93            ; High Byte, ADRES = 1002, Forward Power = 1,000.3 W
      retlw      0x94            ; High Byte, ADRES = 1006, Forward Power = 1,097.4 W
      retlw      0x94            ; High Byte, ADRES = 1010, Forward Power = 1,204.0 W
      retlw      0x95            ; High Byte, ADRES = 1014, Forward Power = 1,320.9 W
      retlw      0x95            ; High Byte, ADRES = 1018, Forward Power = 1,449.2 W
      retlw      0x96            ; High Byte, ADRES = 1022, Forward Power = 1,590.0 W

FHighByteTable3
      retlw      0x0D            ; High Byte, ADRES = 3, Forward Power = 88.55 nW
      retlw      0x0E            ; High Byte, ADRES = 7, Forward Power = 97.15 nW
      retlw      0x0F            ; High Byte, ADRES = 11, Forward Power = 106.6 nW
      retlw      0x0F            ; High Byte, ADRES = 15, Forward Power = 116.9 nW
      retlw      0x10            ; High Byte, ADRES = 19, Forward Power = 128.3 nW
      retlw      0x10            ; High Byte, ADRES = 23, Forward Power = 140.8 nW
      retlw      0x11            ; High Byte, ADRES = 27, Forward Power = 154.4 nW
      retlw      0x11            ; High Byte, ADRES = 31, Forward Power = 169.4 nW
      retlw      0x12            ; High Byte, ADRES = 35, Forward Power = 185.9 nW
      retlw      0x12            ; High Byte, ADRES = 39, Forward Power = 203.9 nW
      retlw      0x13            ; High Byte, ADRES = 43, Forward Power = 223.7 nW
      retlw      0x14            ; High Byte, ADRES = 47, Forward Power = 245.4 nW
      retlw      0x14            ; High Byte, ADRES = 51, Forward Power = 269.3 nW
      retlw      0x14            ; High Byte, ADRES = 55, Forward Power = 295.4 nW
      retlw      0x15            ; High Byte, ADRES = 59, Forward Power = 324.1 nW
      retlw      0x15            ; High Byte, ADRES = 63, Forward Power = 355.6 nW
      retlw      0x16            ; High Byte, ADRES = 67, Forward Power = 390.1 nW
      retlw      0x17            ; High Byte, ADRES = 71, Forward Power = 428.0 nW
      retlw      0x17            ; High Byte, ADRES = 75, Forward Power = 469.6 nW
      retlw      0x18            ; High Byte, ADRES = 79, Forward Power = 515.2 nW
      retlw      0x18            ; High Byte, ADRES = 83, Forward Power = 565.2 nW
      retlw      0x19            ; High Byte, ADRES = 87, Forward Power = 620.1 nW
      retlw      0x19            ; High Byte, ADRES = 91, Forward Power = 680.4 nW
      retlw      0x1A            ; High Byte, ADRES = 95, Forward Power = 746.4 nW
      retlw      0x1A            ; High Byte, ADRES = 99, Forward Power = 818.9 nW
      retlw      0x1B            ; High Byte, ADRES = 103, Forward Power = 898.4 nW
      retlw      0x1C            ; High Byte, ADRES = 107, Forward Power = 985.7 nW
      retlw      0x1C            ; High Byte, ADRES = 111, Forward Power = 1.081 uW
      retlw      0x1C            ; High Byte, ADRES = 115, Forward Power = 1.186 uW
      retlw      0x1D            ; High Byte, ADRES = 119, Forward Power = 1.302 uW
      retlw      0x1D            ; High Byte, ADRES = 123, Forward Power = 1.428 uW
      retlw      0x1E            ; High Byte, ADRES = 127, Forward Power = 1.567 uW
      retlw      0x1F            ; High Byte, ADRES = 131, Forward Power = 1.719 uW
      retlw      0x1F            ; High Byte, ADRES = 135, Forward Power = 1.886 uW
      retlw      0x20            ; High Byte, ADRES = 139, Forward Power = 2.069 uW
      retlw      0x20            ; High Byte, ADRES = 143, Forward Power = 2.270 uW
      retlw      0x21            ; High Byte, ADRES = 147, Forward Power = 2.490 uW
      retlw      0x21            ; High Byte, ADRES = 151, Forward Power = 2.732 uW
      retlw      0x22            ; High Byte, ADRES = 155, Forward Power = 2.998 uW
      retlw      0x22            ; High Byte, ADRES = 159, Forward Power = 3.289 uW
```

```
retlw      0x23            ; High Byte, ADRES = 163, Forward Power = 3.608 uW
retlw      0x24            ; High Byte, ADRES = 167, Forward Power = 3.958 uW
retlw      0x24            ; High Byte, ADRES = 171, Forward Power = 4.343 uW
retlw      0x24            ; High Byte, ADRES = 175, Forward Power = 4.765 uW
retlw      0x25            ; High Byte, ADRES = 179, Forward Power = 5.227 uW
retlw      0x26            ; High Byte, ADRES = 183, Forward Power = 5.735 uW
retlw      0x26            ; High Byte, ADRES = 187, Forward Power = 6.292 uW
retlw      0x27            ; High Byte, ADRES = 191, Forward Power = 6.903 uW
retlw      0x27            ; High Byte, ADRES = 195, Forward Power = 7.573 uW
retlw      0x28            ; High Byte, ADRES = 199, Forward Power = 8.309 uW
retlw      0x28            ; High Byte, ADRES = 203, Forward Power = 9.116 uW
retlw      0x29            ; High Byte, ADRES = 207, Forward Power = 10.00 uW
retlw      0x29            ; High Byte, ADRES = 211, Forward Power = 10.97 uW
retlw      0x2A            ; High Byte, ADRES = 215, Forward Power = 12.04 uW
retlw      0x2A            ; High Byte, ADRES = 219, Forward Power = 13.21 uW
retlw      0x2B            ; High Byte, ADRES = 223, Forward Power = 14.49 uW
retlw      0x2C            ; High Byte, ADRES = 227, Forward Power = 15.90 uW
retlw      0x2C            ; High Byte, ADRES = 231, Forward Power = 17.44 uW
retlw      0x2D            ; High Byte, ADRES = 235, Forward Power = 19.13 uW
retlw      0x2D            ; High Byte, ADRES = 239, Forward Power = 20.99 uW
retlw      0x2E            ; High Byte, ADRES = 243, Forward Power = 23.03 uW
retlw      0x2E            ; High Byte, ADRES = 247, Forward Power = 25.27 uW
retlw      0x2F            ; High Byte, ADRES = 251, Forward Power = 27.72 uW
retlw      0x2F            ; High Byte, ADRES = 255, Forward Power = 30.41 uW
retlw      0x30            ; High Byte, ADRES = 259, Forward Power = 33.37 uW
retlw      0x30            ; High Byte, ADRES = 263, Forward Power = 36.61 uW
retlw      0x31            ; High Byte, ADRES = 267, Forward Power = 40.16 uW
retlw      0x31            ; High Byte, ADRES = 271, Forward Power = 44.06 uW
retlw      0x32            ; High Byte, ADRES = 275, Forward Power = 48.34 uW
retlw      0x32            ; High Byte, ADRES = 279, Forward Power = 53.04 uW
retlw      0x33            ; High Byte, ADRES = 283, Forward Power = 58.19 uW
retlw      0x34            ; High Byte, ADRES = 287, Forward Power = 63.84 uW
retlw      0x34            ; High Byte, ADRES = 291, Forward Power = 70.04 uW
retlw      0x35            ; High Byte, ADRES = 295, Forward Power = 76.84 uW
retlw      0x35            ; High Byte, ADRES = 299, Forward Power = 84.30 uW
retlw      0x36            ; High Byte, ADRES = 303, Forward Power = 92.49 uW
retlw      0x36            ; High Byte, ADRES = 307, Forward Power = 101.5 uW
retlw      0x37            ; High Byte, ADRES = 311, Forward Power = 111.3 uW
retlw      0x38            ; High Byte, ADRES = 315, Forward Power = 122.1 uW
retlw      0x38            ; High Byte, ADRES = 319, Forward Power = 134.0 uW
retlw      0x38            ; High Byte, ADRES = 323, Forward Power = 147.0 uW
retlw      0x39            ; High Byte, ADRES = 327, Forward Power = 161.3 uW
retlw      0x39            ; High Byte, ADRES = 331, Forward Power = 177.0 uW
retlw      0x3A            ; High Byte, ADRES = 335, Forward Power = 194.1 uW
retlw      0x3A            ; High Byte, ADRES = 339, Forward Power = 213.0 uW
retlw      0x3B            ; High Byte, ADRES = 343, Forward Power = 233.7 uW
retlw      0x3C            ; High Byte, ADRES = 347, Forward Power = 256.4 uW
retlw      0x3C            ; High Byte, ADRES = 351, Forward Power = 281.3 uW
retlw      0x3D            ; High Byte, ADRES = 355, Forward Power = 308.6 uW
retlw      0x3D            ; High Byte, ADRES = 359, Forward Power = 338.5 uW
retlw      0x3E            ; High Byte, ADRES = 363, Forward Power = 371.4 uW
retlw      0x3E            ; High Byte, ADRES = 367, Forward Power = 407.5 uW
retlw      0x3F            ; High Byte, ADRES = 371, Forward Power = 447.1 uW
retlw      0x40            ; High Byte, ADRES = 375, Forward Power = 490.5 uW
retlw      0x40            ; High Byte, ADRES = 379, Forward Power = 538.1 uW
retlw      0x40            ; High Byte, ADRES = 383, Forward Power = 590.4 uW
retlw      0x41            ; High Byte, ADRES = 387, Forward Power = 647.7 uW
retlw      0x41            ; High Byte, ADRES = 391, Forward Power = 710.6 uW
retlw      0x42            ; High Byte, ADRES = 395, Forward Power = 779.6 uW
retlw      0x43            ; High Byte, ADRES = 399, Forward Power = 855.3 uW
retlw      0x43            ; High Byte, ADRES = 403, Forward Power = 938.4 uW
retlw      0x44            ; High Byte, ADRES = 407, Forward Power = 1.030 mW
retlw      0x44            ; High Byte, ADRES = 411, Forward Power = 1.130 mW
retlw      0x45            ; High Byte, ADRES = 415, Forward Power = 1.239 mW
retlw      0x45            ; High Byte, ADRES = 419, Forward Power = 1.360 mW
retlw      0x46            ; High Byte, ADRES = 423, Forward Power = 1.492 mW
retlw      0x46            ; High Byte, ADRES = 427, Forward Power = 1.636 mW
retlw      0x47            ; High Byte, ADRES = 431, Forward Power = 1.795 mW
retlw      0x48            ; High Byte, ADRES = 435, Forward Power = 1.970 mW
retlw      0x48            ; High Byte, ADRES = 439, Forward Power = 2.161 mW
retlw      0x48            ; High Byte, ADRES = 443, Forward Power = 2.371 mW
retlw      0x49            ; High Byte, ADRES = 447, Forward Power = 2.601 mW
retlw      0x49            ; High Byte, ADRES = 451, Forward Power = 2.854 mW
retlw      0x4A            ; High Byte, ADRES = 455, Forward Power = 3.131 mW
retlw      0x4B            ; High Byte, ADRES = 459, Forward Power = 3.435 mW
retlw      0x4B            ; High Byte, ADRES = 463, Forward Power = 3.769 mW
retlw      0x4C            ; High Byte, ADRES = 467, Forward Power = 4.134 mW
retlw      0x4C            ; High Byte, ADRES = 471, Forward Power = 4.536 mW
retlw      0x4D            ; High Byte, ADRES = 475, Forward Power = 4.977 mW
retlw      0x4D            ; High Byte, ADRES = 479, Forward Power = 5.460 mW
retlw      0x4E            ; High Byte, ADRES = 483, Forward Power = 5.990 mW
```

```
retlw      0x4E            ; High Byte, ADRES = 487, Forward Power = 6.572 mW
retlw      0x4F            ; High Byte, ADRES = 491, Forward Power = 7.210 mW
retlw      0x50            ; High Byte, ADRES = 495, Forward Power = 7.910 mW
retlw      0x50            ; High Byte, ADRES = 499, Forward Power = 8.678 mW
retlw      0x50            ; High Byte, ADRES = 503, Forward Power = 9.521 mW
retlw      0x51            ; High Byte, ADRES = 507, Forward Power = 10.45 mW
retlw      0x51            ; High Byte, ADRES = 511, Forward Power = 11.46 mW
retlw      0x52            ; High Byte, ADRES = 515, Forward Power = 12.57 mW
retlw      0x53            ; High Byte, ADRES = 519, Forward Power = 13.79 mW
retlw      0x53            ; High Byte, ADRES = 523, Forward Power = 15.13 mW
retlw      0x54            ; High Byte, ADRES = 527, Forward Power = 16.60 mW
retlw      0x54            ; High Byte, ADRES = 531, Forward Power = 18.22 mW
retlw      0x55            ; High Byte, ADRES = 535, Forward Power = 19.99 mW
retlw      0x55            ; High Byte, ADRES = 539, Forward Power = 21.93 mW
retlw      0x56            ; High Byte, ADRES = 543, Forward Power = 24.06 mW
retlw      0x56            ; High Byte, ADRES = 547, Forward Power = 26.39 mW
retlw      0x57            ; High Byte, ADRES = 551, Forward Power = 28.95 mW
retlw      0x58            ; High Byte, ADRES = 555, Forward Power = 31.77 mW
retlw      0x58            ; High Byte, ADRES = 559, Forward Power = 34.85 mW
retlw      0x58            ; High Byte, ADRES = 563, Forward Power = 38.24 mW
retlw      0x59            ; High Byte, ADRES = 567, Forward Power = 41.95 mW
retlw      0x59            ; High Byte, ADRES = 571, Forward Power = 46.02 mW
retlw      0x5A            ; High Byte, ADRES = 575, Forward Power = 50.49 mW
retlw      0x5B            ; High Byte, ADRES = 579, Forward Power = 55.40 mW
retlw      0x5B            ; High Byte, ADRES = 583, Forward Power = 60.78 mW
retlw      0x5C            ; High Byte, ADRES = 587, Forward Power = 66.68 mW
retlw      0x5C            ; High Byte, ADRES = 591, Forward Power = 73.15 mW
retlw      0x5D            ; High Byte, ADRES = 595, Forward Power = 80.26 mW
retlw      0x5D            ; High Byte, ADRES = 599, Forward Power = 88.05 mW
retlw      0x5E            ; High Byte, ADRES = 603, Forward Power = 96.60 mW
retlw      0x5E            ; High Byte, ADRES = 607, Forward Power = 106.0 mW
retlw      0x5F            ; High Byte, ADRES = 611, Forward Power = 116.3 mW
retlw      0x60            ; High Byte, ADRES = 615, Forward Power = 127.6 mW
retlw      0x60            ; High Byte, ADRES = 619, Forward Power = 140.0 mW
retlw      0x60            ; High Byte, ADRES = 623, Forward Power = 153.5 mW
retlw      0x61            ; High Byte, ADRES = 627, Forward Power = 168.5 mW
retlw      0x61            ; High Byte, ADRES = 631, Forward Power = 184.8 mW
retlw      0x62            ; High Byte, ADRES = 635, Forward Power = 202.8 mW
retlw      0x63            ; High Byte, ADRES = 639, Forward Power = 222.5 mW
retlw      0x63            ; High Byte, ADRES = 643, Forward Power = 244.1 mW
retlw      0x64            ; High Byte, ADRES = 647, Forward Power = 267.8 mW
retlw      0x64            ; High Byte, ADRES = 651, Forward Power = 293.8 mW
retlw      0x65            ; High Byte, ADRES = 655, Forward Power = 322.3 mW
retlw      0x65            ; High Byte, ADRES = 659, Forward Power = 353.6 mW
retlw      0x66            ; High Byte, ADRES = 663, Forward Power = 387.9 mW
retlw      0x66            ; High Byte, ADRES = 667, Forward Power = 425.6 mW
retlw      0x67            ; High Byte, ADRES = 671, Forward Power = 467.0 mW
retlw      0x68            ; High Byte, ADRES = 675, Forward Power = 512.3 mW
retlw      0x68            ; High Byte, ADRES = 679, Forward Power = 562.1 mW
retlw      0x68            ; High Byte, ADRES = 683, Forward Power = 616.6 mW
retlw      0x69            ; High Byte, ADRES = 687, Forward Power = 676.5 mW
retlw      0x69            ; High Byte, ADRES = 691, Forward Power = 742.2 mW
retlw      0x6A            ; High Byte, ADRES = 695, Forward Power = 814.3 mW
retlw      0x6B            ; High Byte, ADRES = 699, Forward Power = 893.4 mW
retlw      0x6B            ; High Byte, ADRES = 703, Forward Power = 980.1 mW
retlw      0x6C            ; High Byte, ADRES = 707, Forward Power = 1.075 W
retlw      0x6C            ; High Byte, ADRES = 711, Forward Power = 1.180 W
retlw      0x6D            ; High Byte, ADRES = 715, Forward Power = 1.294 W
retlw      0x6D            ; High Byte, ADRES = 719, Forward Power = 1.420 W
retlw      0x6E            ; High Byte, ADRES = 723, Forward Power = 1.558 W
retlw      0x6E            ; High Byte, ADRES = 727, Forward Power = 1.709 W
retlw      0x6F            ; High Byte, ADRES = 731, Forward Power = 1.875 W
retlw      0x70            ; High Byte, ADRES = 735, Forward Power = 2.057 W
retlw      0x70            ; High Byte, ADRES = 739, Forward Power = 2.257 W
retlw      0x70            ; High Byte, ADRES = 743, Forward Power = 2.476 W
retlw      0x71            ; High Byte, ADRES = 747, Forward Power = 2.717 W
retlw      0x71            ; High Byte, ADRES = 751, Forward Power = 2.981 W
retlw      0x72            ; High Byte, ADRES = 755, Forward Power = 3.270 W
retlw      0x73            ; High Byte, ADRES = 759, Forward Power = 3.588 W
retlw      0x73            ; High Byte, ADRES = 763, Forward Power = 3.936 W
retlw      0x74            ; High Byte, ADRES = 767, Forward Power = 4.318 W
retlw      0x74            ; High Byte, ADRES = 771, Forward Power = 4.738 W
retlw      0x75            ; High Byte, ADRES = 775, Forward Power = 5.198 W
retlw      0x75            ; High Byte, ADRES = 779, Forward Power = 5.703 W
retlw      0x76            ; High Byte, ADRES = 783, Forward Power = 6.256 W
retlw      0x76            ; High Byte, ADRES = 787, Forward Power = 6.864 W
retlw      0x77            ; High Byte, ADRES = 791, Forward Power = 7.531 W
retlw      0x78            ; High Byte, ADRES = 795, Forward Power = 8.262 W
retlw      0x78            ; High Byte, ADRES = 799, Forward Power = 9.064 W
retlw      0x78            ; High Byte, ADRES = 803, Forward Power = 9.945 W
retlw      0x79            ; High Byte, ADRES = 807, Forward Power = 10.91 W
```

```
     retlw      0x79            ; High Byte, ADRES = 811, Forward Power = 11.97 W
     retlw      0x7A            ; High Byte, ADRES = 815, Forward Power = 13.13 W
     retlw      0x7B            ; High Byte, ADRES = 819, Forward Power = 14.41 W
     retlw      0x7B            ; High Byte, ADRES = 823, Forward Power = 15.81 W
     retlw      0x7C            ; High Byte, ADRES = 827, Forward Power = 17.34 W
     retlw      0x7C            ; High Byte, ADRES = 831, Forward Power = 19.03 W
     retlw      0x7D            ; High Byte, ADRES = 835, Forward Power = 20.87 W
     retlw      0x7D            ; High Byte, ADRES = 839, Forward Power = 22.90 W
     retlw      0x7E            ; High Byte, ADRES = 843, Forward Power = 25.13 W
     retlw      0x7E            ; High Byte, ADRES = 847, Forward Power = 27.57 W
     retlw      0x7F            ; High Byte, ADRES = 851, Forward Power = 30.24 W
     retlw      0x80            ; High Byte, ADRES = 855, Forward Power = 33.18 W
     retlw      0x80            ; High Byte, ADRES = 859, Forward Power = 36.40 W
     retlw      0x80            ; High Byte, ADRES = 863, Forward Power = 39.94 W
     retlw      0x81            ; High Byte, ADRES = 867, Forward Power = 43.81 W
     retlw      0x82            ; High Byte, ADRES = 871, Forward Power = 48.07 W
     retlw      0x82            ; High Byte, ADRES = 875, Forward Power = 52.74 W
     retlw      0x83            ; High Byte, ADRES = 879, Forward Power = 57.86 W
     retlw      0x83            ; High Byte, ADRES = 883, Forward Power = 63.48 W
     retlw      0x84            ; High Byte, ADRES = 887, Forward Power = 69.64 W
     retlw      0x84            ; High Byte, ADRES = 891, Forward Power = 76.41 W
     retlw      0x85            ; High Byte, ADRES = 895, Forward Power = 83.83 W
     retlw      0x85            ; High Byte, ADRES = 899, Forward Power = 91.97 W
     retlw      0x86            ; High Byte, ADRES = 903, Forward Power = 100.9 W
     retlw      0x86            ; High Byte, ADRES = 907, Forward Power = 110.7 W
     retlw      0x87            ; High Byte, ADRES = 911, Forward Power = 121.4 W
     retlw      0x88            ; High Byte, ADRES = 915, Forward Power = 133.2 W
     retlw      0x88            ; High Byte, ADRES = 919, Forward Power = 146.2 W
     retlw      0x89            ; High Byte, ADRES = 923, Forward Power = 160.4 W
     retlw      0x89            ; High Byte, ADRES = 927, Forward Power = 176.0 W
     retlw      0x8A            ; High Byte, ADRES = 931, Forward Power = 193.0 W
     retlw      0x8A            ; High Byte, ADRES = 935, Forward Power = 211.8 W
     retlw      0x8B            ; High Byte, ADRES = 939, Forward Power = 232.4 W
     retlw      0x8B            ; High Byte, ADRES = 943, Forward Power = 254.9 W
     retlw      0x8C            ; High Byte, ADRES = 947, Forward Power = 279.7 W
     retlw      0x8C            ; High Byte, ADRES = 951, Forward Power = 306.8 W
     retlw      0x8D            ; High Byte, ADRES = 955, Forward Power = 336.6 W
     retlw      0x8D            ; High Byte, ADRES = 959, Forward Power = 369.3 W
     retlw      0x8E            ; High Byte, ADRES = 963, Forward Power = 405.2 W
     retlw      0x8E            ; High Byte, ADRES = 967, Forward Power = 444.6 W
     retlw      0x8F            ; High Byte, ADRES = 971, Forward Power = 487.7 W
     retlw      0x90            ; High Byte, ADRES = 975, Forward Power = 535.1 W
     retlw      0x90            ; High Byte, ADRES = 979, Forward Power = 587.1 W
     retlw      0x91            ; High Byte, ADRES = 983, Forward Power = 644.1 W
     retlw      0x91            ; High Byte, ADRES = 987, Forward Power = 706.6 W
     retlw      0x92            ; High Byte, ADRES = 991, Forward Power = 775.2 W
     retlw      0x92            ; High Byte, ADRES = 995, Forward Power = 850.5 W
     retlw      0x93            ; High Byte, ADRES = 999, Forward Power = 933.1 W
     retlw      0x93            ; High Byte, ADRES = 1003, Forward Power = 1,023.7 W
     retlw      0x94            ; High Byte, ADRES = 1007, Forward Power = 1,123.2 W
     retlw      0x94            ; High Byte, ADRES = 1011, Forward Power = 1,232.2 W
     retlw      0x95            ; High Byte, ADRES = 1015, Forward Power = 1,351.9 W
     retlw      0x95            ; High Byte, ADRES = 1019, Forward Power = 1,483.2 W
     retlw      0x96            ; High Byte, ADRES = 1023, Forward Power = 1,627.2 W


;*****************************************************
; Lookup Table for Reflected (Channel 1) Power
;   Calibration dBm = .09965*ADC + -41.50


RLowByteTable0
     retlw      0xC0            ; Low Byte, ADRES = 0, Reflected Power = 70.79 nW
     retlw      0x35            ; Low Byte, ADRES = 4, Reflected Power = 77.60 nW
     retlw      0xB5            ; Low Byte, ADRES = 8, Reflected Power = 85.06 nW
     retlw      0x42            ; Low Byte, ADRES = 12, Reflected Power = 93.24 nW
     retlw      0xDC            ; Low Byte, ADRES = 16, Reflected Power = 102.2 nW
     retlw      0x85            ; Low Byte, ADRES = 20, Reflected Power = 112.0 nW
     retlw      0x1F            ; Low Byte, ADRES = 24, Reflected Power = 122.8 nW
     retlw      0x84            ; Low Byte, ADRES = 28, Reflected Power = 134.6 nW
     retlw      0xF3            ; Low Byte, ADRES = 32, Reflected Power = 147.5 nW
     retlw      0x6D            ; Low Byte, ADRES = 36, Reflected Power = 161.7 nW
     retlw      0xF3            ; Low Byte, ADRES = 40, Reflected Power = 177.3 nW
     retlw      0x85            ; Low Byte, ADRES = 44, Reflected Power = 194.3 nW
     retlw      0x25            ; Low Byte, ADRES = 48, Reflected Power = 213.0 nW
     retlw      0xD5            ; Low Byte, ADRES = 52, Reflected Power = 233.4 nW
     retlw      0x4B            ; Low Byte, ADRES = 56, Reflected Power = 255.9 nW
     retlw      0xB5            ; Low Byte, ADRES = 60, Reflected Power = 280.5 nW
     retlw      0x28            ; Low Byte, ADRES = 64, Reflected Power = 307.4 nW
     retlw      0xA7            ; Low Byte, ADRES = 68, Reflected Power = 337.0 nW
```

```
retlw      0x33            ; Low Byte, ADRES = 72, Reflected Power = 369.4 nW
retlw      0xCB            ; Low Byte, ADRES = 76, Reflected Power = 404.9 nW
retlw      0x72            ; Low Byte, ADRES = 80, Reflected Power = 443.8 nW
retlw      0x15            ; Low Byte, ADRES = 84, Reflected Power = 486.5 nW
retlw      0x79            ; Low Byte, ADRES = 88, Reflected Power = 533.2 nW
retlw      0xE7            ; Low Byte, ADRES = 92, Reflected Power = 584.5 nW
retlw      0x60            ; Low Byte, ADRES = 96, Reflected Power = 640.7 nW
retlw      0xE4            ; Low Byte, ADRES = 100, Reflected Power = 702.3 nW
retlw      0x75            ; Low Byte, ADRES = 104, Reflected Power = 769.8 nW
retlw      0x14            ; Low Byte, ADRES = 108, Reflected Power = 843.8 nW
retlw      0xC2            ; Low Byte, ADRES = 112, Reflected Power = 924.9 nW
retlw      0x41            ; Low Byte, ADRES = 116, Reflected Power = 1.014 uW
retlw      0xA9            ; Low Byte, ADRES = 120, Reflected Power = 1.111 uW
retlw      0x1C            ; Low Byte, ADRES = 124, Reflected Power = 1.218 uW
retlw      0x9A            ; Low Byte, ADRES = 128, Reflected Power = 1.335 uW
retlw      0x23            ; Low Byte, ADRES = 132, Reflected Power = 1.463 uW
retlw      0xBA            ; Low Byte, ADRES = 136, Reflected Power = 1.604 uW
retlw      0x60            ; Low Byte, ADRES = 140, Reflected Power = 1.758 uW
retlw      0x0B            ; Low Byte, ADRES = 144, Reflected Power = 1.927 uW
retlw      0x6E            ; Low Byte, ADRES = 148, Reflected Power = 2.113 uW
retlw      0xDB            ; Low Byte, ADRES = 152, Reflected Power = 2.316 uW
retlw      0x53            ; Low Byte, ADRES = 156, Reflected Power = 2.538 uW
retlw      0xD6            ; Low Byte, ADRES = 160, Reflected Power = 2.782 uW
retlw      0x65            ; Low Byte, ADRES = 164, Reflected Power = 3.050 uW
retlw      0x03            ; Low Byte, ADRES = 168, Reflected Power = 3.343 uW
retlw      0xAF            ; Low Byte, ADRES = 172, Reflected Power = 3.664 uW
retlw      0x36            ; Low Byte, ADRES = 176, Reflected Power = 4.016 uW
retlw      0x9E            ; Low Byte, ADRES = 180, Reflected Power = 4.403 uW
retlw      0x0F            ; Low Byte, ADRES = 184, Reflected Power = 4.826 uW
retlw      0x8C            ; Low Byte, ADRES = 188, Reflected Power = 5.290 uW
retlw      0x14            ; Low Byte, ADRES = 192, Reflected Power = 5.798 uW
retlw      0xAA            ; Low Byte, ADRES = 196, Reflected Power = 6.355 uW
retlw      0x4E            ; Low Byte, ADRES = 200, Reflected Power = 6.966 uW
retlw      0x01            ; Low Byte, ADRES = 204, Reflected Power = 7.636 uW
retlw      0x63            ; Low Byte, ADRES = 208, Reflected Power = 8.370 uW
retlw      0xCF            ; Low Byte, ADRES = 212, Reflected Power = 9.174 uW
retlw      0x46            ; Low Byte, ADRES = 216, Reflected Power = 10.06 uW
retlw      0xC7            ; Low Byte, ADRES = 220, Reflected Power = 11.02 uW
retlw      0x56            ; Low Byte, ADRES = 224, Reflected Power = 12.08 uW
retlw      0xF2            ; Low Byte, ADRES = 228, Reflected Power = 13.24 uW
retlw      0x9C            ; Low Byte, ADRES = 232, Reflected Power = 14.52 uW
retlw      0x2C            ; Low Byte, ADRES = 236, Reflected Power = 15.91 uW
retlw      0x93            ; Low Byte, ADRES = 240, Reflected Power = 17.44 uW
retlw      0x03            ; Low Byte, ADRES = 244, Reflected Power = 19.12 uW
retlw      0x7E            ; Low Byte, ADRES = 248, Reflected Power = 20.96 uW
retlw      0x06            ; Low Byte, ADRES = 252, Reflected Power = 22.97 uW
retlw      0x9A            ; Low Byte, ADRES = 256, Reflected Power = 25.18 uW
retlw      0x3C            ; Low Byte, ADRES = 260, Reflected Power = 27.60 uW
retlw      0xEE            ; Low Byte, ADRES = 264, Reflected Power = 30.25 uW
retlw      0x59            ; Low Byte, ADRES = 268, Reflected Power = 33.16 uW
retlw      0xC4            ; Low Byte, ADRES = 272, Reflected Power = 36.35 uW
retlw      0x39            ; Low Byte, ADRES = 276, Reflected Power = 39.84 uW
retlw      0xB9            ; Low Byte, ADRES = 280, Reflected Power = 43.67 uW
retlw      0x46            ; Low Byte, ADRES = 284, Reflected Power = 47.87 uW
retlw      0xE1            ; Low Byte, ADRES = 288, Reflected Power = 52.47 uW
retlw      0x8A            ; Low Byte, ADRES = 292, Reflected Power = 57.51 uW
retlw      0x22            ; Low Byte, ADRES = 296, Reflected Power = 63.04 uW
retlw      0x87            ; Low Byte, ADRES = 300, Reflected Power = 69.10 uW
retlw      0xF7            ; Low Byte, ADRES = 304, Reflected Power = 75.75 uW
retlw      0x71            ; Low Byte, ADRES = 308, Reflected Power = 83.03 uW
retlw      0xF7            ; Low Byte, ADRES = 312, Reflected Power = 91.01 uW
retlw      0x8A            ; Low Byte, ADRES = 316, Reflected Power = 99.76 uW
retlw      0x2B            ; Low Byte, ADRES = 320, Reflected Power = 109.3 uW
retlw      0xDB            ; Low Byte, ADRES = 324, Reflected Power = 119.9 uW
retlw      0x4E            ; Low Byte, ADRES = 328, Reflected Power = 131.4 uW
retlw      0xB8            ; Low Byte, ADRES = 332, Reflected Power = 144.0 uW
retlw      0x2C            ; Low Byte, ADRES = 336, Reflected Power = 157.8 uW
retlw      0xAB            ; Low Byte, ADRES = 340, Reflected Power = 173.0 uW
retlw      0x37            ; Low Byte, ADRES = 344, Reflected Power = 189.7 uW
retlw      0xD0            ; Low Byte, ADRES = 348, Reflected Power = 207.9 uW
retlw      0x77            ; Low Byte, ADRES = 352, Reflected Power = 227.9 uW
retlw      0x18            ; Low Byte, ADRES = 356, Reflected Power = 249.8 uW
retlw      0x7C            ; Low Byte, ADRES = 360, Reflected Power = 273.8 uW
retlw      0xEB            ; Low Byte, ADRES = 364, Reflected Power = 300.1 uW
retlw      0x64            ; Low Byte, ADRES = 368, Reflected Power = 328.9 uW
retlw      0xE8            ; Low Byte, ADRES = 372, Reflected Power = 360.6 uW
retlw      0x7A            ; Low Byte, ADRES = 376, Reflected Power = 395.2 uW
retlw      0x19            ; Low Byte, ADRES = 380, Reflected Power = 433.2 uW
retlw      0xC8            ; Low Byte, ADRES = 384, Reflected Power = 474.9 uW
retlw      0x44            ; Low Byte, ADRES = 388, Reflected Power = 520.5 uW
retlw      0xAD            ; Low Byte, ADRES = 392, Reflected Power = 570.5 uW
```

```
retlw      0x20            ; Low Byte, ADRES = 396, Reflected Power = 625.4 uW
retlw      0x9E            ; Low Byte, ADRES = 400, Reflected Power = 685.5 uW
retlw      0x28            ; Low Byte, ADRES = 404, Reflected Power = 751.4 uW
retlw      0xBF            ; Low Byte, ADRES = 408, Reflected Power = 823.6 uW
retlw      0x65            ; Low Byte, ADRES = 412, Reflected Power = 902.8 uW
retlw      0x0E            ; Low Byte, ADRES = 416, Reflected Power = 989.6 uW
retlw      0x71            ; Low Byte, ADRES = 420, Reflected Power = 1.085 mW
retlw      0xDF            ; Low Byte, ADRES = 424, Reflected Power = 1.189 mW
retlw      0x57            ; Low Byte, ADRES = 428, Reflected Power = 1.303 mW
retlw      0xDA            ; Low Byte, ADRES = 432, Reflected Power = 1.428 mW
retlw      0x6A            ; Low Byte, ADRES = 436, Reflected Power = 1.566 mW
retlw      0x08            ; Low Byte, ADRES = 440, Reflected Power = 1.716 mW
retlw      0xB5            ; Low Byte, ADRES = 444, Reflected Power = 1.881 mW
retlw      0x39            ; Low Byte, ADRES = 448, Reflected Power = 2.062 mW
retlw      0xA1            ; Low Byte, ADRES = 452, Reflected Power = 2.260 mW
retlw      0x13            ; Low Byte, ADRES = 456, Reflected Power = 2.478 mW
retlw      0x90            ; Low Byte, ADRES = 460, Reflected Power = 2.716 mW
retlw      0x19            ; Low Byte, ADRES = 464, Reflected Power = 2.977 mW
retlw      0xAF            ; Low Byte, ADRES = 468, Reflected Power = 3.263 mW
retlw      0x53            ; Low Byte, ADRES = 472, Reflected Power = 3.577 mW
retlw      0x04            ; Low Byte, ADRES = 476, Reflected Power = 3.920 mW
retlw      0x67            ; Low Byte, ADRES = 480, Reflected Power = 4.297 mW
retlw      0xD3            ; Low Byte, ADRES = 484, Reflected Power = 4.710 mW
retlw      0x4A            ; Low Byte, ADRES = 488, Reflected Power = 5.163 mW
retlw      0xCC            ; Low Byte, ADRES = 492, Reflected Power = 5.660 mW
retlw      0x5A            ; Low Byte, ADRES = 496, Reflected Power = 6.204 mW
retlw      0xF7            ; Low Byte, ADRES = 500, Reflected Power = 6.800 mW
retlw      0xA2            ; Low Byte, ADRES = 504, Reflected Power = 7.453 mW
retlw      0x2F            ; Low Byte, ADRES = 508, Reflected Power = 8.170 mW
retlw      0x96            ; Low Byte, ADRES = 512, Reflected Power = 8.955 mW
retlw      0x07            ; Low Byte, ADRES = 516, Reflected Power = 9.816 mW
retlw      0x82            ; Low Byte, ADRES = 520, Reflected Power = 10.76 mW
retlw      0x0A            ; Low Byte, ADRES = 524, Reflected Power = 11.79 mW
retlw      0x9E            ; Low Byte, ADRES = 528, Reflected Power = 12.93 mW
retlw      0x41            ; Low Byte, ADRES = 532, Reflected Power = 14.17 mW
retlw      0xF4            ; Low Byte, ADRES = 536, Reflected Power = 15.53 mW
retlw      0x5C            ; Low Byte, ADRES = 540, Reflected Power = 17.03 mW
retlw      0xC7            ; Low Byte, ADRES = 544, Reflected Power = 18.66 mW
retlw      0x3D            ; Low Byte, ADRES = 548, Reflected Power = 20.46 mW
retlw      0xBD            ; Low Byte, ADRES = 552, Reflected Power = 22.42 mW
retlw      0x4B            ; Low Byte, ADRES = 556, Reflected Power = 24.58 mW
retlw      0xE6            ; Low Byte, ADRES = 560, Reflected Power = 26.94 mW
retlw      0x8F            ; Low Byte, ADRES = 564, Reflected Power = 29.53 mW
retlw      0x25            ; Low Byte, ADRES = 568, Reflected Power = 32.37 mW
retlw      0x8B            ; Low Byte, ADRES = 572, Reflected Power = 35.48 mW
retlw      0xFA            ; Low Byte, ADRES = 576, Reflected Power = 38.89 mW
retlw      0x75            ; Low Byte, ADRES = 580, Reflected Power = 42.63 mW
retlw      0xFB            ; Low Byte, ADRES = 584, Reflected Power = 46.73 mW
retlw      0x8E            ; Low Byte, ADRES = 588, Reflected Power = 51.22 mW
retlw      0x30            ; Low Byte, ADRES = 592, Reflected Power = 56.14 mW
retlw      0xE0            ; Low Byte, ADRES = 596, Reflected Power = 61.54 mW
retlw      0x51            ; Low Byte, ADRES = 600, Reflected Power = 67.45 mW
retlw      0xBB            ; Low Byte, ADRES = 604, Reflected Power = 73.94 mW
retlw      0x30            ; Low Byte, ADRES = 608, Reflected Power = 81.04 mW
retlw      0xAF            ; Low Byte, ADRES = 612, Reflected Power = 88.83 mW
retlw      0x3B            ; Low Byte, ADRES = 616, Reflected Power = 97.37 mW
retlw      0xD5            ; Low Byte, ADRES = 620, Reflected Power = 106.7 mW
retlw      0x7D            ; Low Byte, ADRES = 624, Reflected Power = 117.0 mW
retlw      0x1B            ; Low Byte, ADRES = 628, Reflected Power = 128.2 mW
retlw      0x80            ; Low Byte, ADRES = 632, Reflected Power = 140.6 mW
retlw      0xEE            ; Low Byte, ADRES = 636, Reflected Power = 154.1 mW
retlw      0x68            ; Low Byte, ADRES = 640, Reflected Power = 168.9 mW
retlw      0xED            ; Low Byte, ADRES = 644, Reflected Power = 185.1 mW
retlw      0x7E            ; Low Byte, ADRES = 648, Reflected Power = 202.9 mW
retlw      0x1E            ; Low Byte, ADRES = 652, Reflected Power = 222.4 mW
retlw      0xCD            ; Low Byte, ADRES = 656, Reflected Power = 243.8 mW
retlw      0x47            ; Low Byte, ADRES = 660, Reflected Power = 267.2 mW
retlw      0xB0            ; Low Byte, ADRES = 664, Reflected Power = 292.9 mW
retlw      0x23            ; Low Byte, ADRES = 668, Reflected Power = 321.1 mW
retlw      0xA2            ; Low Byte, ADRES = 672, Reflected Power = 351.9 mW
retlw      0x2C            ; Low Byte, ADRES = 676, Reflected Power = 385.8 mW
retlw      0xC4            ; Low Byte, ADRES = 680, Reflected Power = 422.9 mW
retlw      0x6B            ; Low Byte, ADRES = 684, Reflected Power = 463.5 mW
retlw      0x11            ; Low Byte, ADRES = 688, Reflected Power = 508.1 mW
retlw      0x75            ; Low Byte, ADRES = 692, Reflected Power = 556.9 mW
retlw      0xE2            ; Low Byte, ADRES = 696, Reflected Power = 610.4 mW
retlw      0x5A            ; Low Byte, ADRES = 700, Reflected Power = 669.1 mW
retlw      0xDE            ; Low Byte, ADRES = 704, Reflected Power = 733.4 mW
retlw      0x6E            ; Low Byte, ADRES = 708, Reflected Power = 803.9 mW
retlw      0x0D            ; Low Byte, ADRES = 712, Reflected Power = 881.2 mW
retlw      0xBA            ; Low Byte, ADRES = 716, Reflected Power = 965.9 mW
```

```
        retlw   0x3C            ; Low Byte, ADRES = 720, Reflected Power = 1.059 W
        retlw   0xA4            ; Low Byte, ADRES = 724, Reflected Power = 1.161 W
        retlw   0x17            ; Low Byte, ADRES = 728, Reflected Power = 1.272 W
        retlw   0x94            ; Low Byte, ADRES = 732, Reflected Power = 1.394 W
        retlw   0x1D            ; Low Byte, ADRES = 736, Reflected Power = 1.528 W
        retlw   0xB4            ; Low Byte, ADRES = 740, Reflected Power = 1.675 W
        retlw   0x58            ; Low Byte, ADRES = 744, Reflected Power = 1.836 W
        retlw   0x07            ; Low Byte, ADRES = 748, Reflected Power = 2.013 W
        retlw   0x6A            ; Low Byte, ADRES = 752, Reflected Power = 2.206 W
        retlw   0xD6            ; Low Byte, ADRES = 756, Reflected Power = 2.418 W
        retlw   0x4D            ; Low Byte, ADRES = 760, Reflected Power = 2.651 W
        retlw   0xD0            ; Low Byte, ADRES = 764, Reflected Power = 2.906 W
        retlw   0x5F            ; Low Byte, ADRES = 768, Reflected Power = 3.185 W
        retlw   0xFC            ; Low Byte, ADRES = 772, Reflected Power = 3.491 W
        retlw   0xA7            ; Low Byte, ADRES = 776, Reflected Power = 3.827 W
        retlw   0x32            ; Low Byte, ADRES = 780, Reflected Power = 4.195 W
        retlw   0x99            ; Low Byte, ADRES = 784, Reflected Power = 4.598 W
        retlw   0x0A            ; Low Byte, ADRES = 788, Reflected Power = 5.040 W
        retlw   0x86            ; Low Byte, ADRES = 792, Reflected Power = 5.524 W
        retlw   0x0E            ; Low Byte, ADRES = 796, Reflected Power = 6.055 W
        retlw   0xA3            ; Low Byte, ADRES = 800, Reflected Power = 6.637 W
        retlw   0x47            ; Low Byte, ADRES = 804, Reflected Power = 7.275 W
        retlw   0xFA            ; Low Byte, ADRES = 808, Reflected Power = 7.975 W
        retlw   0x5F            ; Low Byte, ADRES = 812, Reflected Power = 8.741 W
        retlw   0xCA            ; Low Byte, ADRES = 816, Reflected Power = 9.582 W
        retlw   0x40            ; Low Byte, ADRES = 820, Reflected Power = 10.50 W
        retlw   0xC2            ; Low Byte, ADRES = 824, Reflected Power = 11.51 W
        retlw   0x4F            ; Low Byte, ADRES = 828, Reflected Power = 12.62 W
        retlw   0xEA            ; Low Byte, ADRES = 832, Reflected Power = 13.83 W
        retlw   0x95            ; Low Byte, ADRES = 836, Reflected Power = 15.16 W
        retlw   0x28            ; Low Byte, ADRES = 840, Reflected Power = 16.62 W
        retlw   0x8E            ; Low Byte, ADRES = 844, Reflected Power = 18.22 W
        retlw   0xFE            ; Low Byte, ADRES = 848, Reflected Power = 19.97 W
        retlw   0x79            ; Low Byte, ADRES = 852, Reflected Power = 21.89 W
        retlw   0xFF            ; Low Byte, ADRES = 856, Reflected Power = 23.99 W
        retlw   0x93            ; Low Byte, ADRES = 860, Reflected Power = 26.30 W
        retlw   0x35            ; Low Byte, ADRES = 864, Reflected Power = 28.82 W
        retlw   0xE6            ; Low Byte, ADRES = 868, Reflected Power = 31.60 W
        retlw   0x54            ; Low Byte, ADRES = 872, Reflected Power = 34.63 W
        retlw   0xBF            ; Low Byte, ADRES = 876, Reflected Power = 37.96 W
        retlw   0x34            ; Low Byte, ADRES = 880, Reflected Power = 41.61 W
        retlw   0xB4            ; Low Byte, ADRES = 884, Reflected Power = 45.61 W
        retlw   0x40            ; Low Byte, ADRES = 888, Reflected Power = 49.99 W
        retlw   0xDA            ; Low Byte, ADRES = 892, Reflected Power = 54.80 W
        retlw   0x82            ; Low Byte, ADRES = 896, Reflected Power = 60.07 W
        retlw   0x1D            ; Low Byte, ADRES = 900, Reflected Power = 65.84 W
        retlw   0x83            ; Low Byte, ADRES = 904, Reflected Power = 72.17 W
        retlw   0xF2            ; Low Byte, ADRES = 908, Reflected Power = 79.11 W
        retlw   0x6B            ; Low Byte, ADRES = 912, Reflected Power = 86.71 W
        retlw   0xF1            ; Low Byte, ADRES = 916, Reflected Power = 95.05 W
        retlw   0x83            ; Low Byte, ADRES = 920, Reflected Power = 104.2 W
        retlw   0x23            ; Low Byte, ADRES = 924, Reflected Power = 114.2 W
        retlw   0xD3            ; Low Byte, ADRES = 928, Reflected Power = 125.2 W
        retlw   0x4A            ; Low Byte, ADRES = 932, Reflected Power = 137.2 W
        retlw   0xB3            ; Low Byte, ADRES = 936, Reflected Power = 150.4 W
        retlw   0x27            ; Low Byte, ADRES = 940, Reflected Power = 164.9 W
        retlw   0xA6            ; Low Byte, ADRES = 944, Reflected Power = 180.7 W
        retlw   0x31            ; Low Byte, ADRES = 948, Reflected Power = 198.1 W
        retlw   0xC9            ; Low Byte, ADRES = 952, Reflected Power = 217.1 W
        retlw   0x70            ; Low Byte, ADRES = 956, Reflected Power = 238.0 W
        retlw   0x13            ; Low Byte, ADRES = 960, Reflected Power = 260.9 W
        retlw   0x78            ; Low Byte, ADRES = 964, Reflected Power = 285.9 W
        retlw   0xE6            ; Low Byte, ADRES = 968, Reflected Power = 313.4 W
        retlw   0x5E            ; Low Byte, ADRES = 972, Reflected Power = 343.5 W
        retlw   0xE2            ; Low Byte, ADRES = 976, Reflected Power = 376.6 W
        retlw   0x73            ; Low Byte, ADRES = 980, Reflected Power = 412.8 W
        retlw   0x12            ; Low Byte, ADRES = 984, Reflected Power = 452.4 W
        retlw   0xC0            ; Low Byte, ADRES = 988, Reflected Power = 495.9 W
        retlw   0x3F            ; Low Byte, ADRES = 992, Reflected Power = 543.6 W
        retlw   0xA8            ; Low Byte, ADRES = 996, Reflected Power = 595.9 W
        retlw   0x1A            ; Low Byte, ADRES = 1000, Reflected Power = 653.1 W
        retlw   0x98            ; Low Byte, ADRES = 1004, Reflected Power = 715.9 W
        retlw   0x21            ; Low Byte, ADRES = 1008, Reflected Power = 784.7 W
        retlw   0xB8            ; Low Byte, ADRES = 1012, Reflected Power = 860.2 W
        retlw   0x5E            ; Low Byte, ADRES = 1016, Reflected Power = 942.8 W
        retlw   0x09            ; Low Byte, ADRES = 1020, Reflected Power = 1,033.5 W

RLowByteTable1
        retlw   0xDC            ; Low Byte, ADRES = 1, Reflected Power = 72.44 nW
        retlw   0x54            ; Low Byte, ADRES = 5, Reflected Power = 79.40 nW
        retlw   0xD7            ; Low Byte, ADRES = 9, Reflected Power = 87.03 nW
```

```
retlw      0x67            ; Low Byte, ADRES = 13, Reflected Power = 95.40 nW
retlw      0x05            ; Low Byte, ADRES = 17, Reflected Power = 104.6 nW
retlw      0xB1            ; Low Byte, ADRES = 21, Reflected Power = 114.6 nW
retlw      0x37            ; Low Byte, ADRES = 25, Reflected Power = 125.6 nW
retlw      0x9F            ; Low Byte, ADRES = 29, Reflected Power = 137.7 nW
retlw      0x11            ; Low Byte, ADRES = 33, Reflected Power = 151.0 nW
retlw      0x8D            ; Low Byte, ADRES = 37, Reflected Power = 165.5 nW
retlw      0x16            ; Low Byte, ADRES = 41, Reflected Power = 181.4 nW
retlw      0xAC            ; Low Byte, ADRES = 45, Reflected Power = 198.8 nW
retlw      0x50            ; Low Byte, ADRES = 49, Reflected Power = 217.9 nW
retlw      0x02            ; Low Byte, ADRES = 53, Reflected Power = 238.9 nW
retlw      0x65            ; Low Byte, ADRES = 57, Reflected Power = 261.8 nW
retlw      0xD1            ; Low Byte, ADRES = 61, Reflected Power = 287.0 nW
retlw      0x47            ; Low Byte, ADRES = 65, Reflected Power = 314.6 nW
retlw      0xC9            ; Low Byte, ADRES = 69, Reflected Power = 344.8 nW
retlw      0x57            ; Low Byte, ADRES = 73, Reflected Power = 378.0 nW
retlw      0xF3            ; Low Byte, ADRES = 77, Reflected Power = 414.3 nW
retlw      0x9E            ; Low Byte, ADRES = 81, Reflected Power = 454.1 nW
retlw      0x2D            ; Low Byte, ADRES = 85, Reflected Power = 497.8 nW
retlw      0x94            ; Low Byte, ADRES = 89, Reflected Power = 545.6 nW
retlw      0x04            ; Low Byte, ADRES = 93, Reflected Power = 598.1 nW
retlw      0x80            ; Low Byte, ADRES = 97, Reflected Power = 655.5 nW
retlw      0x07            ; Low Byte, ADRES = 101, Reflected Power = 718.6 nW
retlw      0x9B            ; Low Byte, ADRES = 105, Reflected Power = 787.6 nW
retlw      0x3E            ; Low Byte, ADRES = 109, Reflected Power = 863.3 nW
retlw      0xF0            ; Low Byte, ADRES = 113, Reflected Power = 946.3 nW
retlw      0x5A            ; Low Byte, ADRES = 117, Reflected Power = 1.037 uW
retlw      0xC5            ; Low Byte, ADRES = 121, Reflected Power = 1.137 uW
retlw      0x3A            ; Low Byte, ADRES = 125, Reflected Power = 1.246 uW
retlw      0xBB            ; Low Byte, ADRES = 129, Reflected Power = 1.366 uW
retlw      0x48            ; Low Byte, ADRES = 133, Reflected Power = 1.497 uW
retlw      0xE2            ; Low Byte, ADRES = 137, Reflected Power = 1.641 uW
retlw      0x8C            ; Low Byte, ADRES = 141, Reflected Power = 1.799 uW
retlw      0x23            ; Low Byte, ADRES = 145, Reflected Power = 1.972 uW
retlw      0x89            ; Low Byte, ADRES = 149, Reflected Power = 2.162 uW
retlw      0xF8            ; Low Byte, ADRES = 153, Reflected Power = 2.369 uW
retlw      0x72            ; Low Byte, ADRES = 157, Reflected Power = 2.597 uW
retlw      0xF8            ; Low Byte, ADRES = 161, Reflected Power = 2.847 uW
retlw      0x8B            ; Low Byte, ADRES = 165, Reflected Power = 3.121 uW
retlw      0x2C            ; Low Byte, ADRES = 169, Reflected Power = 3.420 uW
retlw      0xDD            ; Low Byte, ADRES = 173, Reflected Power = 3.749 uW
retlw      0x4F            ; Low Byte, ADRES = 177, Reflected Power = 4.110 uW
retlw      0xB9            ; Low Byte, ADRES = 181, Reflected Power = 4.505 uW
retlw      0x2D            ; Low Byte, ADRES = 185, Reflected Power = 4.938 uW
retlw      0xAD            ; Low Byte, ADRES = 189, Reflected Power = 5.412 uW
retlw      0x39            ; Low Byte, ADRES = 193, Reflected Power = 5.933 uW
retlw      0xD2            ; Low Byte, ADRES = 197, Reflected Power = 6.503 uW
retlw      0x79            ; Low Byte, ADRES = 201, Reflected Power = 7.128 uW
retlw      0x19            ; Low Byte, ADRES = 205, Reflected Power = 7.813 uW
retlw      0x7D            ; Low Byte, ADRES = 209, Reflected Power = 8.564 uW
retlw      0xEC            ; Low Byte, ADRES = 213, Reflected Power = 9.387 uW
retlw      0x65            ; Low Byte, ADRES = 217, Reflected Power = 10.29 uW
retlw      0xEA            ; Low Byte, ADRES = 221, Reflected Power = 11.28 uW
retlw      0x7B            ; Low Byte, ADRES = 225, Reflected Power = 12.36 uW
retlw      0x1B            ; Low Byte, ADRES = 229, Reflected Power = 13.55 uW
retlw      0xCA            ; Low Byte, ADRES = 233, Reflected Power = 14.85 uW
retlw      0x45            ; Low Byte, ADRES = 237, Reflected Power = 16.28 uW
retlw      0xAE            ; Low Byte, ADRES = 241, Reflected Power = 17.85 uW
retlw      0x21            ; Low Byte, ADRES = 245, Reflected Power = 19.56 uW
retlw      0x9F            ; Low Byte, ADRES = 249, Reflected Power = 21.44 uW
retlw      0x29            ; Low Byte, ADRES = 253, Reflected Power = 23.50 uW
retlw      0xC1            ; Low Byte, ADRES = 257, Reflected Power = 25.76 uW
retlw      0x67            ; Low Byte, ADRES = 261, Reflected Power = 28.24 uW
retlw      0x0F            ; Low Byte, ADRES = 265, Reflected Power = 30.95 uW
retlw      0x73            ; Low Byte, ADRES = 269, Reflected Power = 33.93 uW
retlw      0xE0            ; Low Byte, ADRES = 273, Reflected Power = 37.19 uW
retlw      0x58            ; Low Byte, ADRES = 277, Reflected Power = 40.77 uW
retlw      0xDB            ; Low Byte, ADRES = 281, Reflected Power = 44.69 uW
retlw      0x6C            ; Low Byte, ADRES = 285, Reflected Power = 48.98 uW
retlw      0x0A            ; Low Byte, ADRES = 289, Reflected Power = 53.69 uW
retlw      0xB7            ; Low Byte, ADRES = 293, Reflected Power = 58.85 uW
retlw      0x3A            ; Low Byte, ADRES = 297, Reflected Power = 64.51 uW
retlw      0xA2            ; Low Byte, ADRES = 301, Reflected Power = 70.71 uW
retlw      0x14            ; Low Byte, ADRES = 305, Reflected Power = 77.50 uW
retlw      0x91            ; Low Byte, ADRES = 309, Reflected Power = 84.95 uW
retlw      0x1A            ; Low Byte, ADRES = 313, Reflected Power = 93.12 uW
retlw      0xB0            ; Low Byte, ADRES = 317, Reflected Power = 102.1 uW
retlw      0x55            ; Low Byte, ADRES = 321, Reflected Power = 111.9 uW
retlw      0x05            ; Low Byte, ADRES = 325, Reflected Power = 122.6 uW
retlw      0x68            ; Low Byte, ADRES = 329, Reflected Power = 134.4 uW
retlw      0xD4            ; Low Byte, ADRES = 333, Reflected Power = 147.3 uW
```

```
retlw      0x4B            ; Low Byte, ADRES = 337, Reflected Power = 161.5 uW
retlw      0xCD            ; Low Byte, ADRES = 341, Reflected Power = 177.0 uW
retlw      0x5C            ; Low Byte, ADRES = 345, Reflected Power = 194.1 uW
retlw      0xF8            ; Low Byte, ADRES = 349, Reflected Power = 212.7 uW
retlw      0xA4            ; Low Byte, ADRES = 353, Reflected Power = 233.2 uW
retlw      0x30            ; Low Byte, ADRES = 357, Reflected Power = 255.6 uW
retlw      0x97            ; Low Byte, ADRES = 361, Reflected Power = 280.1 uW
retlw      0x08            ; Low Byte, ADRES = 365, Reflected Power = 307.1 uW
retlw      0x84            ; Low Byte, ADRES = 369, Reflected Power = 336.6 uW
retlw      0x0B            ; Low Byte, ADRES = 373, Reflected Power = 368.9 uW
retlw      0xA0            ; Low Byte, ADRES = 377, Reflected Power = 404.4 uW
retlw      0x43            ; Low Byte, ADRES = 381, Reflected Power = 443.3 uW
retlw      0xF6            ; Low Byte, ADRES = 385, Reflected Power = 485.9 uW
retlw      0x5D            ; Low Byte, ADRES = 389, Reflected Power = 532.6 uW
retlw      0xC8            ; Low Byte, ADRES = 393, Reflected Power = 583.8 uW
retlw      0x3E            ; Low Byte, ADRES = 397, Reflected Power = 639.9 uW
retlw      0xBF            ; Low Byte, ADRES = 401, Reflected Power = 701.4 uW
retlw      0x4C            ; Low Byte, ADRES = 405, Reflected Power = 768.8 uW
retlw      0xE7            ; Low Byte, ADRES = 409, Reflected Power = 842.7 uW
retlw      0x91            ; Low Byte, ADRES = 413, Reflected Power = 923.7 uW
retlw      0x26            ; Low Byte, ADRES = 417, Reflected Power = 1.013 mW
retlw      0x8C            ; Low Byte, ADRES = 421, Reflected Power = 1.110 mW
retlw      0xFC            ; Low Byte, ADRES = 425, Reflected Power = 1.217 mW
retlw      0x76            ; Low Byte, ADRES = 429, Reflected Power = 1.333 mW
retlw      0xFD            ; Low Byte, ADRES = 433, Reflected Power = 1.462 mW
retlw      0x90            ; Low Byte, ADRES = 437, Reflected Power = 1.602 mW
retlw      0x31            ; Low Byte, ADRES = 441, Reflected Power = 1.756 mW
retlw      0xE2            ; Low Byte, ADRES = 445, Reflected Power = 1.925 mW
retlw      0x52            ; Low Byte, ADRES = 449, Reflected Power = 2.110 mW
retlw      0xBD            ; Low Byte, ADRES = 453, Reflected Power = 2.313 mW
retlw      0x31            ; Low Byte, ADRES = 457, Reflected Power = 2.535 mW
retlw      0xB1            ; Low Byte, ADRES = 461, Reflected Power = 2.779 mW
retlw      0x3D            ; Low Byte, ADRES = 465, Reflected Power = 3.046 mW
retlw      0xD6            ; Low Byte, ADRES = 469, Reflected Power = 3.339 mW
retlw      0x7F            ; Low Byte, ADRES = 473, Reflected Power = 3.660 mW
retlw      0x1C            ; Low Byte, ADRES = 477, Reflected Power = 4.011 mW
retlw      0x81            ; Low Byte, ADRES = 481, Reflected Power = 4.397 mW
retlw      0xEF            ; Low Byte, ADRES = 485, Reflected Power = 4.820 mW
retlw      0x69            ; Low Byte, ADRES = 489, Reflected Power = 5.283 mW
retlw      0xEE            ; Low Byte, ADRES = 493, Reflected Power = 5.791 mW
retlw      0x80            ; Low Byte, ADRES = 497, Reflected Power = 6.348 mW
retlw      0x20            ; Low Byte, ADRES = 501, Reflected Power = 6.958 mW
retlw      0xCF            ; Low Byte, ADRES = 505, Reflected Power = 7.626 mW
retlw      0x48            ; Low Byte, ADRES = 509, Reflected Power = 8.360 mW
retlw      0xB1            ; Low Byte, ADRES = 513, Reflected Power = 9.163 mW
retlw      0x24            ; Low Byte, ADRES = 517, Reflected Power = 10.04 mW
retlw      0xA3            ; Low Byte, ADRES = 521, Reflected Power = 11.01 mW
retlw      0x2E            ; Low Byte, ADRES = 525, Reflected Power = 12.07 mW
retlw      0xC6            ; Low Byte, ADRES = 529, Reflected Power = 13.23 mW
retlw      0x6C            ; Low Byte, ADRES = 533, Reflected Power = 14.50 mW
retlw      0x12            ; Low Byte, ADRES = 537, Reflected Power = 15.89 mW
retlw      0x76            ; Low Byte, ADRES = 541, Reflected Power = 17.42 mW
retlw      0xE3            ; Low Byte, ADRES = 545, Reflected Power = 19.10 mW
retlw      0x5C            ; Low Byte, ADRES = 549, Reflected Power = 20.93 mW
retlw      0xE0            ; Low Byte, ADRES = 553, Reflected Power = 22.94 mW
retlw      0x70            ; Low Byte, ADRES = 557, Reflected Power = 25.15 mW
retlw      0x0F            ; Low Byte, ADRES = 561, Reflected Power = 27.57 mW
retlw      0xBC            ; Low Byte, ADRES = 565, Reflected Power = 30.22 mW
retlw      0x3D            ; Low Byte, ADRES = 569, Reflected Power = 33.12 mW
retlw      0xA6            ; Low Byte, ADRES = 573, Reflected Power = 36.30 mW
retlw      0x18            ; Low Byte, ADRES = 577, Reflected Power = 39.79 mW
retlw      0x95            ; Low Byte, ADRES = 581, Reflected Power = 43.62 mW
retlw      0x1F            ; Low Byte, ADRES = 585, Reflected Power = 47.81 mW
retlw      0xB5            ; Low Byte, ADRES = 589, Reflected Power = 52.41 mW
retlw      0x5A            ; Low Byte, ADRES = 593, Reflected Power = 57.44 mW
retlw      0x08            ; Low Byte, ADRES = 597, Reflected Power = 62.97 mW
retlw      0x6B            ; Low Byte, ADRES = 601, Reflected Power = 69.02 mW
retlw      0xD8            ; Low Byte, ADRES = 605, Reflected Power = 75.65 mW
retlw      0x4F            ; Low Byte, ADRES = 609, Reflected Power = 82.92 mW
retlw      0xD1            ; Low Byte, ADRES = 613, Reflected Power = 90.90 mW
retlw      0x60            ; Low Byte, ADRES = 617, Reflected Power = 99.63 mW
retlw      0xFD            ; Low Byte, ADRES = 621, Reflected Power = 109.2 mW
retlw      0xA9            ; Low Byte, ADRES = 625, Reflected Power = 119.7 mW
retlw      0x33            ; Low Byte, ADRES = 629, Reflected Power = 131.2 mW
retlw      0x9A            ; Low Byte, ADRES = 633, Reflected Power = 143.8 mW
retlw      0x0C            ; Low Byte, ADRES = 637, Reflected Power = 157.7 mW
retlw      0x88            ; Low Byte, ADRES = 641, Reflected Power = 172.8 mW
retlw      0x10            ; Low Byte, ADRES = 645, Reflected Power = 189.4 mW
retlw      0xA5            ; Low Byte, ADRES = 649, Reflected Power = 207.6 mW
retlw      0x48            ; Low Byte, ADRES = 653, Reflected Power = 227.6 mW
retlw      0xFC            ; Low Byte, ADRES = 657, Reflected Power = 249.5 mW
```

```
retlw      0x60            ; Low Byte, ADRES = 661, Reflected Power = 273.4 mW
retlw      0xCC            ; Low Byte, ADRES = 665, Reflected Power = 299.7 mW
retlw      0x42            ; Low Byte, ADRES = 669, Reflected Power = 328.5 mW
retlw      0xC3            ; Low Byte, ADRES = 673, Reflected Power = 360.1 mW
retlw      0x51            ; Low Byte, ADRES = 677, Reflected Power = 394.7 mW
retlw      0xEC            ; Low Byte, ADRES = 681, Reflected Power = 432.7 mW
retlw      0x97            ; Low Byte, ADRES = 685, Reflected Power = 474.3 mW
retlw      0x29            ; Low Byte, ADRES = 689, Reflected Power = 519.9 mW
retlw      0x8F            ; Low Byte, ADRES = 693, Reflected Power = 569.8 mW
retlw      0xFF            ; Low Byte, ADRES = 697, Reflected Power = 624.6 mW
retlw      0x7A            ; Low Byte, ADRES = 701, Reflected Power = 684.6 mW
retlw      0x01            ; Low Byte, ADRES = 705, Reflected Power = 750.5 mW
retlw      0x95            ; Low Byte, ADRES = 709, Reflected Power = 822.6 mW
retlw      0x37            ; Low Byte, ADRES = 713, Reflected Power = 901.7 mW
retlw      0xE8            ; Low Byte, ADRES = 717, Reflected Power = 988.3 mW
retlw      0x55            ; Low Byte, ADRES = 721, Reflected Power = 1.083 W
retlw      0xC0            ; Low Byte, ADRES = 725, Reflected Power = 1.187 W
retlw      0x35            ; Low Byte, ADRES = 729, Reflected Power = 1.302 W
retlw      0xB5            ; Low Byte, ADRES = 733, Reflected Power = 1.427 W
retlw      0x41            ; Low Byte, ADRES = 737, Reflected Power = 1.564 W
retlw      0xDB            ; Low Byte, ADRES = 741, Reflected Power = 1.714 W
retlw      0x84            ; Low Byte, ADRES = 745, Reflected Power = 1.879 W
retlw      0x1F            ; Low Byte, ADRES = 749, Reflected Power = 2.060 W
retlw      0x84            ; Low Byte, ADRES = 753, Reflected Power = 2.258 W
retlw      0xF3            ; Low Byte, ADRES = 757, Reflected Power = 2.475 W
retlw      0x6D            ; Low Byte, ADRES = 761, Reflected Power = 2.712 W
retlw      0xF2            ; Low Byte, ADRES = 765, Reflected Power = 2.973 W
retlw      0x85            ; Low Byte, ADRES = 769, Reflected Power = 3.259 W
retlw      0x25            ; Low Byte, ADRES = 773, Reflected Power = 3.572 W
retlw      0xD5            ; Low Byte, ADRES = 777, Reflected Power = 3.916 W
retlw      0x4B            ; Low Byte, ADRES = 781, Reflected Power = 4.292 W
retlw      0xB4            ; Low Byte, ADRES = 785, Reflected Power = 4.705 W
retlw      0x28            ; Low Byte, ADRES = 789, Reflected Power = 5.157 W
retlw      0xA7            ; Low Byte, ADRES = 793, Reflected Power = 5.653 W
retlw      0x32            ; Low Byte, ADRES = 797, Reflected Power = 6.196 W
retlw      0xCB            ; Low Byte, ADRES = 801, Reflected Power = 6.791 W
retlw      0x72            ; Low Byte, ADRES = 805, Reflected Power = 7.444 W
retlw      0x14            ; Low Byte, ADRES = 809, Reflected Power = 8.160 W
retlw      0x79            ; Low Byte, ADRES = 813, Reflected Power = 8.944 W
retlw      0xE7            ; Low Byte, ADRES = 817, Reflected Power = 9.804 W
retlw      0x60            ; Low Byte, ADRES = 821, Reflected Power = 10.75 W
retlw      0xE4            ; Low Byte, ADRES = 825, Reflected Power = 11.78 W
retlw      0x75            ; Low Byte, ADRES = 829, Reflected Power = 12.91 W
retlw      0x14            ; Low Byte, ADRES = 833, Reflected Power = 14.15 W
retlw      0xC2            ; Low Byte, ADRES = 837, Reflected Power = 15.51 W
retlw      0x40            ; Low Byte, ADRES = 841, Reflected Power = 17.00 W
retlw      0xA9            ; Low Byte, ADRES = 845, Reflected Power = 18.64 W
retlw      0x1C            ; Low Byte, ADRES = 849, Reflected Power = 20.43 W
retlw      0x99            ; Low Byte, ADRES = 853, Reflected Power = 22.39 W
retlw      0x23            ; Low Byte, ADRES = 857, Reflected Power = 24.55 W
retlw      0xBA            ; Low Byte, ADRES = 861, Reflected Power = 26.91 W
retlw      0x60            ; Low Byte, ADRES = 865, Reflected Power = 29.49 W
retlw      0x0B            ; Low Byte, ADRES = 869, Reflected Power = 32.33 W
retlw      0x6E            ; Low Byte, ADRES = 873, Reflected Power = 35.44 W
retlw      0xDB            ; Low Byte, ADRES = 877, Reflected Power = 38.84 W
retlw      0x52            ; Low Byte, ADRES = 881, Reflected Power = 42.58 W
retlw      0xD5            ; Low Byte, ADRES = 885, Reflected Power = 46.67 W
retlw      0x65            ; Low Byte, ADRES = 889, Reflected Power = 51.15 W
retlw      0x02            ; Low Byte, ADRES = 893, Reflected Power = 56.07 W
retlw      0xAF            ; Low Byte, ADRES = 897, Reflected Power = 61.46 W
retlw      0x36            ; Low Byte, ADRES = 901, Reflected Power = 67.37 W
retlw      0x9E            ; Low Byte, ADRES = 905, Reflected Power = 73.85 W
retlw      0x0F            ; Low Byte, ADRES = 909, Reflected Power = 80.94 W
retlw      0x8C            ; Low Byte, ADRES = 913, Reflected Power = 88.72 W
retlw      0x14            ; Low Byte, ADRES = 917, Reflected Power = 97.25 W
retlw      0xAA            ; Low Byte, ADRES = 921, Reflected Power = 106.6 W
retlw      0x4E            ; Low Byte, ADRES = 925, Reflected Power = 116.8 W
retlw      0x01            ; Low Byte, ADRES = 929, Reflected Power = 128.1 W
retlw      0x63            ; Low Byte, ADRES = 933, Reflected Power = 140.4 W
retlw      0xCF            ; Low Byte, ADRES = 937, Reflected Power = 153.9 W
retlw      0x45            ; Low Byte, ADRES = 941, Reflected Power = 168.7 W
retlw      0xC7            ; Low Byte, ADRES = 945, Reflected Power = 184.9 W
retlw      0x55            ; Low Byte, ADRES = 949, Reflected Power = 202.7 W
retlw      0xF1            ; Low Byte, ADRES = 953, Reflected Power = 222.1 W
retlw      0x9C            ; Low Byte, ADRES = 957, Reflected Power = 243.5 W
retlw      0x2C            ; Low Byte, ADRES = 961, Reflected Power = 266.9 W
retlw      0x92            ; Low Byte, ADRES = 965, Reflected Power = 292.6 W
retlw      0x03            ; Low Byte, ADRES = 969, Reflected Power = 320.7 W
retlw      0x7E            ; Low Byte, ADRES = 973, Reflected Power = 351.5 W
retlw      0x05            ; Low Byte, ADRES = 977, Reflected Power = 385.3 W
retlw      0x99            ; Low Byte, ADRES = 981, Reflected Power = 422.3 W
```

```
        retlw   0x3C            ; Low Byte, ADRES = 985, Reflected Power = 462.9 W
        retlw   0xEE            ; Low Byte, ADRES = 989, Reflected Power = 507.4 W
        retlw   0x58            ; Low Byte, ADRES = 993, Reflected Power = 556.2 W
        retlw   0xC3            ; Low Byte, ADRES = 997, Reflected Power = 609.7 W
        retlw   0x39            ; Low Byte, ADRES = 1001, Reflected Power = 668.3 W
        retlw   0xB9            ; Low Byte, ADRES = 1005, Reflected Power = 732.5 W
        retlw   0x46            ; Low Byte, ADRES = 1009, Reflected Power = 802.9 W
        retlw   0xE0            ; Low Byte, ADRES = 1013, Reflected Power = 880.1 W
        retlw   0x89            ; Low Byte, ADRES = 1017, Reflected Power = 964.7 W
        retlw   0x21            ; Low Byte, ADRES = 1021, Reflected Power = 1,057.5 W

RLowByteTable2
        retlw   0xF9            ; Low Byte, ADRES = 2, Reflected Power = 74.12 nW
        retlw   0x74            ; Low Byte, ADRES = 6, Reflected Power = 81.24 nW
        retlw   0xFA            ; Low Byte, ADRES = 10, Reflected Power = 89.05 nW
        retlw   0x8D            ; Low Byte, ADRES = 14, Reflected Power = 97.61 nW
        retlw   0x2E            ; Low Byte, ADRES = 18, Reflected Power = 107.0 nW
        retlw   0xDF            ; Low Byte, ADRES = 22, Reflected Power = 117.3 nW
        retlw   0x50            ; Low Byte, ADRES = 26, Reflected Power = 128.6 nW
        retlw   0xBA            ; Low Byte, ADRES = 30, Reflected Power = 140.9 nW
        retlw   0x2F            ; Low Byte, ADRES = 34, Reflected Power = 154.5 nW
        retlw   0xAE            ; Low Byte, ADRES = 38, Reflected Power = 169.3 nW
        retlw   0x3A            ; Low Byte, ADRES = 42, Reflected Power = 185.6 nW
        retlw   0xD3            ; Low Byte, ADRES = 46, Reflected Power = 203.4 nW
        retlw   0x7B            ; Low Byte, ADRES = 50, Reflected Power = 223.0 nW
        retlw   0x1A            ; Low Byte, ADRES = 54, Reflected Power = 244.4 nW
        retlw   0x7F            ; Low Byte, ADRES = 58, Reflected Power = 267.9 nW
        retlw   0xED            ; Low Byte, ADRES = 62, Reflected Power = 293.7 nW
        retlw   0x66            ; Low Byte, ADRES = 66, Reflected Power = 321.9 nW
        retlw   0xEB            ; Low Byte, ADRES = 70, Reflected Power = 352.8 nW
        retlw   0x7D            ; Low Byte, ADRES = 74, Reflected Power = 386.7 nW
        retlw   0x1D            ; Low Byte, ADRES = 78, Reflected Power = 423.9 nW
        retlw   0xCC            ; Low Byte, ADRES = 82, Reflected Power = 464.7 nW
        retlw   0x46            ; Low Byte, ADRES = 86, Reflected Power = 509.3 nW
        retlw   0xAF            ; Low Byte, ADRES = 90, Reflected Power = 558.3 nW
        retlw   0x22            ; Low Byte, ADRES = 94, Reflected Power = 611.9 nW
        retlw   0xA0            ; Low Byte, ADRES = 98, Reflected Power = 670.8 nW
        retlw   0x2B            ; Low Byte, ADRES = 102, Reflected Power = 735.2 nW
        retlw   0xC3            ; Low Byte, ADRES = 106, Reflected Power = 805.9 nW
        retlw   0x69            ; Low Byte, ADRES = 110, Reflected Power = 883.4 nW
        retlw   0x10            ; Low Byte, ADRES = 114, Reflected Power = 968.3 nW
        retlw   0x74            ; Low Byte, ADRES = 118, Reflected Power = 1.061 uW
        retlw   0xE1            ; Low Byte, ADRES = 122, Reflected Power = 1.163 uW
        retlw   0x59            ; Low Byte, ADRES = 126, Reflected Power = 1.275 uW
        retlw   0xDD            ; Low Byte, ADRES = 130, Reflected Power = 1.398 uW
        retlw   0x6D            ; Low Byte, ADRES = 134, Reflected Power = 1.532 uW
        retlw   0x0B            ; Low Byte, ADRES = 138, Reflected Power = 1.679 uW
        retlw   0xB9            ; Low Byte, ADRES = 142, Reflected Power = 1.841 uW
        retlw   0x3B            ; Low Byte, ADRES = 146, Reflected Power = 2.018 uW
        retlw   0xA3            ; Low Byte, ADRES = 150, Reflected Power = 2.212 uW
        retlw   0x16            ; Low Byte, ADRES = 154, Reflected Power = 2.424 uW
        retlw   0x93            ; Low Byte, ADRES = 158, Reflected Power = 2.657 uW
        retlw   0x1C            ; Low Byte, ADRES = 162, Reflected Power = 2.913 uW
        retlw   0xB2            ; Low Byte, ADRES = 166, Reflected Power = 3.193 uW
        retlw   0x57            ; Low Byte, ADRES = 170, Reflected Power = 3.500 uW
        retlw   0x06            ; Low Byte, ADRES = 174, Reflected Power = 3.836 uW
        retlw   0x69            ; Low Byte, ADRES = 178, Reflected Power = 4.205 uW
        retlw   0xD5            ; Low Byte, ADRES = 182, Reflected Power = 4.609 uW
        retlw   0x4C            ; Low Byte, ADRES = 186, Reflected Power = 5.052 uW
        retlw   0xCF            ; Low Byte, ADRES = 190, Reflected Power = 5.538 uW
        retlw   0x5D            ; Low Byte, ADRES = 194, Reflected Power = 6.070 uW
        retlw   0xFA            ; Low Byte, ADRES = 198, Reflected Power = 6.654 uW
        retlw   0xA6            ; Low Byte, ADRES = 202, Reflected Power = 7.293 uW
        retlw   0x31            ; Low Byte, ADRES = 206, Reflected Power = 7.994 uW
        retlw   0x98            ; Low Byte, ADRES = 210, Reflected Power = 8.763 uW
        retlw   0x09            ; Low Byte, ADRES = 214, Reflected Power = 9.605 uW
        retlw   0x85            ; Low Byte, ADRES = 218, Reflected Power = 10.53 uW
        retlw   0x0D            ; Low Byte, ADRES = 222, Reflected Power = 11.54 uW
        retlw   0xA2            ; Low Byte, ADRES = 226, Reflected Power = 12.65 uW
        retlw   0x45            ; Low Byte, ADRES = 230, Reflected Power = 13.87 uW
        retlw   0xF8            ; Low Byte, ADRES = 234, Reflected Power = 15.20 uW
        retlw   0x5E            ; Low Byte, ADRES = 238, Reflected Power = 16.66 uW
        retlw   0xCA            ; Low Byte, ADRES = 242, Reflected Power = 18.26 uW
        retlw   0x3F            ; Low Byte, ADRES = 246, Reflected Power = 20.02 uW
        retlw   0xC0            ; Low Byte, ADRES = 250, Reflected Power = 21.94 uW
        retlw   0x4E            ; Low Byte, ADRES = 254, Reflected Power = 24.05 uW
        retlw   0xE9            ; Low Byte, ADRES = 258, Reflected Power = 26.36 uW
        retlw   0x93            ; Low Byte, ADRES = 262, Reflected Power = 28.90 uW
        retlw   0x27            ; Low Byte, ADRES = 266, Reflected Power = 31.67 uW
        retlw   0x8D            ; Low Byte, ADRES = 270, Reflected Power = 34.72 uW
        retlw   0xFD            ; Low Byte, ADRES = 274, Reflected Power = 38.05 uW
```

```
retlw      0x78            ; Low Byte, ADRES = 278, Reflected Power = 41.71 uW
retlw      0xFE            ; Low Byte, ADRES = 282, Reflected Power = 45.72 uW
retlw      0x92            ; Low Byte, ADRES = 286, Reflected Power = 50.12 uW
retlw      0x33            ; Low Byte, ADRES = 290, Reflected Power = 54.94 uW
retlw      0xE5            ; Low Byte, ADRES = 294, Reflected Power = 60.22 uW
retlw      0x53            ; Low Byte, ADRES = 298, Reflected Power = 66.00 uW
retlw      0xBE            ; Low Byte, ADRES = 302, Reflected Power = 72.35 uW
retlw      0x33            ; Low Byte, ADRES = 306, Reflected Power = 79.30 uW
retlw      0xB2            ; Low Byte, ADRES = 310, Reflected Power = 86.93 uW
retlw      0x3F            ; Low Byte, ADRES = 314, Reflected Power = 95.28 uW
retlw      0xD8            ; Low Byte, ADRES = 318, Reflected Power = 104.4 uW
retlw      0x81            ; Low Byte, ADRES = 322, Reflected Power = 114.5 uW
retlw      0x1D            ; Low Byte, ADRES = 326, Reflected Power = 125.5 uW
retlw      0x82            ; Low Byte, ADRES = 330, Reflected Power = 137.5 uW
retlw      0xF1            ; Low Byte, ADRES = 334, Reflected Power = 150.8 uW
retlw      0x6A            ; Low Byte, ADRES = 338, Reflected Power = 165.3 uW
retlw      0xF0            ; Low Byte, ADRES = 342, Reflected Power = 181.1 uW
retlw      0x82            ; Low Byte, ADRES = 346, Reflected Power = 198.6 uW
retlw      0x22            ; Low Byte, ADRES = 350, Reflected Power = 217.6 uW
retlw      0xD1            ; Low Byte, ADRES = 354, Reflected Power = 238.6 uW
retlw      0x49            ; Low Byte, ADRES = 358, Reflected Power = 261.5 uW
retlw      0xB2            ; Low Byte, ADRES = 362, Reflected Power = 286.6 uW
retlw      0x26            ; Low Byte, ADRES = 366, Reflected Power = 314.2 uW
retlw      0xA4            ; Low Byte, ADRES = 370, Reflected Power = 344.4 uW
retlw      0x2F            ; Low Byte, ADRES = 374, Reflected Power = 377.5 uW
retlw      0xC8            ; Low Byte, ADRES = 378, Reflected Power = 413.8 uW
retlw      0x6E            ; Low Byte, ADRES = 382, Reflected Power = 453.6 uW
retlw      0x13            ; Low Byte, ADRES = 386, Reflected Power = 497.2 uW
retlw      0x77            ; Low Byte, ADRES = 390, Reflected Power = 544.9 uW
retlw      0xE5            ; Low Byte, ADRES = 394, Reflected Power = 597.3 uW
retlw      0x5D            ; Low Byte, ADRES = 398, Reflected Power = 654.7 uW
retlw      0xE1            ; Low Byte, ADRES = 402, Reflected Power = 717.7 uW
retlw      0x72            ; Low Byte, ADRES = 406, Reflected Power = 786.7 uW
retlw      0x10            ; Low Byte, ADRES = 410, Reflected Power = 862.3 uW
retlw      0xBE            ; Low Byte, ADRES = 414, Reflected Power = 945.2 uW
retlw      0x3E            ; Low Byte, ADRES = 418, Reflected Power = 1.036 mW
retlw      0xA7            ; Low Byte, ADRES = 422, Reflected Power = 1.136 mW
retlw      0x19            ; Low Byte, ADRES = 426, Reflected Power = 1.245 mW
retlw      0x97            ; Low Byte, ADRES = 430, Reflected Power = 1.364 mW
retlw      0x20            ; Low Byte, ADRES = 434, Reflected Power = 1.496 mW
retlw      0xB7            ; Low Byte, ADRES = 438, Reflected Power = 1.639 mW
retlw      0x5C            ; Low Byte, ADRES = 442, Reflected Power = 1.797 mW
retlw      0x09            ; Low Byte, ADRES = 446, Reflected Power = 1.970 mW
retlw      0x6C            ; Low Byte, ADRES = 450, Reflected Power = 2.159 mW
retlw      0xD9            ; Low Byte, ADRES = 454, Reflected Power = 2.367 mW
retlw      0x50            ; Low Byte, ADRES = 458, Reflected Power = 2.594 mW
retlw      0xD3            ; Low Byte, ADRES = 462, Reflected Power = 2.843 mW
retlw      0x62            ; Low Byte, ADRES = 466, Reflected Power = 3.117 mW
retlw      0xFF            ; Low Byte, ADRES = 470, Reflected Power = 3.416 mW
retlw      0xAB            ; Low Byte, ADRES = 474, Reflected Power = 3.745 mW
retlw      0x34            ; Low Byte, ADRES = 478, Reflected Power = 4.105 mW
retlw      0x9B            ; Low Byte, ADRES = 482, Reflected Power = 4.499 mW
retlw      0x0D            ; Low Byte, ADRES = 486, Reflected Power = 4.932 mW
retlw      0x89            ; Low Byte, ADRES = 490, Reflected Power = 5.406 mW
retlw      0x11            ; Low Byte, ADRES = 494, Reflected Power = 5.925 mW
retlw      0xA7            ; Low Byte, ADRES = 498, Reflected Power = 6.495 mW
retlw      0x4A            ; Low Byte, ADRES = 502, Reflected Power = 7.119 mW
retlw      0xFE            ; Low Byte, ADRES = 506, Reflected Power = 7.804 mW
retlw      0x61            ; Low Byte, ADRES = 510, Reflected Power = 8.554 mW
retlw      0xCD            ; Low Byte, ADRES = 514, Reflected Power = 9.376 mW
retlw      0x43            ; Low Byte, ADRES = 518, Reflected Power = 10.28 mW
retlw      0xC5            ; Low Byte, ADRES = 522, Reflected Power = 11.26 mW
retlw      0x52            ; Low Byte, ADRES = 526, Reflected Power = 12.35 mW
retlw      0xEE            ; Low Byte, ADRES = 530, Reflected Power = 13.53 mW
retlw      0x99            ; Low Byte, ADRES = 534, Reflected Power = 14.84 mW
retlw      0x2A            ; Low Byte, ADRES = 538, Reflected Power = 16.26 mW
retlw      0x90            ; Low Byte, ADRES = 542, Reflected Power = 17.83 mW
retlw      0x00            ; Low Byte, ADRES = 546, Reflected Power = 19.54 mW
retlw      0x7C            ; Low Byte, ADRES = 550, Reflected Power = 21.42 mW
retlw      0x02            ; Low Byte, ADRES = 554, Reflected Power = 23.48 mW
retlw      0x96            ; Low Byte, ADRES = 558, Reflected Power = 25.73 mW
retlw      0x38            ; Low Byte, ADRES = 562, Reflected Power = 28.21 mW
retlw      0xEA            ; Low Byte, ADRES = 566, Reflected Power = 30.92 mW
retlw      0x56            ; Low Byte, ADRES = 570, Reflected Power = 33.89 mW
retlw      0xC1            ; Low Byte, ADRES = 574, Reflected Power = 37.15 mW
retlw      0x36            ; Low Byte, ADRES = 578, Reflected Power = 40.72 mW
retlw      0xB6            ; Low Byte, ADRES = 582, Reflected Power = 44.63 mW
retlw      0x43            ; Low Byte, ADRES = 586, Reflected Power = 48.92 mW
retlw      0xDD            ; Low Byte, ADRES = 590, Reflected Power = 53.62 mW
retlw      0x86            ; Low Byte, ADRES = 594, Reflected Power = 58.78 mW
retlw      0x20            ; Low Byte, ADRES = 598, Reflected Power = 64.43 mW
```

```
retlw      0x85            ; Low Byte, ADRES = 602, Reflected Power = 70.62 mW
retlw      0xF4            ; Low Byte, ADRES = 606, Reflected Power = 77.41 mW
retlw      0x6E            ; Low Byte, ADRES = 610, Reflected Power = 84.85 mW
retlw      0xF4            ; Low Byte, ADRES = 614, Reflected Power = 93.01 mW
retlw      0x86            ; Low Byte, ADRES = 618, Reflected Power = 101.9 mW
retlw      0x27            ; Low Byte, ADRES = 622, Reflected Power = 111.7 mW
retlw      0xD7            ; Low Byte, ADRES = 626, Reflected Power = 122.5 mW
retlw      0x4C            ; Low Byte, ADRES = 630, Reflected Power = 134.3 mW
retlw      0xB6            ; Low Byte, ADRES = 634, Reflected Power = 147.2 mW
retlw      0x29            ; Low Byte, ADRES = 638, Reflected Power = 161.3 mW
retlw      0xA9            ; Low Byte, ADRES = 642, Reflected Power = 176.8 mW
retlw      0x34            ; Low Byte, ADRES = 646, Reflected Power = 193.8 mW
retlw      0xCC            ; Low Byte, ADRES = 650, Reflected Power = 212.4 mW
retlw      0x74            ; Low Byte, ADRES = 654, Reflected Power = 232.9 mW
retlw      0x16            ; Low Byte, ADRES = 658, Reflected Power = 255.3 mW
retlw      0x7A            ; Low Byte, ADRES = 662, Reflected Power = 279.8 mW
retlw      0xE8            ; Low Byte, ADRES = 666, Reflected Power = 306.7 mW
retlw      0x61            ; Low Byte, ADRES = 670, Reflected Power = 336.2 mW
retlw      0xE5            ; Low Byte, ADRES = 674, Reflected Power = 368.5 mW
retlw      0x76            ; Low Byte, ADRES = 678, Reflected Power = 403.9 mW
retlw      0x15            ; Low Byte, ADRES = 682, Reflected Power = 442.7 mW
retlw      0xC4            ; Low Byte, ADRES = 686, Reflected Power = 485.3 mW
retlw      0x41            ; Low Byte, ADRES = 690, Reflected Power = 531.9 mW
retlw      0xAA            ; Low Byte, ADRES = 694, Reflected Power = 583.1 mW
retlw      0x1D            ; Low Byte, ADRES = 698, Reflected Power = 639.1 mW
retlw      0x9B            ; Low Byte, ADRES = 702, Reflected Power = 700.5 mW
retlw      0x25            ; Low Byte, ADRES = 706, Reflected Power = 767.9 mW
retlw      0xBC            ; Low Byte, ADRES = 710, Reflected Power = 841.7 mW
retlw      0x61            ; Low Byte, ADRES = 714, Reflected Power = 922.6 mW
retlw      0x0C            ; Low Byte, ADRES = 718, Reflected Power = 1.011 W
retlw      0x6F            ; Low Byte, ADRES = 722, Reflected Power = 1.108 W
retlw      0xDC            ; Low Byte, ADRES = 726, Reflected Power = 1.215 W
retlw      0x54            ; Low Byte, ADRES = 730, Reflected Power = 1.332 W
retlw      0xD7            ; Low Byte, ADRES = 734, Reflected Power = 1.460 W
retlw      0x67            ; Low Byte, ADRES = 738, Reflected Power = 1.600 W
retlw      0x04            ; Low Byte, ADRES = 742, Reflected Power = 1.754 W
retlw      0xB1            ; Low Byte, ADRES = 746, Reflected Power = 1.923 W
retlw      0x37            ; Low Byte, ADRES = 750, Reflected Power = 2.107 W
retlw      0x9F            ; Low Byte, ADRES = 754, Reflected Power = 2.310 W
retlw      0x10            ; Low Byte, ADRES = 758, Reflected Power = 2.532 W
retlw      0x8D            ; Low Byte, ADRES = 762, Reflected Power = 2.775 W
retlw      0x16            ; Low Byte, ADRES = 766, Reflected Power = 3.042 W
retlw      0xAB            ; Low Byte, ADRES = 770, Reflected Power = 3.335 W
retlw      0x4F            ; Low Byte, ADRES = 774, Reflected Power = 3.655 W
retlw      0x02            ; Low Byte, ADRES = 778, Reflected Power = 4.007 W
retlw      0x64            ; Low Byte, ADRES = 782, Reflected Power = 4.392 W
retlw      0xD0            ; Low Byte, ADRES = 786, Reflected Power = 4.814 W
retlw      0x47            ; Low Byte, ADRES = 790, Reflected Power = 5.277 W
retlw      0xC9            ; Low Byte, ADRES = 794, Reflected Power = 5.784 W
retlw      0x57            ; Low Byte, ADRES = 798, Reflected Power = 6.340 W
retlw      0xF3            ; Low Byte, ADRES = 802, Reflected Power = 6.949 W
retlw      0x9E            ; Low Byte, ADRES = 806, Reflected Power = 7.617 W
retlw      0x2D            ; Low Byte, ADRES = 810, Reflected Power = 8.349 W
retlw      0x93            ; Low Byte, ADRES = 814, Reflected Power = 9.152 W
retlw      0x04            ; Low Byte, ADRES = 818, Reflected Power = 10.03 W
retlw      0x7F            ; Low Byte, ADRES = 822, Reflected Power = 11.00 W
retlw      0x07            ; Low Byte, ADRES = 826, Reflected Power = 12.05 W
retlw      0x9B            ; Low Byte, ADRES = 830, Reflected Power = 13.21 W
retlw      0x3E            ; Low Byte, ADRES = 834, Reflected Power = 14.48 W
retlw      0xF0            ; Low Byte, ADRES = 838, Reflected Power = 15.87 W
retlw      0x5A            ; Low Byte, ADRES = 842, Reflected Power = 17.40 W
retlw      0xC5            ; Low Byte, ADRES = 846, Reflected Power = 19.07 W
retlw      0x3A            ; Low Byte, ADRES = 850, Reflected Power = 20.90 W
retlw      0xBB            ; Low Byte, ADRES = 854, Reflected Power = 22.91 W
retlw      0x48            ; Low Byte, ADRES = 858, Reflected Power = 25.12 W
retlw      0xE2            ; Low Byte, ADRES = 862, Reflected Power = 27.53 W
retlw      0x8B            ; Low Byte, ADRES = 866, Reflected Power = 30.18 W
retlw      0x23            ; Low Byte, ADRES = 870, Reflected Power = 33.08 W
retlw      0x88            ; Low Byte, ADRES = 874, Reflected Power = 36.26 W
retlw      0xF8            ; Low Byte, ADRES = 878, Reflected Power = 39.74 W
retlw      0x72            ; Low Byte, ADRES = 882, Reflected Power = 43.56 W
retlw      0xF8            ; Low Byte, ADRES = 886, Reflected Power = 47.75 W
retlw      0x8B            ; Low Byte, ADRES = 890, Reflected Power = 52.34 W
retlw      0x2C            ; Low Byte, ADRES = 894, Reflected Power = 57.37 W
retlw      0xDC            ; Low Byte, ADRES = 898, Reflected Power = 62.89 W
retlw      0x4F            ; Low Byte, ADRES = 902, Reflected Power = 68.93 W
retlw      0xB9            ; Low Byte, ADRES = 906, Reflected Power = 75.56 W
retlw      0x2D            ; Low Byte, ADRES = 910, Reflected Power = 82.82 W
retlw      0xAD            ; Low Byte, ADRES = 914, Reflected Power = 90.78 W
retlw      0x38            ; Low Byte, ADRES = 918, Reflected Power = 99.51 W
retlw      0xD1            ; Low Byte, ADRES = 922, Reflected Power = 109.1 W
```

```
        retlw   0x79            ; Low Byte, ADRES = 926, Reflected Power = 119.6 W
        retlw   0x18            ; Low Byte, ADRES = 930, Reflected Power = 131.1 W
        retlw   0x7D            ; Low Byte, ADRES = 934, Reflected Power = 143.7 W
        retlw   0xEC            ; Low Byte, ADRES = 938, Reflected Power = 157.5 W
        retlw   0x65            ; Low Byte, ADRES = 942, Reflected Power = 172.6 W
        retlw   0xEA            ; Low Byte, ADRES = 946, Reflected Power = 189.2 W
        retlw   0x7B            ; Low Byte, ADRES = 950, Reflected Power = 207.4 W
        retlw   0x1A            ; Low Byte, ADRES = 954, Reflected Power = 227.3 W
        retlw   0xC9            ; Low Byte, ADRES = 958, Reflected Power = 249.2 W
        retlw   0x44            ; Low Byte, ADRES = 962, Reflected Power = 273.1 W
        retlw   0xAD            ; Low Byte, ADRES = 966, Reflected Power = 299.4 W
        retlw   0x21            ; Low Byte, ADRES = 970, Reflected Power = 328.1 W
        retlw   0x9F            ; Low Byte, ADRES = 974, Reflected Power = 359.7 W
        retlw   0x29            ; Low Byte, ADRES = 978, Reflected Power = 394.2 W
        retlw   0xC1            ; Low Byte, ADRES = 982, Reflected Power = 432.1 W
        retlw   0x67            ; Low Byte, ADRES = 986, Reflected Power = 473.7 W
        retlw   0x0E            ; Low Byte, ADRES = 990, Reflected Power = 519.2 W
        retlw   0x72            ; Low Byte, ADRES = 994, Reflected Power = 569.1 W
        retlw   0xE0            ; Low Byte, ADRES = 998, Reflected Power = 623.8 W
        retlw   0x58            ; Low Byte, ADRES = 1002, Reflected Power = 683.8 W
        retlw   0xDB            ; Low Byte, ADRES = 1006, Reflected Power = 749.5 W
        retlw   0x6B            ; Low Byte, ADRES = 1010, Reflected Power = 821.6 W
        retlw   0x09            ; Low Byte, ADRES = 1014, Reflected Power = 900.6 W
        retlw   0xB6            ; Low Byte, ADRES = 1018, Reflected Power = 987.1 W
        retlw   0x3A            ; Low Byte, ADRES = 1022, Reflected Power = 1,082.0 W

RLowByteTable3
        retlw   0x17            ; Low Byte, ADRES = 3, Reflected Power = 75.84 nW
        retlw   0x94            ; Low Byte, ADRES = 7, Reflected Power = 83.13 nW
        retlw   0x1D            ; Low Byte, ADRES = 11, Reflected Power = 91.12 nW
        retlw   0xB4            ; Low Byte, ADRES = 15, Reflected Power = 99.88 nW
        retlw   0x59            ; Low Byte, ADRES = 19, Reflected Power = 109.5 nW
        retlw   0x07            ; Low Byte, ADRES = 23, Reflected Power = 120.0 nW
        retlw   0x6A            ; Low Byte, ADRES = 27, Reflected Power = 131.5 nW
        retlw   0xD7            ; Low Byte, ADRES = 31, Reflected Power = 144.2 nW
        retlw   0x4E            ; Low Byte, ADRES = 35, Reflected Power = 158.0 nW
        retlw   0xD0            ; Low Byte, ADRES = 39, Reflected Power = 173.2 nW
        retlw   0x5F            ; Low Byte, ADRES = 43, Reflected Power = 189.9 nW
        retlw   0xFC            ; Low Byte, ADRES = 47, Reflected Power = 208.1 nW
        retlw   0xA8            ; Low Byte, ADRES = 51, Reflected Power = 228.1 nW
        retlw   0x32            ; Low Byte, ADRES = 55, Reflected Power = 250.1 nW
        retlw   0x99            ; Low Byte, ADRES = 59, Reflected Power = 274.1 nW
        retlw   0x0B            ; Low Byte, ADRES = 63, Reflected Power = 300.5 nW
        retlw   0x87            ; Low Byte, ADRES = 67, Reflected Power = 329.3 nW
        retlw   0x0F            ; Low Byte, ADRES = 71, Reflected Power = 361.0 nW
        retlw   0xA4            ; Low Byte, ADRES = 75, Reflected Power = 395.7 nW
        retlw   0x47            ; Low Byte, ADRES = 79, Reflected Power = 433.7 nW
        retlw   0xFA            ; Low Byte, ADRES = 83, Reflected Power = 475.4 nW
        retlw   0x5F            ; Low Byte, ADRES = 87, Reflected Power = 521.1 nW
        retlw   0xCB            ; Low Byte, ADRES = 91, Reflected Power = 571.2 nW
        retlw   0x41            ; Low Byte, ADRES = 95, Reflected Power = 626.1 nW
        retlw   0xC2            ; Low Byte, ADRES = 99, Reflected Power = 686.3 nW
        retlw   0x50            ; Low Byte, ADRES = 103, Reflected Power = 752.3 nW
        retlw   0xEB            ; Low Byte, ADRES = 107, Reflected Power = 824.6 nW
        retlw   0x95            ; Low Byte, ADRES = 111, Reflected Power = 903.9 nW
        retlw   0x28            ; Low Byte, ADRES = 115, Reflected Power = 990.8 nW
        retlw   0x8E            ; Low Byte, ADRES = 119, Reflected Power = 1.086 uW
        retlw   0xFE            ; Low Byte, ADRES = 123, Reflected Power = 1.190 uW
        retlw   0x79            ; Low Byte, ADRES = 127, Reflected Power = 1.305 uW
        retlw   0x00            ; Low Byte, ADRES = 131, Reflected Power = 1.430 uW
        retlw   0x93            ; Low Byte, ADRES = 135, Reflected Power = 1.568 uW
        retlw   0x35            ; Low Byte, ADRES = 139, Reflected Power = 1.718 uW
        retlw   0xE7            ; Low Byte, ADRES = 143, Reflected Power = 1.884 uW
        retlw   0x54            ; Low Byte, ADRES = 147, Reflected Power = 2.065 uW
        retlw   0xBF            ; Low Byte, ADRES = 151, Reflected Power = 2.263 uW
        retlw   0x34            ; Low Byte, ADRES = 155, Reflected Power = 2.481 uW
        retlw   0xB4            ; Low Byte, ADRES = 159, Reflected Power = 2.719 uW
        retlw   0x40            ; Low Byte, ADRES = 163, Reflected Power = 2.981 uW
        retlw   0xDA            ; Low Byte, ADRES = 167, Reflected Power = 3.267 uW
        retlw   0x83            ; Low Byte, ADRES = 171, Reflected Power = 3.581 uW
        retlw   0x1E            ; Low Byte, ADRES = 175, Reflected Power = 3.925 uW
        retlw   0x83            ; Low Byte, ADRES = 179, Reflected Power = 4.303 uW
        retlw   0xF2            ; Low Byte, ADRES = 183, Reflected Power = 4.716 uW
        retlw   0x6C            ; Low Byte, ADRES = 187, Reflected Power = 5.170 uW
        retlw   0xF1            ; Low Byte, ADRES = 191, Reflected Power = 5.667 uW
        retlw   0x83            ; Low Byte, ADRES = 195, Reflected Power = 6.211 uW
        retlw   0x24            ; Low Byte, ADRES = 199, Reflected Power = 6.808 uW
        retlw   0xD3            ; Low Byte, ADRES = 203, Reflected Power = 7.463 uW
        retlw   0x4A            ; Low Byte, ADRES = 207, Reflected Power = 8.180 uW
        retlw   0xB3            ; Low Byte, ADRES = 211, Reflected Power = 8.966 uW
        retlw   0x27            ; Low Byte, ADRES = 215, Reflected Power = 9.828 uW
```

```
retlw      0xA6            ; Low Byte, ADRES = 219, Reflected Power = 10.77 uW
retlw      0x31            ; Low Byte, ADRES = 223, Reflected Power = 11.81 uW
retlw      0xC9            ; Low Byte, ADRES = 227, Reflected Power = 12.94 uW
retlw      0x70            ; Low Byte, ADRES = 231, Reflected Power = 14.19 uW
retlw      0x14            ; Low Byte, ADRES = 235, Reflected Power = 15.55 uW
retlw      0x78            ; Low Byte, ADRES = 239, Reflected Power = 17.05 uW
retlw      0xE6            ; Low Byte, ADRES = 243, Reflected Power = 18.69 uW
retlw      0x5E            ; Low Byte, ADRES = 247, Reflected Power = 20.48 uW
retlw      0xE3            ; Low Byte, ADRES = 251, Reflected Power = 22.45 uW
retlw      0x73            ; Low Byte, ADRES = 255, Reflected Power = 24.61 uW
retlw      0x12            ; Low Byte, ADRES = 259, Reflected Power = 26.97 uW
retlw      0xC0            ; Low Byte, ADRES = 263, Reflected Power = 29.57 uW
retlw      0x3F            ; Low Byte, ADRES = 267, Reflected Power = 32.41 uW
retlw      0xA8            ; Low Byte, ADRES = 271, Reflected Power = 35.52 uW
retlw      0x1B            ; Low Byte, ADRES = 275, Reflected Power = 38.94 uW
retlw      0x98            ; Low Byte, ADRES = 279, Reflected Power = 42.68 uW
retlw      0x22            ; Low Byte, ADRES = 283, Reflected Power = 46.78 uW
retlw      0xB9            ; Low Byte, ADRES = 287, Reflected Power = 51.28 uW
retlw      0x5E            ; Low Byte, ADRES = 291, Reflected Power = 56.21 uW
retlw      0x0A            ; Low Byte, ADRES = 295, Reflected Power = 61.61 uW
retlw      0x6D            ; Low Byte, ADRES = 299, Reflected Power = 67.54 uW
retlw      0xDA            ; Low Byte, ADRES = 303, Reflected Power = 74.03 uW
retlw      0x51            ; Low Byte, ADRES = 307, Reflected Power = 81.14 uW
retlw      0xD4            ; Low Byte, ADRES = 311, Reflected Power = 88.94 uW
retlw      0x64            ; Low Byte, ADRES = 315, Reflected Power = 97.49 uW
retlw      0x01            ; Low Byte, ADRES = 319, Reflected Power = 106.9 uW
retlw      0xAD            ; Low Byte, ADRES = 323, Reflected Power = 117.1 uW
retlw      0x35            ; Low Byte, ADRES = 327, Reflected Power = 128.4 uW
retlw      0x9D            ; Low Byte, ADRES = 331, Reflected Power = 140.7 uW
retlw      0x0E            ; Low Byte, ADRES = 335, Reflected Power = 154.3 uW
retlw      0x8B            ; Low Byte, ADRES = 339, Reflected Power = 169.1 uW
retlw      0x13            ; Low Byte, ADRES = 343, Reflected Power = 185.4 uW
retlw      0xA8            ; Low Byte, ADRES = 347, Reflected Power = 203.2 uW
retlw      0x4C            ; Low Byte, ADRES = 351, Reflected Power = 222.7 uW
retlw      0x00            ; Low Byte, ADRES = 355, Reflected Power = 244.1 uW
retlw      0x62            ; Low Byte, ADRES = 359, Reflected Power = 267.6 uW
retlw      0xCE            ; Low Byte, ADRES = 363, Reflected Power = 293.3 uW
retlw      0x44            ; Low Byte, ADRES = 367, Reflected Power = 321.5 uW
retlw      0xC6            ; Low Byte, ADRES = 371, Reflected Power = 352.4 uW
retlw      0x54            ; Low Byte, ADRES = 375, Reflected Power = 386.3 uW
retlw      0xF0            ; Low Byte, ADRES = 379, Reflected Power = 423.4 uW
retlw      0x9B            ; Low Byte, ADRES = 383, Reflected Power = 464.1 uW
retlw      0x2B            ; Low Byte, ADRES = 387, Reflected Power = 508.7 uW
retlw      0x91            ; Low Byte, ADRES = 391, Reflected Power = 557.6 uW
retlw      0x02            ; Low Byte, ADRES = 395, Reflected Power = 611.2 uW
retlw      0x7D            ; Low Byte, ADRES = 399, Reflected Power = 669.9 uW
retlw      0x04            ; Low Byte, ADRES = 403, Reflected Power = 734.3 uW
retlw      0x98            ; Low Byte, ADRES = 407, Reflected Power = 804.9 uW
retlw      0x3A            ; Low Byte, ADRES = 411, Reflected Power = 882.3 uW
retlw      0xEC            ; Low Byte, ADRES = 415, Reflected Power = 967.1 uW
retlw      0x58            ; Low Byte, ADRES = 419, Reflected Power = 1.060 mW
retlw      0xC2            ; Low Byte, ADRES = 423, Reflected Power = 1.162 mW
retlw      0x38            ; Low Byte, ADRES = 427, Reflected Power = 1.274 mW
retlw      0xB8            ; Low Byte, ADRES = 431, Reflected Power = 1.396 mW
retlw      0x45            ; Low Byte, ADRES = 435, Reflected Power = 1.530 mW
retlw      0xDF            ; Low Byte, ADRES = 439, Reflected Power = 1.677 mW
retlw      0x88            ; Low Byte, ADRES = 443, Reflected Power = 1.839 mW
retlw      0x21            ; Low Byte, ADRES = 447, Reflected Power = 2.015 mW
retlw      0x86            ; Low Byte, ADRES = 451, Reflected Power = 2.209 mW
retlw      0xF6            ; Low Byte, ADRES = 455, Reflected Power = 2.421 mW
retlw      0x70            ; Low Byte, ADRES = 459, Reflected Power = 2.654 mW
retlw      0xF5            ; Low Byte, ADRES = 463, Reflected Power = 2.909 mW
retlw      0x88            ; Low Byte, ADRES = 467, Reflected Power = 3.189 mW
retlw      0x29            ; Low Byte, ADRES = 471, Reflected Power = 3.496 mW
retlw      0xD9            ; Low Byte, ADRES = 475, Reflected Power = 3.832 mW
retlw      0x4D            ; Low Byte, ADRES = 479, Reflected Power = 4.200 mW
retlw      0xB7            ; Low Byte, ADRES = 483, Reflected Power = 4.604 mW
retlw      0x2B            ; Low Byte, ADRES = 487, Reflected Power = 5.046 mW
retlw      0xAA            ; Low Byte, ADRES = 491, Reflected Power = 5.531 mW
retlw      0x35            ; Low Byte, ADRES = 495, Reflected Power = 6.063 mW
retlw      0xCE            ; Low Byte, ADRES = 499, Reflected Power = 6.646 mW
retlw      0x76            ; Low Byte, ADRES = 503, Reflected Power = 7.284 mW
retlw      0x17            ; Low Byte, ADRES = 507, Reflected Power = 7.985 mW
retlw      0x7B            ; Low Byte, ADRES = 511, Reflected Power = 8.752 mW
retlw      0xE9            ; Low Byte, ADRES = 515, Reflected Power = 9.593 mW
retlw      0x62            ; Low Byte, ADRES = 519, Reflected Power = 10.52 mW
retlw      0xE7            ; Low Byte, ADRES = 523, Reflected Power = 11.53 mW
retlw      0x78            ; Low Byte, ADRES = 527, Reflected Power = 12.63 mW
retlw      0x17            ; Low Byte, ADRES = 531, Reflected Power = 13.85 mW
retlw      0xC6            ; Low Byte, ADRES = 535, Reflected Power = 15.18 mW
retlw      0x42            ; Low Byte, ADRES = 539, Reflected Power = 16.64 mW
```

```
retlw      0xAB            ; Low Byte, ADRES = 543, Reflected Power = 18.24 mW
retlw      0x1E            ; Low Byte, ADRES = 547, Reflected Power = 19.99 mW
retlw      0x9C            ; Low Byte, ADRES = 551, Reflected Power = 21.91 mW
retlw      0x26            ; Low Byte, ADRES = 555, Reflected Power = 24.02 mW
retlw      0xBE            ; Low Byte, ADRES = 559, Reflected Power = 26.33 mW
retlw      0x63            ; Low Byte, ADRES = 563, Reflected Power = 28.86 mW
retlw      0x0D            ; Low Byte, ADRES = 567, Reflected Power = 31.63 mW
retlw      0x70            ; Low Byte, ADRES = 571, Reflected Power = 34.67 mW
retlw      0xDD            ; Low Byte, ADRES = 575, Reflected Power = 38.01 mW
retlw      0x55            ; Low Byte, ADRES = 579, Reflected Power = 41.66 mW
retlw      0xD8            ; Low Byte, ADRES = 583, Reflected Power = 45.67 mW
retlw      0x68            ; Low Byte, ADRES = 587, Reflected Power = 50.06 mW
retlw      0x06            ; Low Byte, ADRES = 591, Reflected Power = 54.87 mW
retlw      0xB3            ; Low Byte, ADRES = 595, Reflected Power = 60.14 mW
retlw      0x38            ; Low Byte, ADRES = 599, Reflected Power = 65.92 mW
retlw      0xA0            ; Low Byte, ADRES = 603, Reflected Power = 72.26 mW
retlw      0x12            ; Low Byte, ADRES = 607, Reflected Power = 79.21 mW
retlw      0x8E            ; Low Byte, ADRES = 611, Reflected Power = 86.82 mW
retlw      0x17            ; Low Byte, ADRES = 615, Reflected Power = 95.16 mW
retlw      0xAD            ; Low Byte, ADRES = 619, Reflected Power = 104.3 mW
retlw      0x51            ; Low Byte, ADRES = 623, Reflected Power = 114.3 mW
retlw      0x03            ; Low Byte, ADRES = 627, Reflected Power = 125.3 mW
retlw      0x65            ; Low Byte, ADRES = 631, Reflected Power = 137.4 mW
retlw      0xD2            ; Low Byte, ADRES = 635, Reflected Power = 150.6 mW
retlw      0x48            ; Low Byte, ADRES = 639, Reflected Power = 165.1 mW
retlw      0xCA            ; Low Byte, ADRES = 643, Reflected Power = 180.9 mW
retlw      0x59            ; Low Byte, ADRES = 647, Reflected Power = 198.3 mW
retlw      0xF5            ; Low Byte, ADRES = 651, Reflected Power = 217.4 mW
retlw      0xA0            ; Low Byte, ADRES = 655, Reflected Power = 238.3 mW
retlw      0x2E            ; Low Byte, ADRES = 659, Reflected Power = 261.2 mW
retlw      0x95            ; Low Byte, ADRES = 663, Reflected Power = 286.3 mW
retlw      0x05            ; Low Byte, ADRES = 667, Reflected Power = 313.8 mW
retlw      0x81            ; Low Byte, ADRES = 671, Reflected Power = 344.0 mW
retlw      0x08            ; Low Byte, ADRES = 675, Reflected Power = 377.0 mW
retlw      0x9D            ; Low Byte, ADRES = 679, Reflected Power = 413.3 mW
retlw      0x3F            ; Low Byte, ADRES = 683, Reflected Power = 453.0 mW
retlw      0xF2            ; Low Byte, ADRES = 687, Reflected Power = 496.5 mW
retlw      0x5B            ; Low Byte, ADRES = 691, Reflected Power = 544.3 mW
retlw      0xC6            ; Low Byte, ADRES = 695, Reflected Power = 596.6 mW
retlw      0x3B            ; Low Byte, ADRES = 699, Reflected Power = 653.9 mW
retlw      0xBC            ; Low Byte, ADRES = 703, Reflected Power = 716.8 mW
retlw      0x49            ; Low Byte, ADRES = 707, Reflected Power = 785.7 mW
retlw      0xE4            ; Low Byte, ADRES = 711, Reflected Power = 861.2 mW
retlw      0x8D            ; Low Byte, ADRES = 715, Reflected Power = 944.0 mW
retlw      0x24            ; Low Byte, ADRES = 719, Reflected Power = 1.035 W
retlw      0x89            ; Low Byte, ADRES = 723, Reflected Power = 1.134 W
retlw      0xF9            ; Low Byte, ADRES = 727, Reflected Power = 1.243 W
retlw      0x73            ; Low Byte, ADRES = 731, Reflected Power = 1.363 W
retlw      0xFA            ; Low Byte, ADRES = 735, Reflected Power = 1.494 W
retlw      0x8D            ; Low Byte, ADRES = 739, Reflected Power = 1.637 W
retlw      0x2E            ; Low Byte, ADRES = 743, Reflected Power = 1.795 W
retlw      0xDE            ; Low Byte, ADRES = 747, Reflected Power = 1.967 W
retlw      0x50            ; Low Byte, ADRES = 751, Reflected Power = 2.156 W
retlw      0xBA            ; Low Byte, ADRES = 755, Reflected Power = 2.364 W
retlw      0x2F            ; Low Byte, ADRES = 759, Reflected Power = 2.591 W
retlw      0xAE            ; Low Byte, ADRES = 763, Reflected Power = 2.840 W
retlw      0x3A            ; Low Byte, ADRES = 767, Reflected Power = 3.113 W
retlw      0xD3            ; Low Byte, ADRES = 771, Reflected Power = 3.412 W
retlw      0x7B            ; Low Byte, ADRES = 775, Reflected Power = 3.740 W
retlw      0x19            ; Low Byte, ADRES = 779, Reflected Power = 4.100 W
retlw      0x7E            ; Low Byte, ADRES = 783, Reflected Power = 4.494 W
retlw      0xED            ; Low Byte, ADRES = 787, Reflected Power = 4.926 W
retlw      0x66            ; Low Byte, ADRES = 791, Reflected Power = 5.399 W
retlw      0xEB            ; Low Byte, ADRES = 795, Reflected Power = 5.918 W
retlw      0x7D            ; Low Byte, ADRES = 799, Reflected Power = 6.487 W
retlw      0x1C            ; Low Byte, ADRES = 803, Reflected Power = 7.110 W
retlw      0xCB            ; Low Byte, ADRES = 807, Reflected Power = 7.794 W
retlw      0x46            ; Low Byte, ADRES = 811, Reflected Power = 8.543 W
retlw      0xAF            ; Low Byte, ADRES = 815, Reflected Power = 9.364 W
retlw      0x22            ; Low Byte, ADRES = 819, Reflected Power = 10.26 W
retlw      0xA0            ; Low Byte, ADRES = 823, Reflected Power = 11.25 W
retlw      0x2B            ; Low Byte, ADRES = 827, Reflected Power = 12.33 W
retlw      0xC2            ; Low Byte, ADRES = 831, Reflected Power = 13.52 W
retlw      0x69            ; Low Byte, ADRES = 835, Reflected Power = 14.82 W
retlw      0x0F            ; Low Byte, ADRES = 839, Reflected Power = 16.24 W
retlw      0x73            ; Low Byte, ADRES = 843, Reflected Power = 17.80 W
retlw      0xE1            ; Low Byte, ADRES = 847, Reflected Power = 19.51 W
retlw      0x59            ; Low Byte, ADRES = 851, Reflected Power = 21.39 W
retlw      0xDD            ; Low Byte, ADRES = 855, Reflected Power = 23.45 W
retlw      0x6D            ; Low Byte, ADRES = 859, Reflected Power = 25.70 W
retlw      0x0B            ; Low Byte, ADRES = 863, Reflected Power = 28.17 W
```

```
        retlw   0xB8            ; Low Byte, ADRES = 867, Reflected Power = 30.88 W
        retlw   0x3B            ; Low Byte, ADRES = 871, Reflected Power = 33.85 W
        retlw   0xA3            ; Low Byte, ADRES = 875, Reflected Power = 37.10 W
        retlw   0x15            ; Low Byte, ADRES = 879, Reflected Power = 40.67 W
        retlw   0x92            ; Low Byte, ADRES = 883, Reflected Power = 44.58 W
        retlw   0x1C            ; Low Byte, ADRES = 887, Reflected Power = 48.86 W
        retlw   0xB2            ; Low Byte, ADRES = 891, Reflected Power = 53.56 W
        retlw   0x57            ; Low Byte, ADRES = 895, Reflected Power = 58.70 W
        retlw   0x06            ; Low Byte, ADRES = 899, Reflected Power = 64.35 W
        retlw   0x69            ; Low Byte, ADRES = 903, Reflected Power = 70.53 W
        retlw   0xD5            ; Low Byte, ADRES = 907, Reflected Power = 77.31 W
        retlw   0x4C            ; Low Byte, ADRES = 911, Reflected Power = 84.75 W
        retlw   0xCE            ; Low Byte, ADRES = 915, Reflected Power = 92.89 W
        retlw   0x5D            ; Low Byte, ADRES = 919, Reflected Power = 101.8 W
        retlw   0xFA            ; Low Byte, ADRES = 923, Reflected Power = 111.6 W
        retlw   0xA5            ; Low Byte, ADRES = 927, Reflected Power = 122.3 W
        retlw   0x31            ; Low Byte, ADRES = 931, Reflected Power = 134.1 W
        retlw   0x98            ; Low Byte, ADRES = 935, Reflected Power = 147.0 W
        retlw   0x09            ; Low Byte, ADRES = 939, Reflected Power = 161.1 W
        retlw   0x85            ; Low Byte, ADRES = 943, Reflected Power = 176.6 W
        retlw   0x0D            ; Low Byte, ADRES = 947, Reflected Power = 193.6 W
        retlw   0xA1            ; Low Byte, ADRES = 951, Reflected Power = 212.2 W
        retlw   0x45            ; Low Byte, ADRES = 955, Reflected Power = 232.6 W
        retlw   0xF8            ; Low Byte, ADRES = 959, Reflected Power = 254.9 W
        retlw   0x5E            ; Low Byte, ADRES = 963, Reflected Power = 279.4 W
        retlw   0xC9            ; Low Byte, ADRES = 967, Reflected Power = 306.3 W
        retlw   0x3F            ; Low Byte, ADRES = 971, Reflected Power = 335.7 W
        retlw   0xC0            ; Low Byte, ADRES = 975, Reflected Power = 368.0 W
        retlw   0x4E            ; Low Byte, ADRES = 979, Reflected Power = 403.4 W
        retlw   0xE9            ; Low Byte, ADRES = 983, Reflected Power = 442.2 W
        retlw   0x93            ; Low Byte, ADRES = 987, Reflected Power = 484.7 W
        retlw   0x27            ; Low Byte, ADRES = 991, Reflected Power = 531.3 W
        retlw   0x8D            ; Low Byte, ADRES = 995, Reflected Power = 582.3 W
        retlw   0xFD            ; Low Byte, ADRES = 999, Reflected Power = 638.3 W
        retlw   0x77            ; Low Byte, ADRES = 1003, Reflected Power = 699.7 W
        retlw   0xFE            ; Low Byte, ADRES = 1007, Reflected Power = 766.9 W
        retlw   0x91            ; Low Byte, ADRES = 1011, Reflected Power = 840.6 W
        retlw   0x33            ; Low Byte, ADRES = 1015, Reflected Power = 921.5 W
        retlw   0xE4            ; Low Byte, ADRES = 1019, Reflected Power = 1,010.0 W
        retlw   0x53            ; Low Byte, ADRES = 1023, Reflected Power = 1,107.1 W

RHighByteTable0
        retlw   0x0C            ; High Byte, ADRES = 0, Reflected Power = 70.79 nW
        retlw   0x0D            ; High Byte, ADRES = 4, Reflected Power = 77.60 nW
        retlw   0x0D            ; High Byte, ADRES = 8, Reflected Power = 85.06 nW
        retlw   0x0E            ; High Byte, ADRES = 12, Reflected Power = 93.24 nW
        retlw   0x0E            ; High Byte, ADRES = 16, Reflected Power = 102.2 nW
        retlw   0x0F            ; High Byte, ADRES = 20, Reflected Power = 112.0 nW
        retlw   0x10            ; High Byte, ADRES = 24, Reflected Power = 122.8 nW
        retlw   0x10            ; High Byte, ADRES = 28, Reflected Power = 134.6 nW
        retlw   0x10            ; High Byte, ADRES = 32, Reflected Power = 147.5 nW
        retlw   0x11            ; High Byte, ADRES = 36, Reflected Power = 161.7 nW
        retlw   0x11            ; High Byte, ADRES = 40, Reflected Power = 177.3 nW
        retlw   0x12            ; High Byte, ADRES = 44, Reflected Power = 194.3 nW
        retlw   0x13            ; High Byte, ADRES = 48, Reflected Power = 213.0 nW
        retlw   0x13            ; High Byte, ADRES = 52, Reflected Power = 233.4 nW
        retlw   0x14            ; High Byte, ADRES = 56, Reflected Power = 255.9 nW
        retlw   0x14            ; High Byte, ADRES = 60, Reflected Power = 280.5 nW
        retlw   0x15            ; High Byte, ADRES = 64, Reflected Power = 307.4 nW
        retlw   0x15            ; High Byte, ADRES = 68, Reflected Power = 337.0 nW
        retlw   0x16            ; High Byte, ADRES = 72, Reflected Power = 369.4 nW
        retlw   0x16            ; High Byte, ADRES = 76, Reflected Power = 404.9 nW
        retlw   0x17            ; High Byte, ADRES = 80, Reflected Power = 443.8 nW
        retlw   0x18            ; High Byte, ADRES = 84, Reflected Power = 486.5 nW
        retlw   0x18            ; High Byte, ADRES = 88, Reflected Power = 533.2 nW
        retlw   0x18            ; High Byte, ADRES = 92, Reflected Power = 584.5 nW
        retlw   0x19            ; High Byte, ADRES = 96, Reflected Power = 640.7 nW
        retlw   0x19            ; High Byte, ADRES = 100, Reflected Power = 702.3 nW
        retlw   0x1A            ; High Byte, ADRES = 104, Reflected Power = 769.8 nW
        retlw   0x1B            ; High Byte, ADRES = 108, Reflected Power = 843.8 nW
        retlw   0x1B            ; High Byte, ADRES = 112, Reflected Power = 924.9 nW
        retlw   0x1C            ; High Byte, ADRES = 116, Reflected Power = 1.014 uW
        retlw   0x1C            ; High Byte, ADRES = 120, Reflected Power = 1.111 uW
        retlw   0x1D            ; High Byte, ADRES = 124, Reflected Power = 1.218 uW
        retlw   0x1D            ; High Byte, ADRES = 128, Reflected Power = 1.335 uW
        retlw   0x1E            ; High Byte, ADRES = 132, Reflected Power = 1.463 uW
        retlw   0x1E            ; High Byte, ADRES = 136, Reflected Power = 1.604 uW
        retlw   0x1F            ; High Byte, ADRES = 140, Reflected Power = 1.758 uW
        retlw   0x20            ; High Byte, ADRES = 144, Reflected Power = 1.927 uW
        retlw   0x20            ; High Byte, ADRES = 148, Reflected Power = 2.113 uW
        retlw   0x20            ; High Byte, ADRES = 152, Reflected Power = 2.316 uW
```

```
retlw      0x21            ; High Byte, ADRES = 156, Reflected Power = 2.538 uW
retlw      0x21            ; High Byte, ADRES = 160, Reflected Power = 2.782 uW
retlw      0x22            ; High Byte, ADRES = 164, Reflected Power = 3.050 uW
retlw      0x23            ; High Byte, ADRES = 168, Reflected Power = 3.343 uW
retlw      0x23            ; High Byte, ADRES = 172, Reflected Power = 3.664 uW
retlw      0x24            ; High Byte, ADRES = 176, Reflected Power = 4.016 uW
retlw      0x24            ; High Byte, ADRES = 180, Reflected Power = 4.403 uW
retlw      0x25            ; High Byte, ADRES = 184, Reflected Power = 4.826 uW
retlw      0x25            ; High Byte, ADRES = 188, Reflected Power = 5.290 uW
retlw      0x26            ; High Byte, ADRES = 192, Reflected Power = 5.798 uW
retlw      0x26            ; High Byte, ADRES = 196, Reflected Power = 6.355 uW
retlw      0x27            ; High Byte, ADRES = 200, Reflected Power = 6.966 uW
retlw      0x28            ; High Byte, ADRES = 204, Reflected Power = 7.636 uW
retlw      0x28            ; High Byte, ADRES = 208, Reflected Power = 8.370 uW
retlw      0x28            ; High Byte, ADRES = 212, Reflected Power = 9.174 uW
retlw      0x29            ; High Byte, ADRES = 216, Reflected Power = 10.06 uW
retlw      0x29            ; High Byte, ADRES = 220, Reflected Power = 11.02 uW
retlw      0x2A            ; High Byte, ADRES = 224, Reflected Power = 12.08 uW
retlw      0x2A            ; High Byte, ADRES = 228, Reflected Power = 13.24 uW
retlw      0x2B            ; High Byte, ADRES = 232, Reflected Power = 14.52 uW
retlw      0x2C            ; High Byte, ADRES = 236, Reflected Power = 15.91 uW
retlw      0x2C            ; High Byte, ADRES = 240, Reflected Power = 17.44 uW
retlw      0x2D            ; High Byte, ADRES = 244, Reflected Power = 19.12 uW
retlw      0x2D            ; High Byte, ADRES = 248, Reflected Power = 20.96 uW
retlw      0x2E            ; High Byte, ADRES = 252, Reflected Power = 22.97 uW
retlw      0x2E            ; High Byte, ADRES = 256, Reflected Power = 25.18 uW
retlw      0x2F            ; High Byte, ADRES = 260, Reflected Power = 27.60 uW
retlw      0x2F            ; High Byte, ADRES = 264, Reflected Power = 30.25 uW
retlw      0x30            ; High Byte, ADRES = 268, Reflected Power = 33.16 uW
retlw      0x30            ; High Byte, ADRES = 272, Reflected Power = 36.35 uW
retlw      0x31            ; High Byte, ADRES = 276, Reflected Power = 39.84 uW
retlw      0x31            ; High Byte, ADRES = 280, Reflected Power = 43.67 uW
retlw      0x32            ; High Byte, ADRES = 284, Reflected Power = 47.87 uW
retlw      0x32            ; High Byte, ADRES = 288, Reflected Power = 52.47 uW
retlw      0x33            ; High Byte, ADRES = 292, Reflected Power = 57.51 uW
retlw      0x34            ; High Byte, ADRES = 296, Reflected Power = 63.04 uW
retlw      0x34            ; High Byte, ADRES = 300, Reflected Power = 69.10 uW
retlw      0x34            ; High Byte, ADRES = 304, Reflected Power = 75.75 uW
retlw      0x35            ; High Byte, ADRES = 308, Reflected Power = 83.03 uW
retlw      0x35            ; High Byte, ADRES = 312, Reflected Power = 91.01 uW
retlw      0x36            ; High Byte, ADRES = 316, Reflected Power = 99.76 uW
retlw      0x37            ; High Byte, ADRES = 320, Reflected Power = 109.3 uW
retlw      0x37            ; High Byte, ADRES = 324, Reflected Power = 119.9 uW
retlw      0x38            ; High Byte, ADRES = 328, Reflected Power = 131.4 uW
retlw      0x38            ; High Byte, ADRES = 332, Reflected Power = 144.0 uW
retlw      0x39            ; High Byte, ADRES = 336, Reflected Power = 157.8 uW
retlw      0x39            ; High Byte, ADRES = 340, Reflected Power = 173.0 uW
retlw      0x3A            ; High Byte, ADRES = 344, Reflected Power = 189.7 uW
retlw      0x3A            ; High Byte, ADRES = 348, Reflected Power = 207.9 uW
retlw      0x3B            ; High Byte, ADRES = 352, Reflected Power = 227.9 uW
retlw      0x3C            ; High Byte, ADRES = 356, Reflected Power = 249.8 uW
retlw      0x3C            ; High Byte, ADRES = 360, Reflected Power = 273.8 uW
retlw      0x3C            ; High Byte, ADRES = 364, Reflected Power = 300.1 uW
retlw      0x3D            ; High Byte, ADRES = 368, Reflected Power = 328.9 uW
retlw      0x3D            ; High Byte, ADRES = 372, Reflected Power = 360.6 uW
retlw      0x3E            ; High Byte, ADRES = 376, Reflected Power = 395.2 uW
retlw      0x3F            ; High Byte, ADRES = 380, Reflected Power = 433.2 uW
retlw      0x3F            ; High Byte, ADRES = 384, Reflected Power = 474.9 uW
retlw      0x40            ; High Byte, ADRES = 388, Reflected Power = 520.5 uW
retlw      0x40            ; High Byte, ADRES = 392, Reflected Power = 570.5 uW
retlw      0x41            ; High Byte, ADRES = 396, Reflected Power = 625.4 uW
retlw      0x41            ; High Byte, ADRES = 400, Reflected Power = 685.5 uW
retlw      0x42            ; High Byte, ADRES = 404, Reflected Power = 751.4 uW
retlw      0x42            ; High Byte, ADRES = 408, Reflected Power = 823.6 uW
retlw      0x43            ; High Byte, ADRES = 412, Reflected Power = 902.8 uW
retlw      0x44            ; High Byte, ADRES = 416, Reflected Power = 989.6 uW
retlw      0x44            ; High Byte, ADRES = 420, Reflected Power = 1.085 mW
retlw      0x44            ; High Byte, ADRES = 424, Reflected Power = 1.189 mW
retlw      0x45            ; High Byte, ADRES = 428, Reflected Power = 1.303 mW
retlw      0x45            ; High Byte, ADRES = 432, Reflected Power = 1.428 mW
retlw      0x46            ; High Byte, ADRES = 436, Reflected Power = 1.566 mW
retlw      0x47            ; High Byte, ADRES = 440, Reflected Power = 1.716 mW
retlw      0x47            ; High Byte, ADRES = 444, Reflected Power = 1.881 mW
retlw      0x48            ; High Byte, ADRES = 448, Reflected Power = 2.062 mW
retlw      0x48            ; High Byte, ADRES = 452, Reflected Power = 2.260 mW
retlw      0x49            ; High Byte, ADRES = 456, Reflected Power = 2.478 mW
retlw      0x49            ; High Byte, ADRES = 460, Reflected Power = 2.716 mW
retlw      0x4A            ; High Byte, ADRES = 464, Reflected Power = 2.977 mW
retlw      0x4A            ; High Byte, ADRES = 468, Reflected Power = 3.263 mW
retlw      0x4B            ; High Byte, ADRES = 472, Reflected Power = 3.577 mW
retlw      0x4C            ; High Byte, ADRES = 476, Reflected Power = 3.920 mW
```

```
retlw      0x4C            ; High Byte, ADRES = 480, Reflected Power = 4.297 mW
retlw      0x4C            ; High Byte, ADRES = 484, Reflected Power = 4.710 mW
retlw      0x4D            ; High Byte, ADRES = 488, Reflected Power = 5.163 mW
retlw      0x4D            ; High Byte, ADRES = 492, Reflected Power = 5.660 mW
retlw      0x4E            ; High Byte, ADRES = 496, Reflected Power = 6.204 mW
retlw      0x4E            ; High Byte, ADRES = 500, Reflected Power = 6.800 mW
retlw      0x4F            ; High Byte, ADRES = 504, Reflected Power = 7.453 mW
retlw      0x50            ; High Byte, ADRES = 508, Reflected Power = 8.170 mW
retlw      0x50            ; High Byte, ADRES = 512, Reflected Power = 8.955 mW
retlw      0x51            ; High Byte, ADRES = 516, Reflected Power = 9.816 mW
retlw      0x51            ; High Byte, ADRES = 520, Reflected Power = 10.76 mW
retlw      0x52            ; High Byte, ADRES = 524, Reflected Power = 11.79 mW
retlw      0x52            ; High Byte, ADRES = 528, Reflected Power = 12.93 mW
retlw      0x53            ; High Byte, ADRES = 532, Reflected Power = 14.17 mW
retlw      0x53            ; High Byte, ADRES = 536, Reflected Power = 15.53 mW
retlw      0x54            ; High Byte, ADRES = 540, Reflected Power = 17.03 mW
retlw      0x54            ; High Byte, ADRES = 544, Reflected Power = 18.66 mW
retlw      0x55            ; High Byte, ADRES = 548, Reflected Power = 20.46 mW
retlw      0x55            ; High Byte, ADRES = 552, Reflected Power = 22.42 mW
retlw      0x56            ; High Byte, ADRES = 556, Reflected Power = 24.58 mW
retlw      0x56            ; High Byte, ADRES = 560, Reflected Power = 26.94 mW
retlw      0x57            ; High Byte, ADRES = 564, Reflected Power = 29.53 mW
retlw      0x58            ; High Byte, ADRES = 568, Reflected Power = 32.37 mW
retlw      0x58            ; High Byte, ADRES = 572, Reflected Power = 35.48 mW
retlw      0x58            ; High Byte, ADRES = 576, Reflected Power = 38.89 mW
retlw      0x59            ; High Byte, ADRES = 580, Reflected Power = 42.63 mW
retlw      0x59            ; High Byte, ADRES = 584, Reflected Power = 46.73 mW
retlw      0x5A            ; High Byte, ADRES = 588, Reflected Power = 51.22 mW
retlw      0x5B            ; High Byte, ADRES = 592, Reflected Power = 56.14 mW
retlw      0x5B            ; High Byte, ADRES = 596, Reflected Power = 61.54 mW
retlw      0x5C            ; High Byte, ADRES = 600, Reflected Power = 67.45 mW
retlw      0x5C            ; High Byte, ADRES = 604, Reflected Power = 73.94 mW
retlw      0x5D            ; High Byte, ADRES = 608, Reflected Power = 81.04 mW
retlw      0x5D            ; High Byte, ADRES = 612, Reflected Power = 88.83 mW
retlw      0x5E            ; High Byte, ADRES = 616, Reflected Power = 97.37 mW
retlw      0x5E            ; High Byte, ADRES = 620, Reflected Power = 106.7 mW
retlw      0x5F            ; High Byte, ADRES = 624, Reflected Power = 117.0 mW
retlw      0x60            ; High Byte, ADRES = 628, Reflected Power = 128.2 mW
retlw      0x60            ; High Byte, ADRES = 632, Reflected Power = 140.6 mW
retlw      0x60            ; High Byte, ADRES = 636, Reflected Power = 154.1 mW
retlw      0x61            ; High Byte, ADRES = 640, Reflected Power = 168.9 mW
retlw      0x61            ; High Byte, ADRES = 644, Reflected Power = 185.1 mW
retlw      0x62            ; High Byte, ADRES = 648, Reflected Power = 202.9 mW
retlw      0x63            ; High Byte, ADRES = 652, Reflected Power = 222.4 mW
retlw      0x63            ; High Byte, ADRES = 656, Reflected Power = 243.8 mW
retlw      0x64            ; High Byte, ADRES = 660, Reflected Power = 267.2 mW
retlw      0x64            ; High Byte, ADRES = 664, Reflected Power = 292.9 mW
retlw      0x65            ; High Byte, ADRES = 668, Reflected Power = 321.1 mW
retlw      0x65            ; High Byte, ADRES = 672, Reflected Power = 351.9 mW
retlw      0x66            ; High Byte, ADRES = 676, Reflected Power = 385.8 mW
retlw      0x66            ; High Byte, ADRES = 680, Reflected Power = 422.9 mW
retlw      0x67            ; High Byte, ADRES = 684, Reflected Power = 463.5 mW
retlw      0x68            ; High Byte, ADRES = 688, Reflected Power = 508.1 mW
retlw      0x68            ; High Byte, ADRES = 692, Reflected Power = 556.9 mW
retlw      0x68            ; High Byte, ADRES = 696, Reflected Power = 610.4 mW
retlw      0x69            ; High Byte, ADRES = 700, Reflected Power = 669.1 mW
retlw      0x69            ; High Byte, ADRES = 704, Reflected Power = 733.4 mW
retlw      0x6A            ; High Byte, ADRES = 708, Reflected Power = 803.9 mW
retlw      0x6B            ; High Byte, ADRES = 712, Reflected Power = 881.2 mW
retlw      0x6B            ; High Byte, ADRES = 716, Reflected Power = 965.9 mW
retlw      0x6C            ; High Byte, ADRES = 720, Reflected Power = 1.059 W
retlw      0x6C            ; High Byte, ADRES = 724, Reflected Power = 1.161 W
retlw      0x6D            ; High Byte, ADRES = 728, Reflected Power = 1.272 W
retlw      0x6D            ; High Byte, ADRES = 732, Reflected Power = 1.394 W
retlw      0x6E            ; High Byte, ADRES = 736, Reflected Power = 1.528 W
retlw      0x6E            ; High Byte, ADRES = 740, Reflected Power = 1.675 W
retlw      0x6F            ; High Byte, ADRES = 744, Reflected Power = 1.836 W
retlw      0x70            ; High Byte, ADRES = 748, Reflected Power = 2.013 W
retlw      0x70            ; High Byte, ADRES = 752, Reflected Power = 2.206 W
retlw      0x70            ; High Byte, ADRES = 756, Reflected Power = 2.418 W
retlw      0x71            ; High Byte, ADRES = 760, Reflected Power = 2.651 W
retlw      0x71            ; High Byte, ADRES = 764, Reflected Power = 2.906 W
retlw      0x72            ; High Byte, ADRES = 768, Reflected Power = 3.185 W
retlw      0x72            ; High Byte, ADRES = 772, Reflected Power = 3.491 W
retlw      0x73            ; High Byte, ADRES = 776, Reflected Power = 3.827 W
retlw      0x74            ; High Byte, ADRES = 780, Reflected Power = 4.195 W
retlw      0x74            ; High Byte, ADRES = 784, Reflected Power = 4.598 W
retlw      0x75            ; High Byte, ADRES = 788, Reflected Power = 5.040 W
retlw      0x75            ; High Byte, ADRES = 792, Reflected Power = 5.524 W
retlw      0x76            ; High Byte, ADRES = 796, Reflected Power = 6.055 W
retlw      0x76            ; High Byte, ADRES = 800, Reflected Power = 6.637 W
```

```
      retlw      0x77            ; High Byte, ADRES = 804, Reflected Power = 7.275 W
      retlw      0x77            ; High Byte, ADRES = 808, Reflected Power = 7.975 W
      retlw      0x78            ; High Byte, ADRES = 812, Reflected Power = 8.741 W
      retlw      0x78            ; High Byte, ADRES = 816, Reflected Power = 9.582 W
      retlw      0x79            ; High Byte, ADRES = 820, Reflected Power = 10.50 W
      retlw      0x79            ; High Byte, ADRES = 824, Reflected Power = 11.51 W
      retlw      0x7A            ; High Byte, ADRES = 828, Reflected Power = 12.62 W
      retlw      0x7A            ; High Byte, ADRES = 832, Reflected Power = 13.83 W
      retlw      0x7B            ; High Byte, ADRES = 836, Reflected Power = 15.16 W
      retlw      0x7C            ; High Byte, ADRES = 840, Reflected Power = 16.62 W
      retlw      0x7C            ; High Byte, ADRES = 844, Reflected Power = 18.22 W
      retlw      0x7C            ; High Byte, ADRES = 848, Reflected Power = 19.97 W
      retlw      0x7D            ; High Byte, ADRES = 852, Reflected Power = 21.89 W
      retlw      0x7D            ; High Byte, ADRES = 856, Reflected Power = 23.99 W
      retlw      0x7E            ; High Byte, ADRES = 860, Reflected Power = 26.30 W
      retlw      0x7F            ; High Byte, ADRES = 864, Reflected Power = 28.82 W
      retlw      0x7F            ; High Byte, ADRES = 868, Reflected Power = 31.60 W
      retlw      0x80            ; High Byte, ADRES = 872, Reflected Power = 34.63 W
      retlw      0x80            ; High Byte, ADRES = 876, Reflected Power = 37.96 W
      retlw      0x81            ; High Byte, ADRES = 880, Reflected Power = 41.61 W
      retlw      0x81            ; High Byte, ADRES = 884, Reflected Power = 45.61 W
      retlw      0x82            ; High Byte, ADRES = 888, Reflected Power = 49.99 W
      retlw      0x82            ; High Byte, ADRES = 892, Reflected Power = 54.80 W
      retlw      0x83            ; High Byte, ADRES = 896, Reflected Power = 60.07 W
      retlw      0x84            ; High Byte, ADRES = 900, Reflected Power = 65.84 W
      retlw      0x84            ; High Byte, ADRES = 904, Reflected Power = 72.17 W
      retlw      0x84            ; High Byte, ADRES = 908, Reflected Power = 79.11 W
      retlw      0x85            ; High Byte, ADRES = 912, Reflected Power = 86.71 W
      retlw      0x85            ; High Byte, ADRES = 916, Reflected Power = 95.05 W
      retlw      0x86            ; High Byte, ADRES = 920, Reflected Power = 104.2 W
      retlw      0x87            ; High Byte, ADRES = 924, Reflected Power = 114.2 W
      retlw      0x87            ; High Byte, ADRES = 928, Reflected Power = 125.2 W
      retlw      0x88            ; High Byte, ADRES = 932, Reflected Power = 137.2 W
      retlw      0x88            ; High Byte, ADRES = 936, Reflected Power = 150.4 W
      retlw      0x89            ; High Byte, ADRES = 940, Reflected Power = 164.9 W
      retlw      0x89            ; High Byte, ADRES = 944, Reflected Power = 180.7 W
      retlw      0x8A            ; High Byte, ADRES = 948, Reflected Power = 198.1 W
      retlw      0x8A            ; High Byte, ADRES = 952, Reflected Power = 217.1 W
      retlw      0x8B            ; High Byte, ADRES = 956, Reflected Power = 238.0 W
      retlw      0x8C            ; High Byte, ADRES = 960, Reflected Power = 260.9 W
      retlw      0x8C            ; High Byte, ADRES = 964, Reflected Power = 285.9 W
      retlw      0x8C            ; High Byte, ADRES = 968, Reflected Power = 313.4 W
      retlw      0x8D            ; High Byte, ADRES = 972, Reflected Power = 343.5 W
      retlw      0x8D            ; High Byte, ADRES = 976, Reflected Power = 376.6 W
      retlw      0x8E            ; High Byte, ADRES = 980, Reflected Power = 412.8 W
      retlw      0x8F            ; High Byte, ADRES = 984, Reflected Power = 452.4 W
      retlw      0x8F            ; High Byte, ADRES = 988, Reflected Power = 495.9 W
      retlw      0x90            ; High Byte, ADRES = 992, Reflected Power = 543.6 W
      retlw      0x90            ; High Byte, ADRES = 996, Reflected Power = 595.9 W
      retlw      0x91            ; High Byte, ADRES = 1000, Reflected Power = 653.1 W
      retlw      0x91            ; High Byte, ADRES = 1004, Reflected Power = 715.9 W
      retlw      0x92            ; High Byte, ADRES = 1008, Reflected Power = 784.7 W
      retlw      0x92            ; High Byte, ADRES = 1012, Reflected Power = 860.2 W
      retlw      0x93            ; High Byte, ADRES = 1016, Reflected Power = 942.8 W
      retlw      0x94            ; High Byte, ADRES = 1020, Reflected Power = 1,033.5 W

RHighByteTable1
      retlw      0x0C            ; High Byte, ADRES = 1, Reflected Power = 72.44 nW
      retlw      0x0D            ; High Byte, ADRES = 5, Reflected Power = 79.40 nW
      retlw      0x0D            ; High Byte, ADRES = 9, Reflected Power = 87.03 nW
      retlw      0x0E            ; High Byte, ADRES = 13, Reflected Power = 95.40 nW
      retlw      0x0F            ; High Byte, ADRES = 17, Reflected Power = 104.6 nW
      retlw      0x0F            ; High Byte, ADRES = 21, Reflected Power = 114.6 nW
      retlw      0x10            ; High Byte, ADRES = 25, Reflected Power = 125.6 nW
      retlw      0x10            ; High Byte, ADRES = 29, Reflected Power = 137.7 nW
      retlw      0x11            ; High Byte, ADRES = 33, Reflected Power = 151.0 nW
      retlw      0x11            ; High Byte, ADRES = 37, Reflected Power = 165.5 nW
      retlw      0x12            ; High Byte, ADRES = 41, Reflected Power = 181.4 nW
      retlw      0x12            ; High Byte, ADRES = 45, Reflected Power = 198.8 nW
      retlw      0x13            ; High Byte, ADRES = 49, Reflected Power = 217.9 nW
      retlw      0x14            ; High Byte, ADRES = 53, Reflected Power = 238.9 nW
      retlw      0x14            ; High Byte, ADRES = 57, Reflected Power = 261.8 nW
      retlw      0x14            ; High Byte, ADRES = 61, Reflected Power = 287.0 nW
      retlw      0x15            ; High Byte, ADRES = 65, Reflected Power = 314.6 nW
      retlw      0x15            ; High Byte, ADRES = 69, Reflected Power = 344.8 nW
      retlw      0x16            ; High Byte, ADRES = 73, Reflected Power = 378.0 nW
      retlw      0x16            ; High Byte, ADRES = 77, Reflected Power = 414.3 nW
      retlw      0x17            ; High Byte, ADRES = 81, Reflected Power = 454.1 nW
      retlw      0x18            ; High Byte, ADRES = 85, Reflected Power = 497.8 nW
      retlw      0x18            ; High Byte, ADRES = 89, Reflected Power = 545.6 nW
      retlw      0x19            ; High Byte, ADRES = 93, Reflected Power = 598.1 nW
```

```
retlw      0x19            ; High Byte, ADRES = 97, Reflected Power = 655.5 nW
retlw      0x1A            ; High Byte, ADRES = 101, Reflected Power = 718.6 nW
retlw      0x1A            ; High Byte, ADRES = 105, Reflected Power = 787.6 nW
retlw      0x1B            ; High Byte, ADRES = 109, Reflected Power = 863.3 nW
retlw      0x1B            ; High Byte, ADRES = 113, Reflected Power = 946.3 nW
retlw      0x1C            ; High Byte, ADRES = 117, Reflected Power = 1.037 uW
retlw      0x1C            ; High Byte, ADRES = 121, Reflected Power = 1.137 uW
retlw      0x1D            ; High Byte, ADRES = 125, Reflected Power = 1.246 uW
retlw      0x1D            ; High Byte, ADRES = 129, Reflected Power = 1.366 uW
retlw      0x1E            ; High Byte, ADRES = 133, Reflected Power = 1.497 uW
retlw      0x1E            ; High Byte, ADRES = 137, Reflected Power = 1.641 uW
retlw      0x1F            ; High Byte, ADRES = 141, Reflected Power = 1.799 uW
retlw      0x20            ; High Byte, ADRES = 145, Reflected Power = 1.972 uW
retlw      0x20            ; High Byte, ADRES = 149, Reflected Power = 2.162 uW
retlw      0x20            ; High Byte, ADRES = 153, Reflected Power = 2.369 uW
retlw      0x21            ; High Byte, ADRES = 157, Reflected Power = 2.597 uW
retlw      0x21            ; High Byte, ADRES = 161, Reflected Power = 2.847 uW
retlw      0x22            ; High Byte, ADRES = 165, Reflected Power = 3.121 uW
retlw      0x23            ; High Byte, ADRES = 169, Reflected Power = 3.420 uW
retlw      0x23            ; High Byte, ADRES = 173, Reflected Power = 3.749 uW
retlw      0x24            ; High Byte, ADRES = 177, Reflected Power = 4.110 uW
retlw      0x24            ; High Byte, ADRES = 181, Reflected Power = 4.505 uW
retlw      0x25            ; High Byte, ADRES = 185, Reflected Power = 4.938 uW
retlw      0x25            ; High Byte, ADRES = 189, Reflected Power = 5.412 uW
retlw      0x26            ; High Byte, ADRES = 193, Reflected Power = 5.933 uW
retlw      0x26            ; High Byte, ADRES = 197, Reflected Power = 6.503 uW
retlw      0x27            ; High Byte, ADRES = 201, Reflected Power = 7.128 uW
retlw      0x28            ; High Byte, ADRES = 205, Reflected Power = 7.813 uW
retlw      0x28            ; High Byte, ADRES = 209, Reflected Power = 8.564 uW
retlw      0x28            ; High Byte, ADRES = 213, Reflected Power = 9.387 uW
retlw      0x29            ; High Byte, ADRES = 217, Reflected Power = 10.29 uW
retlw      0x29            ; High Byte, ADRES = 221, Reflected Power = 11.28 uW
retlw      0x2A            ; High Byte, ADRES = 225, Reflected Power = 12.36 uW
retlw      0x2B            ; High Byte, ADRES = 229, Reflected Power = 13.55 uW
retlw      0x2B            ; High Byte, ADRES = 233, Reflected Power = 14.85 uW
retlw      0x2C            ; High Byte, ADRES = 237, Reflected Power = 16.28 uW
retlw      0x2C            ; High Byte, ADRES = 241, Reflected Power = 17.85 uW
retlw      0x2D            ; High Byte, ADRES = 245, Reflected Power = 19.56 uW
retlw      0x2D            ; High Byte, ADRES = 249, Reflected Power = 21.44 uW
retlw      0x2E            ; High Byte, ADRES = 253, Reflected Power = 23.50 uW
retlw      0x2E            ; High Byte, ADRES = 257, Reflected Power = 25.76 uW
retlw      0x2F            ; High Byte, ADRES = 261, Reflected Power = 28.24 uW
retlw      0x30            ; High Byte, ADRES = 265, Reflected Power = 30.95 uW
retlw      0x30            ; High Byte, ADRES = 269, Reflected Power = 33.93 uW
retlw      0x30            ; High Byte, ADRES = 273, Reflected Power = 37.19 uW
retlw      0x31            ; High Byte, ADRES = 277, Reflected Power = 40.77 uW
retlw      0x31            ; High Byte, ADRES = 281, Reflected Power = 44.69 uW
retlw      0x32            ; High Byte, ADRES = 285, Reflected Power = 48.98 uW
retlw      0x33            ; High Byte, ADRES = 289, Reflected Power = 53.69 uW
retlw      0x33            ; High Byte, ADRES = 293, Reflected Power = 58.85 uW
retlw      0x34            ; High Byte, ADRES = 297, Reflected Power = 64.51 uW
retlw      0x34            ; High Byte, ADRES = 301, Reflected Power = 70.71 uW
retlw      0x35            ; High Byte, ADRES = 305, Reflected Power = 77.50 uW
retlw      0x35            ; High Byte, ADRES = 309, Reflected Power = 84.95 uW
retlw      0x36            ; High Byte, ADRES = 313, Reflected Power = 93.12 uW
retlw      0x36            ; High Byte, ADRES = 317, Reflected Power = 102.1 uW
retlw      0x37            ; High Byte, ADRES = 321, Reflected Power = 111.9 uW
retlw      0x38            ; High Byte, ADRES = 325, Reflected Power = 122.6 uW
retlw      0x38            ; High Byte, ADRES = 329, Reflected Power = 134.4 uW
retlw      0x38            ; High Byte, ADRES = 333, Reflected Power = 147.3 uW
retlw      0x39            ; High Byte, ADRES = 337, Reflected Power = 161.5 uW
retlw      0x39            ; High Byte, ADRES = 341, Reflected Power = 177.0 uW
retlw      0x3A            ; High Byte, ADRES = 345, Reflected Power = 194.1 uW
retlw      0x3A            ; High Byte, ADRES = 349, Reflected Power = 212.7 uW
retlw      0x3B            ; High Byte, ADRES = 353, Reflected Power = 233.2 uW
retlw      0x3C            ; High Byte, ADRES = 357, Reflected Power = 255.6 uW
retlw      0x3C            ; High Byte, ADRES = 361, Reflected Power = 280.1 uW
retlw      0x3D            ; High Byte, ADRES = 365, Reflected Power = 307.1 uW
retlw      0x3D            ; High Byte, ADRES = 369, Reflected Power = 336.6 uW
retlw      0x3E            ; High Byte, ADRES = 373, Reflected Power = 368.9 uW
retlw      0x3E            ; High Byte, ADRES = 377, Reflected Power = 404.4 uW
retlw      0x3F            ; High Byte, ADRES = 381, Reflected Power = 443.3 uW
retlw      0x3F            ; High Byte, ADRES = 385, Reflected Power = 485.9 uW
retlw      0x40            ; High Byte, ADRES = 389, Reflected Power = 532.6 uW
retlw      0x40            ; High Byte, ADRES = 393, Reflected Power = 583.8 uW
retlw      0x41            ; High Byte, ADRES = 397, Reflected Power = 639.9 uW
retlw      0x41            ; High Byte, ADRES = 401, Reflected Power = 701.4 uW
retlw      0x42            ; High Byte, ADRES = 405, Reflected Power = 768.8 uW
retlw      0x42            ; High Byte, ADRES = 409, Reflected Power = 842.7 uW
retlw      0x43            ; High Byte, ADRES = 413, Reflected Power = 923.7 uW
retlw      0x44            ; High Byte, ADRES = 417, Reflected Power = 1.013 mW
```

```
retlw      0x44            ; High Byte, ADRES = 421, Reflected Power = 1.110 mW
retlw      0x44            ; High Byte, ADRES = 425, Reflected Power = 1.217 mW
retlw      0x45            ; High Byte, ADRES = 429, Reflected Power = 1.333 mW
retlw      0x45            ; High Byte, ADRES = 433, Reflected Power = 1.462 mW
retlw      0x46            ; High Byte, ADRES = 437, Reflected Power = 1.602 mW
retlw      0x47            ; High Byte, ADRES = 441, Reflected Power = 1.756 mW
retlw      0x47            ; High Byte, ADRES = 445, Reflected Power = 1.925 mW
retlw      0x48            ; High Byte, ADRES = 449, Reflected Power = 2.110 mW
retlw      0x48            ; High Byte, ADRES = 453, Reflected Power = 2.313 mW
retlw      0x49            ; High Byte, ADRES = 457, Reflected Power = 2.535 mW
retlw      0x49            ; High Byte, ADRES = 461, Reflected Power = 2.779 mW
retlw      0x4A            ; High Byte, ADRES = 465, Reflected Power = 3.046 mW
retlw      0x4A            ; High Byte, ADRES = 469, Reflected Power = 3.339 mW
retlw      0x4B            ; High Byte, ADRES = 473, Reflected Power = 3.660 mW
retlw      0x4C            ; High Byte, ADRES = 477, Reflected Power = 4.011 mW
retlw      0x4C            ; High Byte, ADRES = 481, Reflected Power = 4.397 mW
retlw      0x4C            ; High Byte, ADRES = 485, Reflected Power = 4.820 mW
retlw      0x4D            ; High Byte, ADRES = 489, Reflected Power = 5.283 mW
retlw      0x4D            ; High Byte, ADRES = 493, Reflected Power = 5.791 mW
retlw      0x4E            ; High Byte, ADRES = 497, Reflected Power = 6.348 mW
retlw      0x4F            ; High Byte, ADRES = 501, Reflected Power = 6.958 mW
retlw      0x4F            ; High Byte, ADRES = 505, Reflected Power = 7.626 mW
retlw      0x50            ; High Byte, ADRES = 509, Reflected Power = 8.360 mW
retlw      0x50            ; High Byte, ADRES = 513, Reflected Power = 9.163 mW
retlw      0x51            ; High Byte, ADRES = 517, Reflected Power = 10.04 mW
retlw      0x51            ; High Byte, ADRES = 521, Reflected Power = 11.01 mW
retlw      0x52            ; High Byte, ADRES = 525, Reflected Power = 12.07 mW
retlw      0x52            ; High Byte, ADRES = 529, Reflected Power = 13.23 mW
retlw      0x53            ; High Byte, ADRES = 533, Reflected Power = 14.50 mW
retlw      0x54            ; High Byte, ADRES = 537, Reflected Power = 15.89 mW
retlw      0x54            ; High Byte, ADRES = 541, Reflected Power = 17.42 mW
retlw      0x54            ; High Byte, ADRES = 545, Reflected Power = 19.10 mW
retlw      0x55            ; High Byte, ADRES = 549, Reflected Power = 20.93 mW
retlw      0x55            ; High Byte, ADRES = 553, Reflected Power = 22.94 mW
retlw      0x56            ; High Byte, ADRES = 557, Reflected Power = 25.15 mW
retlw      0x57            ; High Byte, ADRES = 561, Reflected Power = 27.57 mW
retlw      0x57            ; High Byte, ADRES = 565, Reflected Power = 30.22 mW
retlw      0x58            ; High Byte, ADRES = 569, Reflected Power = 33.12 mW
retlw      0x58            ; High Byte, ADRES = 573, Reflected Power = 36.30 mW
retlw      0x59            ; High Byte, ADRES = 577, Reflected Power = 39.79 mW
retlw      0x59            ; High Byte, ADRES = 581, Reflected Power = 43.62 mW
retlw      0x5A            ; High Byte, ADRES = 585, Reflected Power = 47.81 mW
retlw      0x5A            ; High Byte, ADRES = 589, Reflected Power = 52.41 mW
retlw      0x5B            ; High Byte, ADRES = 593, Reflected Power = 57.44 mW
retlw      0x5C            ; High Byte, ADRES = 597, Reflected Power = 62.97 mW
retlw      0x5C            ; High Byte, ADRES = 601, Reflected Power = 69.02 mW
retlw      0x5C            ; High Byte, ADRES = 605, Reflected Power = 75.65 mW
retlw      0x5D            ; High Byte, ADRES = 609, Reflected Power = 82.92 mW
retlw      0x5D            ; High Byte, ADRES = 613, Reflected Power = 90.90 mW
retlw      0x5E            ; High Byte, ADRES = 617, Reflected Power = 99.63 mW
retlw      0x5E            ; High Byte, ADRES = 621, Reflected Power = 109.2 mW
retlw      0x5F            ; High Byte, ADRES = 625, Reflected Power = 119.7 mW
retlw      0x60            ; High Byte, ADRES = 629, Reflected Power = 131.2 mW
retlw      0x60            ; High Byte, ADRES = 633, Reflected Power = 143.8 mW
retlw      0x61            ; High Byte, ADRES = 637, Reflected Power = 157.7 mW
retlw      0x61            ; High Byte, ADRES = 641, Reflected Power = 172.8 mW
retlw      0x62            ; High Byte, ADRES = 645, Reflected Power = 189.4 mW
retlw      0x62            ; High Byte, ADRES = 649, Reflected Power = 207.6 mW
retlw      0x63            ; High Byte, ADRES = 653, Reflected Power = 227.6 mW
retlw      0x63            ; High Byte, ADRES = 657, Reflected Power = 249.5 mW
retlw      0x64            ; High Byte, ADRES = 661, Reflected Power = 273.4 mW
retlw      0x64            ; High Byte, ADRES = 665, Reflected Power = 299.7 mW
retlw      0x65            ; High Byte, ADRES = 669, Reflected Power = 328.5 mW
retlw      0x65            ; High Byte, ADRES = 673, Reflected Power = 360.1 mW
retlw      0x66            ; High Byte, ADRES = 677, Reflected Power = 394.7 mW
retlw      0x66            ; High Byte, ADRES = 681, Reflected Power = 432.7 mW
retlw      0x67            ; High Byte, ADRES = 685, Reflected Power = 474.3 mW
retlw      0x68            ; High Byte, ADRES = 689, Reflected Power = 519.9 mW
retlw      0x68            ; High Byte, ADRES = 693, Reflected Power = 569.8 mW
retlw      0x68            ; High Byte, ADRES = 697, Reflected Power = 624.6 mW
retlw      0x69            ; High Byte, ADRES = 701, Reflected Power = 684.6 mW
retlw      0x6A            ; High Byte, ADRES = 705, Reflected Power = 750.5 mW
retlw      0x6A            ; High Byte, ADRES = 709, Reflected Power = 822.6 mW
retlw      0x6B            ; High Byte, ADRES = 713, Reflected Power = 901.7 mW
retlw      0x6B            ; High Byte, ADRES = 717, Reflected Power = 988.3 mW
retlw      0x6C            ; High Byte, ADRES = 721, Reflected Power = 1.083 W
retlw      0x6C            ; High Byte, ADRES = 725, Reflected Power = 1.187 W
retlw      0x6D            ; High Byte, ADRES = 729, Reflected Power = 1.302 W
retlw      0x6D            ; High Byte, ADRES = 733, Reflected Power = 1.427 W
retlw      0x6E            ; High Byte, ADRES = 737, Reflected Power = 1.564 W
retlw      0x6E            ; High Byte, ADRES = 741, Reflected Power = 1.714 W
```

```
      retlw      0x6F            ; High Byte, ADRES = 745, Reflected Power = 1.879 W
      retlw      0x70            ; High Byte, ADRES = 749, Reflected Power = 2.060 W
      retlw      0x70            ; High Byte, ADRES = 753, Reflected Power = 2.258 W
      retlw      0x70            ; High Byte, ADRES = 757, Reflected Power = 2.475 W
      retlw      0x71            ; High Byte, ADRES = 761, Reflected Power = 2.712 W
      retlw      0x71            ; High Byte, ADRES = 765, Reflected Power = 2.973 W
      retlw      0x72            ; High Byte, ADRES = 769, Reflected Power = 3.259 W
      retlw      0x73            ; High Byte, ADRES = 773, Reflected Power = 3.572 W
      retlw      0x73            ; High Byte, ADRES = 777, Reflected Power = 3.916 W
      retlw      0x74            ; High Byte, ADRES = 781, Reflected Power = 4.292 W
      retlw      0x74            ; High Byte, ADRES = 785, Reflected Power = 4.705 W
      retlw      0x75            ; High Byte, ADRES = 789, Reflected Power = 5.157 W
      retlw      0x75            ; High Byte, ADRES = 793, Reflected Power = 5.653 W
      retlw      0x76            ; High Byte, ADRES = 797, Reflected Power = 6.196 W
      retlw      0x76            ; High Byte, ADRES = 801, Reflected Power = 6.791 W
      retlw      0x77            ; High Byte, ADRES = 805, Reflected Power = 7.444 W
      retlw      0x78            ; High Byte, ADRES = 809, Reflected Power = 8.160 W
      retlw      0x78            ; High Byte, ADRES = 813, Reflected Power = 8.944 W
      retlw      0x78            ; High Byte, ADRES = 817, Reflected Power = 9.804 W
      retlw      0x79            ; High Byte, ADRES = 821, Reflected Power = 10.75 W
      retlw      0x79            ; High Byte, ADRES = 825, Reflected Power = 11.78 W
      retlw      0x7A            ; High Byte, ADRES = 829, Reflected Power = 12.91 W
      retlw      0x7B            ; High Byte, ADRES = 833, Reflected Power = 14.15 W
      retlw      0x7B            ; High Byte, ADRES = 837, Reflected Power = 15.51 W
      retlw      0x7C            ; High Byte, ADRES = 841, Reflected Power = 17.00 W
      retlw      0x7C            ; High Byte, ADRES = 845, Reflected Power = 18.64 W
      retlw      0x7D            ; High Byte, ADRES = 849, Reflected Power = 20.43 W
      retlw      0x7D            ; High Byte, ADRES = 853, Reflected Power = 22.39 W
      retlw      0x7E            ; High Byte, ADRES = 857, Reflected Power = 24.55 W
      retlw      0x7E            ; High Byte, ADRES = 861, Reflected Power = 26.91 W
      retlw      0x7F            ; High Byte, ADRES = 865, Reflected Power = 29.49 W
      retlw      0x80            ; High Byte, ADRES = 869, Reflected Power = 32.33 W
      retlw      0x80            ; High Byte, ADRES = 873, Reflected Power = 35.44 W
      retlw      0x80            ; High Byte, ADRES = 877, Reflected Power = 38.84 W
      retlw      0x81            ; High Byte, ADRES = 881, Reflected Power = 42.58 W
      retlw      0x81            ; High Byte, ADRES = 885, Reflected Power = 46.67 W
      retlw      0x82            ; High Byte, ADRES = 889, Reflected Power = 51.15 W
      retlw      0x83            ; High Byte, ADRES = 893, Reflected Power = 56.07 W
      retlw      0x83            ; High Byte, ADRES = 897, Reflected Power = 61.46 W
      retlw      0x84            ; High Byte, ADRES = 901, Reflected Power = 67.37 W
      retlw      0x84            ; High Byte, ADRES = 905, Reflected Power = 73.85 W
      retlw      0x85            ; High Byte, ADRES = 909, Reflected Power = 80.94 W
      retlw      0x85            ; High Byte, ADRES = 913, Reflected Power = 88.72 W
      retlw      0x86            ; High Byte, ADRES = 917, Reflected Power = 97.25 W
      retlw      0x86            ; High Byte, ADRES = 921, Reflected Power = 106.6 W
      retlw      0x87            ; High Byte, ADRES = 925, Reflected Power = 116.8 W
      retlw      0x88            ; High Byte, ADRES = 929, Reflected Power = 128.1 W
      retlw      0x88            ; High Byte, ADRES = 933, Reflected Power = 140.4 W
      retlw      0x88            ; High Byte, ADRES = 937, Reflected Power = 153.9 W
      retlw      0x89            ; High Byte, ADRES = 941, Reflected Power = 168.7 W
      retlw      0x89            ; High Byte, ADRES = 945, Reflected Power = 184.9 W
      retlw      0x8A            ; High Byte, ADRES = 949, Reflected Power = 202.7 W
      retlw      0x8A            ; High Byte, ADRES = 953, Reflected Power = 222.1 W
      retlw      0x8B            ; High Byte, ADRES = 957, Reflected Power = 243.5 W
      retlw      0x8C            ; High Byte, ADRES = 961, Reflected Power = 266.9 W
      retlw      0x8C            ; High Byte, ADRES = 965, Reflected Power = 292.6 W
      retlw      0x8D            ; High Byte, ADRES = 969, Reflected Power = 320.7 W
      retlw      0x8D            ; High Byte, ADRES = 973, Reflected Power = 351.5 W
      retlw      0x8E            ; High Byte, ADRES = 977, Reflected Power = 385.3 W
      retlw      0x8E            ; High Byte, ADRES = 981, Reflected Power = 422.3 W
      retlw      0x8F            ; High Byte, ADRES = 985, Reflected Power = 462.9 W
      retlw      0x8F            ; High Byte, ADRES = 989, Reflected Power = 507.4 W
      retlw      0x90            ; High Byte, ADRES = 993, Reflected Power = 556.2 W
      retlw      0x90            ; High Byte, ADRES = 997, Reflected Power = 609.7 W
      retlw      0x91            ; High Byte, ADRES = 1001, Reflected Power = 668.3 W
      retlw      0x91            ; High Byte, ADRES = 1005, Reflected Power = 732.5 W
      retlw      0x92            ; High Byte, ADRES = 1009, Reflected Power = 802.9 W
      retlw      0x92            ; High Byte, ADRES = 1013, Reflected Power = 880.1 W
      retlw      0x93            ; High Byte, ADRES = 1017, Reflected Power = 964.7 W
      retlw      0x94            ; High Byte, ADRES = 1021, Reflected Power = 1,057.5 W

RHighByteTable2
      retlw      0x0C            ; High Byte, ADRES = 2, Reflected Power = 74.12 nW
      retlw      0x0D            ; High Byte, ADRES = 6, Reflected Power = 81.24 nW
      retlw      0x0D            ; High Byte, ADRES = 10, Reflected Power = 89.05 nW
      retlw      0x0E            ; High Byte, ADRES = 14, Reflected Power = 97.61 nW
      retlw      0x0F            ; High Byte, ADRES = 18, Reflected Power = 107.0 nW
      retlw      0x0F            ; High Byte, ADRES = 22, Reflected Power = 117.3 nW
      retlw      0x10            ; High Byte, ADRES = 26, Reflected Power = 128.6 nW
      retlw      0x10            ; High Byte, ADRES = 30, Reflected Power = 140.9 nW
      retlw      0x11            ; High Byte, ADRES = 34, Reflected Power = 154.5 nW
```

```
retlw      0x11            ; High Byte, ADRES = 38, Reflected Power = 169.3 nW
retlw      0x12            ; High Byte, ADRES = 42, Reflected Power = 185.6 nW
retlw      0x12            ; High Byte, ADRES = 46, Reflected Power = 203.4 nW
retlw      0x13            ; High Byte, ADRES = 50, Reflected Power = 223.0 nW
retlw      0x14            ; High Byte, ADRES = 54, Reflected Power = 244.4 nW
retlw      0x14            ; High Byte, ADRES = 58, Reflected Power = 267.9 nW
retlw      0x14            ; High Byte, ADRES = 62, Reflected Power = 293.7 nW
retlw      0x15            ; High Byte, ADRES = 66, Reflected Power = 321.9 nW
retlw      0x15            ; High Byte, ADRES = 70, Reflected Power = 352.8 nW
retlw      0x16            ; High Byte, ADRES = 74, Reflected Power = 386.7 nW
retlw      0x17            ; High Byte, ADRES = 78, Reflected Power = 423.9 nW
retlw      0x17            ; High Byte, ADRES = 82, Reflected Power = 464.7 nW
retlw      0x18            ; High Byte, ADRES = 86, Reflected Power = 509.3 nW
retlw      0x18            ; High Byte, ADRES = 90, Reflected Power = 558.3 nW
retlw      0x19            ; High Byte, ADRES = 94, Reflected Power = 611.9 nW
retlw      0x19            ; High Byte, ADRES = 98, Reflected Power = 670.8 nW
retlw      0x1A            ; High Byte, ADRES = 102, Reflected Power = 735.2 nW
retlw      0x1A            ; High Byte, ADRES = 106, Reflected Power = 805.9 nW
retlw      0x1B            ; High Byte, ADRES = 110, Reflected Power = 883.4 nW
retlw      0x1C            ; High Byte, ADRES = 114, Reflected Power = 968.3 nW
retlw      0x1C            ; High Byte, ADRES = 118, Reflected Power = 1.061 uW
retlw      0x1C            ; High Byte, ADRES = 122, Reflected Power = 1.163 uW
retlw      0x1D            ; High Byte, ADRES = 126, Reflected Power = 1.275 uW
retlw      0x1D            ; High Byte, ADRES = 130, Reflected Power = 1.398 uW
retlw      0x1E            ; High Byte, ADRES = 134, Reflected Power = 1.532 uW
retlw      0x1F            ; High Byte, ADRES = 138, Reflected Power = 1.679 uW
retlw      0x1F            ; High Byte, ADRES = 142, Reflected Power = 1.841 uW
retlw      0x20            ; High Byte, ADRES = 146, Reflected Power = 2.018 uW
retlw      0x20            ; High Byte, ADRES = 150, Reflected Power = 2.212 uW
retlw      0x21            ; High Byte, ADRES = 154, Reflected Power = 2.424 uW
retlw      0x21            ; High Byte, ADRES = 158, Reflected Power = 2.657 uW
retlw      0x22            ; High Byte, ADRES = 162, Reflected Power = 2.913 uW
retlw      0x22            ; High Byte, ADRES = 166, Reflected Power = 3.193 uW
retlw      0x23            ; High Byte, ADRES = 170, Reflected Power = 3.500 uW
retlw      0x24            ; High Byte, ADRES = 174, Reflected Power = 3.836 uW
retlw      0x24            ; High Byte, ADRES = 178, Reflected Power = 4.205 uW
retlw      0x24            ; High Byte, ADRES = 182, Reflected Power = 4.609 uW
retlw      0x25            ; High Byte, ADRES = 186, Reflected Power = 5.052 uW
retlw      0x25            ; High Byte, ADRES = 190, Reflected Power = 5.538 uW
retlw      0x26            ; High Byte, ADRES = 194, Reflected Power = 6.070 uW
retlw      0x26            ; High Byte, ADRES = 198, Reflected Power = 6.654 uW
retlw      0x27            ; High Byte, ADRES = 202, Reflected Power = 7.293 uW
retlw      0x28            ; High Byte, ADRES = 206, Reflected Power = 7.994 uW
retlw      0x28            ; High Byte, ADRES = 210, Reflected Power = 8.763 uW
retlw      0x29            ; High Byte, ADRES = 214, Reflected Power = 9.605 uW
retlw      0x29            ; High Byte, ADRES = 218, Reflected Power = 10.53 uW
retlw      0x2A            ; High Byte, ADRES = 222, Reflected Power = 11.54 uW
retlw      0x2A            ; High Byte, ADRES = 226, Reflected Power = 12.65 uW
retlw      0x2B            ; High Byte, ADRES = 230, Reflected Power = 13.87 uW
retlw      0x2B            ; High Byte, ADRES = 234, Reflected Power = 15.20 uW
retlw      0x2C            ; High Byte, ADRES = 238, Reflected Power = 16.66 uW
retlw      0x2C            ; High Byte, ADRES = 242, Reflected Power = 18.26 uW
retlw      0x2D            ; High Byte, ADRES = 246, Reflected Power = 20.02 uW
retlw      0x2D            ; High Byte, ADRES = 250, Reflected Power = 21.94 uW
retlw      0x2E            ; High Byte, ADRES = 254, Reflected Power = 24.05 uW
retlw      0x2E            ; High Byte, ADRES = 258, Reflected Power = 26.36 uW
retlw      0x2F            ; High Byte, ADRES = 262, Reflected Power = 28.90 uW
retlw      0x30            ; High Byte, ADRES = 266, Reflected Power = 31.67 uW
retlw      0x30            ; High Byte, ADRES = 270, Reflected Power = 34.72 uW
retlw      0x30            ; High Byte, ADRES = 274, Reflected Power = 38.05 uW
retlw      0x31            ; High Byte, ADRES = 278, Reflected Power = 41.71 uW
retlw      0x31            ; High Byte, ADRES = 282, Reflected Power = 45.72 uW
retlw      0x32            ; High Byte, ADRES = 286, Reflected Power = 50.12 uW
retlw      0x33            ; High Byte, ADRES = 290, Reflected Power = 54.94 uW
retlw      0x33            ; High Byte, ADRES = 294, Reflected Power = 60.22 uW
retlw      0x34            ; High Byte, ADRES = 298, Reflected Power = 66.00 uW
retlw      0x34            ; High Byte, ADRES = 302, Reflected Power = 72.35 uW
retlw      0x35            ; High Byte, ADRES = 306, Reflected Power = 79.30 uW
retlw      0x35            ; High Byte, ADRES = 310, Reflected Power = 86.93 uW
retlw      0x36            ; High Byte, ADRES = 314, Reflected Power = 95.28 uW
retlw      0x36            ; High Byte, ADRES = 318, Reflected Power = 104.4 uW
retlw      0x37            ; High Byte, ADRES = 322, Reflected Power = 114.5 uW
retlw      0x38            ; High Byte, ADRES = 326, Reflected Power = 125.5 uW
retlw      0x38            ; High Byte, ADRES = 330, Reflected Power = 137.5 uW
retlw      0x38            ; High Byte, ADRES = 334, Reflected Power = 150.8 uW
retlw      0x39            ; High Byte, ADRES = 338, Reflected Power = 165.3 uW
retlw      0x39            ; High Byte, ADRES = 342, Reflected Power = 181.1 uW
retlw      0x3A            ; High Byte, ADRES = 346, Reflected Power = 198.6 uW
retlw      0x3B            ; High Byte, ADRES = 350, Reflected Power = 217.6 uW
retlw      0x3B            ; High Byte, ADRES = 354, Reflected Power = 238.6 uW
retlw      0x3C            ; High Byte, ADRES = 358, Reflected Power = 261.5 uW
```

```
retlw      0x3C            ; High Byte, ADRES = 362, Reflected Power = 286.6 uW
retlw      0x3D            ; High Byte, ADRES = 366, Reflected Power = 314.2 uW
retlw      0x3D            ; High Byte, ADRES = 370, Reflected Power = 344.4 uW
retlw      0x3E            ; High Byte, ADRES = 374, Reflected Power = 377.5 uW
retlw      0x3E            ; High Byte, ADRES = 378, Reflected Power = 413.8 uW
retlw      0x3F            ; High Byte, ADRES = 382, Reflected Power = 453.6 uW
retlw      0x40            ; High Byte, ADRES = 386, Reflected Power = 497.2 uW
retlw      0x40            ; High Byte, ADRES = 390, Reflected Power = 544.9 uW
retlw      0x40            ; High Byte, ADRES = 394, Reflected Power = 597.3 uW
retlw      0x41            ; High Byte, ADRES = 398, Reflected Power = 654.7 uW
retlw      0x41            ; High Byte, ADRES = 402, Reflected Power = 717.7 uW
retlw      0x42            ; High Byte, ADRES = 406, Reflected Power = 786.7 uW
retlw      0x43            ; High Byte, ADRES = 410, Reflected Power = 862.3 uW
retlw      0x43            ; High Byte, ADRES = 414, Reflected Power = 945.2 uW
retlw      0x44            ; High Byte, ADRES = 418, Reflected Power = 1.036 mW
retlw      0x44            ; High Byte, ADRES = 422, Reflected Power = 1.136 mW
retlw      0x45            ; High Byte, ADRES = 426, Reflected Power = 1.245 mW
retlw      0x45            ; High Byte, ADRES = 430, Reflected Power = 1.364 mW
retlw      0x46            ; High Byte, ADRES = 434, Reflected Power = 1.496 mW
retlw      0x46            ; High Byte, ADRES = 438, Reflected Power = 1.639 mW
retlw      0x47            ; High Byte, ADRES = 442, Reflected Power = 1.797 mW
retlw      0x48            ; High Byte, ADRES = 446, Reflected Power = 1.970 mW
retlw      0x48            ; High Byte, ADRES = 450, Reflected Power = 2.159 mW
retlw      0x48            ; High Byte, ADRES = 454, Reflected Power = 2.367 mW
retlw      0x49            ; High Byte, ADRES = 458, Reflected Power = 2.594 mW
retlw      0x49            ; High Byte, ADRES = 462, Reflected Power = 2.843 mW
retlw      0x4A            ; High Byte, ADRES = 466, Reflected Power = 3.117 mW
retlw      0x4A            ; High Byte, ADRES = 470, Reflected Power = 3.416 mW
retlw      0x4B            ; High Byte, ADRES = 474, Reflected Power = 3.745 mW
retlw      0x4C            ; High Byte, ADRES = 478, Reflected Power = 4.105 mW
retlw      0x4C            ; High Byte, ADRES = 482, Reflected Power = 4.499 mW
retlw      0x4D            ; High Byte, ADRES = 486, Reflected Power = 4.932 mW
retlw      0x4D            ; High Byte, ADRES = 490, Reflected Power = 5.406 mW
retlw      0x4E            ; High Byte, ADRES = 494, Reflected Power = 5.925 mW
retlw      0x4E            ; High Byte, ADRES = 498, Reflected Power = 6.495 mW
retlw      0x4F            ; High Byte, ADRES = 502, Reflected Power = 7.119 mW
retlw      0x4F            ; High Byte, ADRES = 506, Reflected Power = 7.804 mW
retlw      0x50            ; High Byte, ADRES = 510, Reflected Power = 8.554 mW
retlw      0x50            ; High Byte, ADRES = 514, Reflected Power = 9.376 mW
retlw      0x51            ; High Byte, ADRES = 518, Reflected Power = 10.28 mW
retlw      0x51            ; High Byte, ADRES = 522, Reflected Power = 11.26 mW
retlw      0x52            ; High Byte, ADRES = 526, Reflected Power = 12.35 mW
retlw      0x52            ; High Byte, ADRES = 530, Reflected Power = 13.53 mW
retlw      0x53            ; High Byte, ADRES = 534, Reflected Power = 14.84 mW
retlw      0x54            ; High Byte, ADRES = 538, Reflected Power = 16.26 mW
retlw      0x54            ; High Byte, ADRES = 542, Reflected Power = 17.83 mW
retlw      0x55            ; High Byte, ADRES = 546, Reflected Power = 19.54 mW
retlw      0x55            ; High Byte, ADRES = 550, Reflected Power = 21.42 mW
retlw      0x56            ; High Byte, ADRES = 554, Reflected Power = 23.48 mW
retlw      0x56            ; High Byte, ADRES = 558, Reflected Power = 25.73 mW
retlw      0x57            ; High Byte, ADRES = 562, Reflected Power = 28.21 mW
retlw      0x57            ; High Byte, ADRES = 566, Reflected Power = 30.92 mW
retlw      0x58            ; High Byte, ADRES = 570, Reflected Power = 33.89 mW
retlw      0x58            ; High Byte, ADRES = 574, Reflected Power = 37.15 mW
retlw      0x59            ; High Byte, ADRES = 578, Reflected Power = 40.72 mW
retlw      0x59            ; High Byte, ADRES = 582, Reflected Power = 44.63 mW
retlw      0x5A            ; High Byte, ADRES = 586, Reflected Power = 48.92 mW
retlw      0x5A            ; High Byte, ADRES = 590, Reflected Power = 53.62 mW
retlw      0x5B            ; High Byte, ADRES = 594, Reflected Power = 58.78 mW
retlw      0x5C            ; High Byte, ADRES = 598, Reflected Power = 64.43 mW
retlw      0x5C            ; High Byte, ADRES = 602, Reflected Power = 70.62 mW
retlw      0x5C            ; High Byte, ADRES = 606, Reflected Power = 77.41 mW
retlw      0x5D            ; High Byte, ADRES = 610, Reflected Power = 84.85 mW
retlw      0x5D            ; High Byte, ADRES = 614, Reflected Power = 93.01 mW
retlw      0x5E            ; High Byte, ADRES = 618, Reflected Power = 101.9 mW
retlw      0x5F            ; High Byte, ADRES = 622, Reflected Power = 111.7 mW
retlw      0x5F            ; High Byte, ADRES = 626, Reflected Power = 122.5 mW
retlw      0x60            ; High Byte, ADRES = 630, Reflected Power = 134.3 mW
retlw      0x60            ; High Byte, ADRES = 634, Reflected Power = 147.2 mW
retlw      0x61            ; High Byte, ADRES = 638, Reflected Power = 161.3 mW
retlw      0x61            ; High Byte, ADRES = 642, Reflected Power = 176.8 mW
retlw      0x62            ; High Byte, ADRES = 646, Reflected Power = 193.8 mW
retlw      0x62            ; High Byte, ADRES = 650, Reflected Power = 212.4 mW
retlw      0x63            ; High Byte, ADRES = 654, Reflected Power = 232.9 mW
retlw      0x64            ; High Byte, ADRES = 658, Reflected Power = 255.3 mW
retlw      0x64            ; High Byte, ADRES = 662, Reflected Power = 279.8 mW
retlw      0x64            ; High Byte, ADRES = 666, Reflected Power = 306.7 mW
retlw      0x65            ; High Byte, ADRES = 670, Reflected Power = 336.2 mW
retlw      0x65            ; High Byte, ADRES = 674, Reflected Power = 368.5 mW
retlw      0x66            ; High Byte, ADRES = 678, Reflected Power = 403.9 mW
retlw      0x67            ; High Byte, ADRES = 682, Reflected Power = 442.7 mW
```

```
retlw      0x67            ; High Byte, ADRES = 686, Reflected Power = 485.3 mW
retlw      0x68            ; High Byte, ADRES = 690, Reflected Power = 531.9 mW
retlw      0x68            ; High Byte, ADRES = 694, Reflected Power = 583.1 mW
retlw      0x69            ; High Byte, ADRES = 698, Reflected Power = 639.1 mW
retlw      0x69            ; High Byte, ADRES = 702, Reflected Power = 700.5 mW
retlw      0x6A            ; High Byte, ADRES = 706, Reflected Power = 767.9 mW
retlw      0x6A            ; High Byte, ADRES = 710, Reflected Power = 841.7 mW
retlw      0x6B            ; High Byte, ADRES = 714, Reflected Power = 922.6 mW
retlw      0x6C            ; High Byte, ADRES = 718, Reflected Power = 1.011 W
retlw      0x6C            ; High Byte, ADRES = 722, Reflected Power = 1.108 W
retlw      0x6C            ; High Byte, ADRES = 726, Reflected Power = 1.215 W
retlw      0x6D            ; High Byte, ADRES = 730, Reflected Power = 1.332 W
retlw      0x6D            ; High Byte, ADRES = 734, Reflected Power = 1.460 W
retlw      0x6E            ; High Byte, ADRES = 738, Reflected Power = 1.600 W
retlw      0x6F            ; High Byte, ADRES = 742, Reflected Power = 1.754 W
retlw      0x6F            ; High Byte, ADRES = 746, Reflected Power = 1.923 W
retlw      0x70            ; High Byte, ADRES = 750, Reflected Power = 2.107 W
retlw      0x70            ; High Byte, ADRES = 754, Reflected Power = 2.310 W
retlw      0x71            ; High Byte, ADRES = 758, Reflected Power = 2.532 W
retlw      0x71            ; High Byte, ADRES = 762, Reflected Power = 2.775 W
retlw      0x72            ; High Byte, ADRES = 766, Reflected Power = 3.042 W
retlw      0x72            ; High Byte, ADRES = 770, Reflected Power = 3.335 W
retlw      0x73            ; High Byte, ADRES = 774, Reflected Power = 3.655 W
retlw      0x74            ; High Byte, ADRES = 778, Reflected Power = 4.007 W
retlw      0x74            ; High Byte, ADRES = 782, Reflected Power = 4.392 W
retlw      0x74            ; High Byte, ADRES = 786, Reflected Power = 4.814 W
retlw      0x75            ; High Byte, ADRES = 790, Reflected Power = 5.277 W
retlw      0x75            ; High Byte, ADRES = 794, Reflected Power = 5.784 W
retlw      0x76            ; High Byte, ADRES = 798, Reflected Power = 6.340 W
retlw      0x76            ; High Byte, ADRES = 802, Reflected Power = 6.949 W
retlw      0x77            ; High Byte, ADRES = 806, Reflected Power = 7.617 W
retlw      0x78            ; High Byte, ADRES = 810, Reflected Power = 8.349 W
retlw      0x78            ; High Byte, ADRES = 814, Reflected Power = 9.152 W
retlw      0x79            ; High Byte, ADRES = 818, Reflected Power = 10.03 W
retlw      0x79            ; High Byte, ADRES = 822, Reflected Power = 11.00 W
retlw      0x7A            ; High Byte, ADRES = 826, Reflected Power = 12.05 W
retlw      0x7A            ; High Byte, ADRES = 830, Reflected Power = 13.21 W
retlw      0x7B            ; High Byte, ADRES = 834, Reflected Power = 14.48 W
retlw      0x7B            ; High Byte, ADRES = 838, Reflected Power = 15.87 W
retlw      0x7C            ; High Byte, ADRES = 842, Reflected Power = 17.40 W
retlw      0x7C            ; High Byte, ADRES = 846, Reflected Power = 19.07 W
retlw      0x7D            ; High Byte, ADRES = 850, Reflected Power = 20.90 W
retlw      0x7D            ; High Byte, ADRES = 854, Reflected Power = 22.91 W
retlw      0x7E            ; High Byte, ADRES = 858, Reflected Power = 25.12 W
retlw      0x7E            ; High Byte, ADRES = 862, Reflected Power = 27.53 W
retlw      0x7F            ; High Byte, ADRES = 866, Reflected Power = 30.18 W
retlw      0x80            ; High Byte, ADRES = 870, Reflected Power = 33.08 W
retlw      0x80            ; High Byte, ADRES = 874, Reflected Power = 36.26 W
retlw      0x80            ; High Byte, ADRES = 878, Reflected Power = 39.74 W
retlw      0x81            ; High Byte, ADRES = 882, Reflected Power = 43.56 W
retlw      0x81            ; High Byte, ADRES = 886, Reflected Power = 47.75 W
retlw      0x82            ; High Byte, ADRES = 890, Reflected Power = 52.34 W
retlw      0x83            ; High Byte, ADRES = 894, Reflected Power = 57.37 W
retlw      0x83            ; High Byte, ADRES = 898, Reflected Power = 62.89 W
retlw      0x84            ; High Byte, ADRES = 902, Reflected Power = 68.93 W
retlw      0x84            ; High Byte, ADRES = 906, Reflected Power = 75.56 W
retlw      0x85            ; High Byte, ADRES = 910, Reflected Power = 82.82 W
retlw      0x85            ; High Byte, ADRES = 914, Reflected Power = 90.78 W
retlw      0x86            ; High Byte, ADRES = 918, Reflected Power = 99.51 W
retlw      0x86            ; High Byte, ADRES = 922, Reflected Power = 109.1 W
retlw      0x87            ; High Byte, ADRES = 926, Reflected Power = 119.6 W
retlw      0x88            ; High Byte, ADRES = 930, Reflected Power = 131.1 W
retlw      0x88            ; High Byte, ADRES = 934, Reflected Power = 143.7 W
retlw      0x88            ; High Byte, ADRES = 938, Reflected Power = 157.5 W
retlw      0x89            ; High Byte, ADRES = 942, Reflected Power = 172.6 W
retlw      0x89            ; High Byte, ADRES = 946, Reflected Power = 189.2 W
retlw      0x8A            ; High Byte, ADRES = 950, Reflected Power = 207.4 W
retlw      0x8B            ; High Byte, ADRES = 954, Reflected Power = 227.3 W
retlw      0x8B            ; High Byte, ADRES = 958, Reflected Power = 249.2 W
retlw      0x8C            ; High Byte, ADRES = 962, Reflected Power = 273.1 W
retlw      0x8C            ; High Byte, ADRES = 966, Reflected Power = 299.4 W
retlw      0x8D            ; High Byte, ADRES = 970, Reflected Power = 328.1 W
retlw      0x8D            ; High Byte, ADRES = 974, Reflected Power = 359.7 W
retlw      0x8E            ; High Byte, ADRES = 978, Reflected Power = 394.2 W
retlw      0x8E            ; High Byte, ADRES = 982, Reflected Power = 432.1 W
retlw      0x8F            ; High Byte, ADRES = 986, Reflected Power = 473.7 W
retlw      0x90            ; High Byte, ADRES = 990, Reflected Power = 519.2 W
retlw      0x90            ; High Byte, ADRES = 994, Reflected Power = 569.1 W
retlw      0x90            ; High Byte, ADRES = 998, Reflected Power = 623.8 W
retlw      0x91            ; High Byte, ADRES = 1002, Reflected Power = 683.8 W
retlw      0x91            ; High Byte, ADRES = 1006, Reflected Power = 749.5 W
```

```
      retlw      0x92            ; High Byte, ADRES = 1010, Reflected Power = 821.6 W
      retlw      0x93            ; High Byte, ADRES = 1014, Reflected Power = 900.6 W
      retlw      0x93            ; High Byte, ADRES = 1018, Reflected Power = 987.1 W
      retlw      0x94            ; High Byte, ADRES = 1022, Reflected Power = 1,082.0 W

RHighByteTable3
      retlw      0x0D            ; High Byte, ADRES = 3, Reflected Power = 75.84 nW
      retlw      0x0D            ; High Byte, ADRES = 7, Reflected Power = 83.13 nW
      retlw      0x0E            ; High Byte, ADRES = 11, Reflected Power = 91.12 nW
      retlw      0x0E            ; High Byte, ADRES = 15, Reflected Power = 99.88 nW
      retlw      0x0F            ; High Byte, ADRES = 19, Reflected Power = 109.5 nW
      retlw      0x10            ; High Byte, ADRES = 23, Reflected Power = 120.0 nW
      retlw      0x10            ; High Byte, ADRES = 27, Reflected Power = 131.5 nW
      retlw      0x10            ; High Byte, ADRES = 31, Reflected Power = 144.2 nW
      retlw      0x11            ; High Byte, ADRES = 35, Reflected Power = 158.0 nW
      retlw      0x11            ; High Byte, ADRES = 39, Reflected Power = 173.2 nW
      retlw      0x12            ; High Byte, ADRES = 43, Reflected Power = 189.9 nW
      retlw      0x12            ; High Byte, ADRES = 47, Reflected Power = 208.1 nW
      retlw      0x13            ; High Byte, ADRES = 51, Reflected Power = 228.1 nW
      retlw      0x14            ; High Byte, ADRES = 55, Reflected Power = 250.1 nW
      retlw      0x14            ; High Byte, ADRES = 59, Reflected Power = 274.1 nW
      retlw      0x15            ; High Byte, ADRES = 63, Reflected Power = 300.5 nW
      retlw      0x15            ; High Byte, ADRES = 67, Reflected Power = 329.3 nW
      retlw      0x16            ; High Byte, ADRES = 71, Reflected Power = 361.0 nW
      retlw      0x16            ; High Byte, ADRES = 75, Reflected Power = 395.7 nW
      retlw      0x17            ; High Byte, ADRES = 79, Reflected Power = 433.7 nW
      retlw      0x17            ; High Byte, ADRES = 83, Reflected Power = 475.4 nW
      retlw      0x18            ; High Byte, ADRES = 87, Reflected Power = 521.1 nW
      retlw      0x18            ; High Byte, ADRES = 91, Reflected Power = 571.2 nW
      retlw      0x19            ; High Byte, ADRES = 95, Reflected Power = 626.1 nW
      retlw      0x19            ; High Byte, ADRES = 99, Reflected Power = 686.3 nW
      retlw      0x1A            ; High Byte, ADRES = 103, Reflected Power = 752.3 nW
      retlw      0x1A            ; High Byte, ADRES = 107, Reflected Power = 824.6 nW
      retlw      0x1B            ; High Byte, ADRES = 111, Reflected Power = 903.9 nW
      retlw      0x1C            ; High Byte, ADRES = 115, Reflected Power = 990.8 nW
      retlw      0x1C            ; High Byte, ADRES = 119, Reflected Power = 1.086 uW
      retlw      0x1C            ; High Byte, ADRES = 123, Reflected Power = 1.190 uW
      retlw      0x1D            ; High Byte, ADRES = 127, Reflected Power = 1.305 uW
      retlw      0x1E            ; High Byte, ADRES = 131, Reflected Power = 1.430 uW
      retlw      0x1E            ; High Byte, ADRES = 135, Reflected Power = 1.568 uW
      retlw      0x1F            ; High Byte, ADRES = 139, Reflected Power = 1.718 uW
      retlw      0x1F            ; High Byte, ADRES = 143, Reflected Power = 1.884 uW
      retlw      0x20            ; High Byte, ADRES = 147, Reflected Power = 2.065 uW
      retlw      0x20            ; High Byte, ADRES = 151, Reflected Power = 2.263 uW
      retlw      0x21            ; High Byte, ADRES = 155, Reflected Power = 2.481 uW
      retlw      0x21            ; High Byte, ADRES = 159, Reflected Power = 2.719 uW
      retlw      0x22            ; High Byte, ADRES = 163, Reflected Power = 2.981 uW
      retlw      0x22            ; High Byte, ADRES = 167, Reflected Power = 3.267 uW
      retlw      0x23            ; High Byte, ADRES = 171, Reflected Power = 3.581 uW
      retlw      0x24            ; High Byte, ADRES = 175, Reflected Power = 3.925 uW
      retlw      0x24            ; High Byte, ADRES = 179, Reflected Power = 4.303 uW
      retlw      0x24            ; High Byte, ADRES = 183, Reflected Power = 4.716 uW
      retlw      0x25            ; High Byte, ADRES = 187, Reflected Power = 5.170 uW
      retlw      0x25            ; High Byte, ADRES = 191, Reflected Power = 5.667 uW
      retlw      0x26            ; High Byte, ADRES = 195, Reflected Power = 6.211 uW
      retlw      0x27            ; High Byte, ADRES = 199, Reflected Power = 6.808 uW
      retlw      0x27            ; High Byte, ADRES = 203, Reflected Power = 7.463 uW
      retlw      0x28            ; High Byte, ADRES = 207, Reflected Power = 8.180 uW
      retlw      0x28            ; High Byte, ADRES = 211, Reflected Power = 8.966 uW
      retlw      0x29            ; High Byte, ADRES = 215, Reflected Power = 9.828 uW
      retlw      0x29            ; High Byte, ADRES = 219, Reflected Power = 10.77 uW
      retlw      0x2A            ; High Byte, ADRES = 223, Reflected Power = 11.81 uW
      retlw      0x2A            ; High Byte, ADRES = 227, Reflected Power = 12.94 uW
      retlw      0x2B            ; High Byte, ADRES = 231, Reflected Power = 14.19 uW
      retlw      0x2C            ; High Byte, ADRES = 235, Reflected Power = 15.55 uW
      retlw      0x2C            ; High Byte, ADRES = 239, Reflected Power = 17.05 uW
      retlw      0x2C            ; High Byte, ADRES = 243, Reflected Power = 18.69 uW
      retlw      0x2D            ; High Byte, ADRES = 247, Reflected Power = 20.48 uW
      retlw      0x2D            ; High Byte, ADRES = 251, Reflected Power = 22.45 uW
      retlw      0x2E            ; High Byte, ADRES = 255, Reflected Power = 24.61 uW
      retlw      0x2F            ; High Byte, ADRES = 259, Reflected Power = 26.97 uW
      retlw      0x2F            ; High Byte, ADRES = 263, Reflected Power = 29.57 uW
      retlw      0x30            ; High Byte, ADRES = 267, Reflected Power = 32.41 uW
      retlw      0x30            ; High Byte, ADRES = 271, Reflected Power = 35.52 uW
      retlw      0x31            ; High Byte, ADRES = 275, Reflected Power = 38.94 uW
      retlw      0x31            ; High Byte, ADRES = 279, Reflected Power = 42.68 uW
      retlw      0x32            ; High Byte, ADRES = 283, Reflected Power = 46.78 uW
      retlw      0x32            ; High Byte, ADRES = 287, Reflected Power = 51.28 uW
      retlw      0x33            ; High Byte, ADRES = 291, Reflected Power = 56.21 uW
      retlw      0x34            ; High Byte, ADRES = 295, Reflected Power = 61.61 uW
      retlw      0x34            ; High Byte, ADRES = 299, Reflected Power = 67.54 uW
```

```
retlw      0x34            ; High Byte, ADRES = 303, Reflected Power = 74.03 uW
retlw      0x35            ; High Byte, ADRES = 307, Reflected Power = 81.14 uW
retlw      0x35            ; High Byte, ADRES = 311, Reflected Power = 88.94 uW
retlw      0x36            ; High Byte, ADRES = 315, Reflected Power = 97.49 uW
retlw      0x37            ; High Byte, ADRES = 319, Reflected Power = 106.9 uW
retlw      0x37            ; High Byte, ADRES = 323, Reflected Power = 117.1 uW
retlw      0x38            ; High Byte, ADRES = 327, Reflected Power = 128.4 uW
retlw      0x38            ; High Byte, ADRES = 331, Reflected Power = 140.7 uW
retlw      0x39            ; High Byte, ADRES = 335, Reflected Power = 154.3 uW
retlw      0x39            ; High Byte, ADRES = 339, Reflected Power = 169.1 uW
retlw      0x3A            ; High Byte, ADRES = 343, Reflected Power = 185.4 uW
retlw      0x3A            ; High Byte, ADRES = 347, Reflected Power = 203.2 uW
retlw      0x3B            ; High Byte, ADRES = 351, Reflected Power = 222.7 uW
retlw      0x3E            ; High Byte, ADRES = 355, Reflected Power = 244.1 uW
retlw      0x3C            ; High Byte, ADRES = 359, Reflected Power = 267.6 uW
retlw      0x3C            ; High Byte, ADRES = 363, Reflected Power = 293.3 uW
retlw      0x3D            ; High Byte, ADRES = 367, Reflected Power = 321.5 uW
retlw      0x3D            ; High Byte, ADRES = 371, Reflected Power = 352.4 uW
retlw      0x3E            ; High Byte, ADRES = 375, Reflected Power = 386.3 uW
retlw      0x3E            ; High Byte, ADRES = 379, Reflected Power = 423.4 uW
retlw      0x3F            ; High Byte, ADRES = 383, Reflected Power = 464.1 uW
retlw      0x40            ; High Byte, ADRES = 387, Reflected Power = 508.7 uW
retlw      0x40            ; High Byte, ADRES = 391, Reflected Power = 557.6 uW
retlw      0x41            ; High Byte, ADRES = 395, Reflected Power = 611.2 uW
retlw      0x41            ; High Byte, ADRES = 399, Reflected Power = 669.9 uW
retlw      0x42            ; High Byte, ADRES = 403, Reflected Power = 734.3 uW
retlw      0x42            ; High Byte, ADRES = 407, Reflected Power = 804.9 uW
retlw      0x43            ; High Byte, ADRES = 411, Reflected Power = 882.3 uW
retlw      0x43            ; High Byte, ADRES = 415, Reflected Power = 967.1 uW
retlw      0x44            ; High Byte, ADRES = 419, Reflected Power = 1.060 mW
retlw      0x44            ; High Byte, ADRES = 423, Reflected Power = 1.162 mW
retlw      0x45            ; High Byte, ADRES = 427, Reflected Power = 1.274 mW
retlw      0x45            ; High Byte, ADRES = 431, Reflected Power = 1.396 mW
retlw      0x46            ; High Byte, ADRES = 435, Reflected Power = 1.530 mW
retlw      0x46            ; High Byte, ADRES = 439, Reflected Power = 1.677 mW
retlw      0x47            ; High Byte, ADRES = 443, Reflected Power = 1.839 mW
retlw      0x48            ; High Byte, ADRES = 447, Reflected Power = 2.015 mW
retlw      0x48            ; High Byte, ADRES = 451, Reflected Power = 2.209 mW
retlw      0x48            ; High Byte, ADRES = 455, Reflected Power = 2.421 mW
retlw      0x49            ; High Byte, ADRES = 459, Reflected Power = 2.654 mW
retlw      0x49            ; High Byte, ADRES = 463, Reflected Power = 2.909 mW
retlw      0x4A            ; High Byte, ADRES = 467, Reflected Power = 3.189 mW
retlw      0x4B            ; High Byte, ADRES = 471, Reflected Power = 3.496 mW
retlw      0x4B            ; High Byte, ADRES = 475, Reflected Power = 3.832 mW
retlw      0x4C            ; High Byte, ADRES = 479, Reflected Power = 4.200 mW
retlw      0x4C            ; High Byte, ADRES = 483, Reflected Power = 4.604 mW
retlw      0x4D            ; High Byte, ADRES = 487, Reflected Power = 5.046 mW
retlw      0x4D            ; High Byte, ADRES = 491, Reflected Power = 5.531 mW
retlw      0x4E            ; High Byte, ADRES = 495, Reflected Power = 6.063 mW
retlw      0x4E            ; High Byte, ADRES = 499, Reflected Power = 6.646 mW
retlw      0x4F            ; High Byte, ADRES = 503, Reflected Power = 7.284 mW
retlw      0x50            ; High Byte, ADRES = 507, Reflected Power = 7.985 mW
retlw      0x50            ; High Byte, ADRES = 511, Reflected Power = 8.752 mW
retlw      0x50            ; High Byte, ADRES = 515, Reflected Power = 9.593 mW
retlw      0x51            ; High Byte, ADRES = 519, Reflected Power = 10.52 mW
retlw      0x51            ; High Byte, ADRES = 523, Reflected Power = 11.53 mW
retlw      0x52            ; High Byte, ADRES = 527, Reflected Power = 12.63 mW
retlw      0x53            ; High Byte, ADRES = 531, Reflected Power = 13.85 mW
retlw      0x53            ; High Byte, ADRES = 535, Reflected Power = 15.18 mW
retlw      0x54            ; High Byte, ADRES = 539, Reflected Power = 16.64 mW
retlw      0x54            ; High Byte, ADRES = 543, Reflected Power = 18.24 mW
retlw      0x55            ; High Byte, ADRES = 547, Reflected Power = 19.99 mW
retlw      0x55            ; High Byte, ADRES = 551, Reflected Power = 21.91 mW
retlw      0x56            ; High Byte, ADRES = 555, Reflected Power = 24.02 mW
retlw      0x56            ; High Byte, ADRES = 559, Reflected Power = 26.33 mW
retlw      0x57            ; High Byte, ADRES = 563, Reflected Power = 28.86 mW
retlw      0x58            ; High Byte, ADRES = 567, Reflected Power = 31.63 mW
retlw      0x58            ; High Byte, ADRES = 571, Reflected Power = 34.67 mW
retlw      0x58            ; High Byte, ADRES = 575, Reflected Power = 38.01 mW
retlw      0x59            ; High Byte, ADRES = 579, Reflected Power = 41.66 mW
retlw      0x59            ; High Byte, ADRES = 583, Reflected Power = 45.67 mW
retlw      0x5A            ; High Byte, ADRES = 587, Reflected Power = 50.06 mW
retlw      0x5B            ; High Byte, ADRES = 591, Reflected Power = 54.87 mW
retlw      0x5B            ; High Byte, ADRES = 595, Reflected Power = 60.14 mW
retlw      0x5C            ; High Byte, ADRES = 599, Reflected Power = 65.92 mW
retlw      0x5C            ; High Byte, ADRES = 603, Reflected Power = 72.26 mW
retlw      0x5D            ; High Byte, ADRES = 607, Reflected Power = 79.21 mW
retlw      0x5D            ; High Byte, ADRES = 611, Reflected Power = 86.82 mW
retlw      0x5E            ; High Byte, ADRES = 615, Reflected Power = 95.16 mW
retlw      0x5E            ; High Byte, ADRES = 619, Reflected Power = 104.3 mW
retlw      0x5F            ; High Byte, ADRES = 623, Reflected Power = 114.3 mW
```

```
retlw      0x60            ; High Byte, ADRES = 627, Reflected Power = 125.3 mW
retlw      0x60            ; High Byte, ADRES = 631, Reflected Power = 137.4 mW
retlw      0x60            ; High Byte, ADRES = 635, Reflected Power = 150.6 mW
retlw      0x61            ; High Byte, ADRES = 639, Reflected Power = 165.1 mW
retlw      0x61            ; High Byte, ADRES = 643, Reflected Power = 180.9 mW
retlw      0x62            ; High Byte, ADRES = 647, Reflected Power = 198.3 mW
retlw      0x62            ; High Byte, ADRES = 651, Reflected Power = 217.4 mW
retlw      0x63            ; High Byte, ADRES = 655, Reflected Power = 238.3 mW
retlw      0x64            ; High Byte, ADRES = 659, Reflected Power = 261.2 mW
retlw      0x64            ; High Byte, ADRES = 663, Reflected Power = 286.3 mW
retlw      0x65            ; High Byte, ADRES = 667, Reflected Power = 313.8 mW
retlw      0x65            ; High Byte, ADRES = 671, Reflected Power = 344.0 mW
retlw      0x66            ; High Byte, ADRES = 675, Reflected Power = 377.0 mW
retlw      0x66            ; High Byte, ADRES = 679, Reflected Power = 413.3 mW
retlw      0x67            ; High Byte, ADRES = 683, Reflected Power = 453.0 mW
retlw      0x67            ; High Byte, ADRES = 687, Reflected Power = 496.5 mW
retlw      0x68            ; High Byte, ADRES = 691, Reflected Power = 544.3 mW
retlw      0x68            ; High Byte, ADRES = 695, Reflected Power = 596.6 mW
retlw      0x69            ; High Byte, ADRES = 699, Reflected Power = 653.9 mW
retlw      0x69            ; High Byte, ADRES = 703, Reflected Power = 716.8 mW
retlw      0x6A            ; High Byte, ADRES = 707, Reflected Power = 785.7 mW
retlw      0x6A            ; High Byte, ADRES = 711, Reflected Power = 861.2 mW
retlw      0x6B            ; High Byte, ADRES = 715, Reflected Power = 944.0 mW
retlw      0x6C            ; High Byte, ADRES = 719, Reflected Power = 1.035 W
retlw      0x6C            ; High Byte, ADRES = 723, Reflected Power = 1.134 W
retlw      0x6C            ; High Byte, ADRES = 727, Reflected Power = 1.243 W
retlw      0x6D            ; High Byte, ADRES = 731, Reflected Power = 1.363 W
retlw      0x6D            ; High Byte, ADRES = 735, Reflected Power = 1.494 W
retlw      0x6E            ; High Byte, ADRES = 739, Reflected Power = 1.637 W
retlw      0x6F            ; High Byte, ADRES = 743, Reflected Power = 1.795 W
retlw      0x6F            ; High Byte, ADRES = 747, Reflected Power = 1.967 W
retlw      0x70            ; High Byte, ADRES = 751, Reflected Power = 2.156 W
retlw      0x70            ; High Byte, ADRES = 755, Reflected Power = 2.364 W
retlw      0x71            ; High Byte, ADRES = 759, Reflected Power = 2.591 W
retlw      0x71            ; High Byte, ADRES = 763, Reflected Power = 2.840 W
retlw      0x72            ; High Byte, ADRES = 767, Reflected Power = 3.113 W
retlw      0x72            ; High Byte, ADRES = 771, Reflected Power = 3.412 W
retlw      0x73            ; High Byte, ADRES = 775, Reflected Power = 3.740 W
retlw      0x74            ; High Byte, ADRES = 779, Reflected Power = 4.100 W
retlw      0x74            ; High Byte, ADRES = 783, Reflected Power = 4.494 W
retlw      0x74            ; High Byte, ADRES = 787, Reflected Power = 4.926 W
retlw      0x75            ; High Byte, ADRES = 791, Reflected Power = 5.399 W
retlw      0x75            ; High Byte, ADRES = 795, Reflected Power = 5.918 W
retlw      0x76            ; High Byte, ADRES = 799, Reflected Power = 6.487 W
retlw      0x77            ; High Byte, ADRES = 803, Reflected Power = 7.110 W
retlw      0x77            ; High Byte, ADRES = 807, Reflected Power = 7.794 W
retlw      0x78            ; High Byte, ADRES = 811, Reflected Power = 8.543 W
retlw      0x78            ; High Byte, ADRES = 815, Reflected Power = 9.364 W
retlw      0x79            ; High Byte, ADRES = 819, Reflected Power = 10.26 W
retlw      0x79            ; High Byte, ADRES = 823, Reflected Power = 11.25 W
retlw      0x7A            ; High Byte, ADRES = 827, Reflected Power = 12.33 W
retlw      0x7A            ; High Byte, ADRES = 831, Reflected Power = 13.52 W
retlw      0x7B            ; High Byte, ADRES = 835, Reflected Power = 14.82 W
retlw      0x7C            ; High Byte, ADRES = 839, Reflected Power = 16.24 W
retlw      0x7C            ; High Byte, ADRES = 843, Reflected Power = 17.80 W
retlw      0x7C            ; High Byte, ADRES = 847, Reflected Power = 19.51 W
retlw      0x7D            ; High Byte, ADRES = 851, Reflected Power = 21.39 W
retlw      0x7D            ; High Byte, ADRES = 855, Reflected Power = 23.45 W
retlw      0x7E            ; High Byte, ADRES = 859, Reflected Power = 25.70 W
retlw      0x7F            ; High Byte, ADRES = 863, Reflected Power = 28.17 W
retlw      0x7F            ; High Byte, ADRES = 867, Reflected Power = 30.88 W
retlw      0x80            ; High Byte, ADRES = 871, Reflected Power = 33.85 W
retlw      0x80            ; High Byte, ADRES = 875, Reflected Power = 37.10 W
retlw      0x81            ; High Byte, ADRES = 879, Reflected Power = 40.67 W
retlw      0x81            ; High Byte, ADRES = 883, Reflected Power = 44.58 W
retlw      0x82            ; High Byte, ADRES = 887, Reflected Power = 48.86 W
retlw      0x82            ; High Byte, ADRES = 891, Reflected Power = 53.56 W
retlw      0x83            ; High Byte, ADRES = 895, Reflected Power = 58.70 W
retlw      0x84            ; High Byte, ADRES = 899, Reflected Power = 64.35 W
retlw      0x84            ; High Byte, ADRES = 903, Reflected Power = 70.53 W
retlw      0x84            ; High Byte, ADRES = 907, Reflected Power = 77.31 W
retlw      0x85            ; High Byte, ADRES = 911, Reflected Power = 84.75 W
retlw      0x85            ; High Byte, ADRES = 915, Reflected Power = 92.89 W
retlw      0x86            ; High Byte, ADRES = 919, Reflected Power = 101.8 W
retlw      0x86            ; High Byte, ADRES = 923, Reflected Power = 111.6 W
retlw      0x87            ; High Byte, ADRES = 927, Reflected Power = 122.3 W
retlw      0x88            ; High Byte, ADRES = 931, Reflected Power = 134.1 W
retlw      0x88            ; High Byte, ADRES = 935, Reflected Power = 147.0 W
retlw      0x89            ; High Byte, ADRES = 939, Reflected Power = 161.1 W
retlw      0x89            ; High Byte, ADRES = 943, Reflected Power = 176.6 W
retlw      0x8A            ; High Byte, ADRES = 947, Reflected Power = 193.6 W
```

```
    retlw      0x8A            ; High Byte, ADRES = 951, Reflected Power = 212.2 W
    retlw      0x8B            ; High Byte, ADRES = 955, Reflected Power = 232.6 W
    retlw      0x8B            ; High Byte, ADRES = 959, Reflected Power = 254.9 W
    retlw      0x8C            ; High Byte, ADRES = 963, Reflected Power = 279.4 W
    retlw      0x8C            ; High Byte, ADRES = 967, Reflected Power = 306.3 W
    retlw      0x8D            ; High Byte, ADRES = 971, Reflected Power = 335.7 W
    retlw      0x8D            ; High Byte, ADRES = 975, Reflected Power = 368.0 W
    retlw      0x8E            ; High Byte, ADRES = 979, Reflected Power = 403.4 W
    retlw      0x8E            ; High Byte, ADRES = 983, Reflected Power = 442.2 W
    retlw      0x8F            ; High Byte, ADRES = 987, Reflected Power = 484.7 W
    retlw      0x90            ; High Byte, ADRES = 991, Reflected Power = 531.3 W
    retlw      0x90            ; High Byte, ADRES = 995, Reflected Power = 582.3 W
    retlw      0x90            ; High Byte, ADRES = 999, Reflected Power = 638.3 W
    retlw      0x91            ; High Byte, ADRES = 1003, Reflected Power = 699.7 W
    retlw      0x91            ; High Byte, ADRES = 1007, Reflected Power = 766.9 W
    retlw      0x92            ; High Byte, ADRES = 1011, Reflected Power = 840.6 W
    retlw      0x93            ; High Byte, ADRES = 1015, Reflected Power = 921.5 W
    retlw      0x93            ; High Byte, ADRES = 1019, Reflected Power = 1,010.0 W
    retlw      0x94            ; High Byte, ADRES = 1023, Reflected Power = 1,107.1 W

endif


   end
```

# ARRL Periodicals Archive – Search Results

## A membership benefit of ARRL and the ARRL Technical Information Service

**ARRL Members:** You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

**Need a higher quality reprint or scan?** Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

**QST on CD-ROM**: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

**Non-Members:** Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.


### Copyright/Reprint Notice

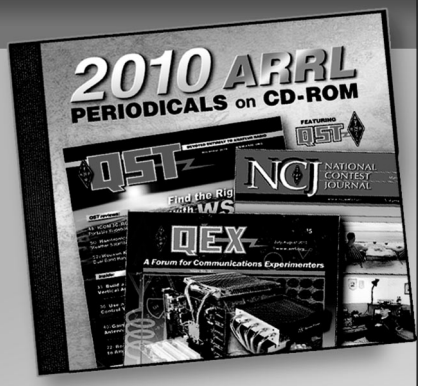
In general, all ARRL content is copyrighted. ARRL articles, pages, or documents--printed and online--are not in the public domain. Therefore, they may not be freely distributed or copied. Additionally, no part of this document may be copied, sold to third parties, or otherwise commercially exploited without the explicit prior written consent of ARRL. You cannot post this document to a Web site or otherwise distribute it to others through any electronic medium.

For permission to quote or reprint material from ARRL, send a request including the issue date, a description of the material requested, and a description of where you intend to use the reprinted material to the ARRL Editorial & Production Department: permission@arrl.org.


**QST Issue:** Jan 1988
**Title:** Tandem Match Corrections
**Author:** John Grebenkemper, KI6WX

Click Here to Report a Problem with this File


ADVERTISEMENT




**2010 ARRL Periodicals**
on CD-ROM

**ARRL's popular journals are available on a compact, fully-searchable CD-ROM. Every word and photo published throughout 2010 is included!**

- ***QST*** **The official membership journal of ARRL**
- ***NCJ*** **National Contest Journal**
- ***QEX*** **Forum for Communications Experimenters**

**SEARCH** the full text of every article by entering titles, call signs, names—almost any word. **SEE** every word, photo (including color images), drawing and table in technical and general-interest features, columns and product reviews, plus all advertisements. **PRINT** what you see, or copy it into other applications.

**System Requirements**: Microsoft Windows™ and Macintosh systems, using the industry standard Adobe® Acrobat® Reader® software. The Acrobat Reader is a free download at **www.adobe.com**.

**2010 ARRL Periodicals on CD-ROM**

ARRL Order No. 2001
**Only $24.95***
*plus shipping and handling

Additional sets available:
2009 Ed., ARRL Order No. 1486, $24.95
2008 Ed., ARRL Order No. 9406, $24.95
2007 Ed., ARRL Order No. 1204, $19.95
2006 Ed., ARRL Order No. 9841, $19.95
2005 Ed., ARRL Order No. 9574, $19.95
2004 Ed., ARRL Order No. 9396, $19.95
2003 Ed., ARRL Order No. 9124, $19.95
2002 Ed., ARRL Order No. 8802, $19.95
2001 Ed., ARRL Order No. 8632, $19.95

ARRL The national association for AMATEUR RADIO™
SHOP DIRECT *or call for a dealer near you.*
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)

Jan 1988 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

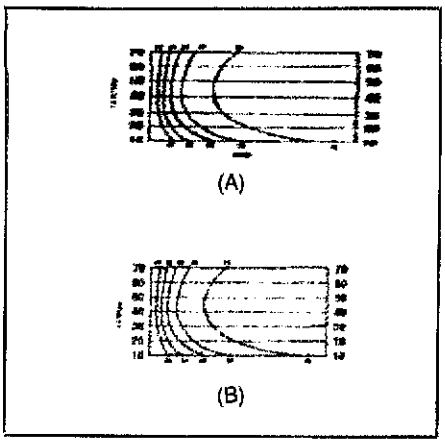



Fig 5—A comparison of picture sections made with the Sueker interface (A) and the simple interface (B).

gram to allow the use of the simpler interface.) These two picture samples were taken at noon to minimize multipath smear. The condensed mode, incidentally, is more readable than the full-width mode even though it is smaller. (The difference is comparable to that of NLQ v draft printing.) I do, however, occasionally have to use a magnifying glass to read the map. The condensed mode uses paper at half the rate of the full-width mode, and each map has a convenient note space next to it.

An automatic synchronizing code measures the black-dot count per line to detect the repeated 95% black lines during the sync period. The code then actuates the left edge of the picture just as a white tick occurs.

The scheduling program simply uses the computer's clock to turn the copying program on and off at whatever times you enter. A TUNE program is included to set up the receiver tuning before each copying period. This program also directly measures the clipper asymmetry so you can adjust the audio level for best results.

All of the accompanying pictures were made with an 8-dot, IBM-compatible printer using the Cardco + G printer interface with the C64. (The Xetec printer interface should also be compatible.) As is, the program should work with any printer using the 960-dot line command: CHR($27);"L";CHR$(n1);CHR$(n2), and the command CHR$(27);"J";CHR$(n) to advance the printer platen by n/216 inch. You can POKE in your own printer commands by merely adding a program line to the BASIC program used to control the whole operation.

Because a lot of folks still have 7-dot printers and may be interested in using the simple interface of Fig 4, I developed a program to accommodate them. Of course, the condensed mode and gray shades require an 8-dot NLQ printer, so are not available with the 7-dot program. (A good medium-resolution screen display program is included on the disk.)

I'll be on a sailing cruise when this information is published, so I've arranged for the New Bern Amateur Radio Club to provide disks to interested readers; their fee is $10 per disk. Be sure to specify which printer you're using (7- or 8-dot), and which interface (Sueker or simpler clipper) you plan to use. You can obtain the disks by contacting the New Bern ARC, PO Box 2483, New Bern, NC 28561.—*Ben Vester, K3BC, 4921 Bonnie Branch Rd, Ellicott City, MD 21043*

### TANDEM MATCH CORRECTIONS

☐ There are six errors in the schematic diagram of "The Tandem Match—An Accurate Directional Wattmeter," Jan 1987 *QST*, pp 18-26. These errors are all located in the signal-processing portion of the circuit (p 23) and occurred from my incorrect tracing of the maze of wires on the original breadboard. The corrections include:

- The 330-kΩ resistor should be connected between the −2.5 V supply and the cathode of D14 (TP 9), not pin 6 of U4B.
- Change the diagram to show pins 4 and 7 of U7 to the 2.2-kΩ resistor attached to pin 1 of U3A; pin 8 of U7 connected to pin 2 of U3A.
- Insert a 1N914 diode in the line between the collector of Q4 and the circuitry associated with U3D and U7. The anode of the diode is connected to the collector of Q4, the diode cathode attaches to pin 13 of U3D.
- The 1-kΩ resistor and D10 connected between U4A pin 1 and the two 100-kΩ resistors should be eliminated and replaced by a direct connection between these two points.
- The jacks on the right-hand side of the diagram (p 23) labeled J1 and J2 should be labeled J3 and J4, respectively.
- On p 20, Fig 5, the 57-Ω resistor should be labeled 50-Ω.

Radio Shack no longer lists ICs U1-U4 numbered 276-1749 and 276-1750. Some stores may still have them in stock, however. There are other sources of supply for various items. The TLC27L2, TLC27L4 and CA3146 ICs can be purchased from Newark Electronics, 4801 N Ravenswood St, Chicago, IL 60640, tel 312-784-5100. The 1N5711 diodes are available from Surplus Sales, 2412 Chandler Rd, Bellevue, NE 68005, tel 402-733-9190. The LM334, LM336, 1% tolerance resistors and trimmer potentiometers are carried by Digi-Key Corporation, 701 Brooks Ave South, PO Box 677, Thief River Falls, MN 56701, tel 800-344-4539.

I've received many inquiries concerning the T-50-3 toroid material used for transformers T1 and T2 mentioned on p 24; the number is correct. Type 3 material is required to obtain sufficient inductance for the transformers to work on 160 meters. If Type 2 or 6 material is used, the directional coupler will not work on 160 meters, and may not work on 80 meters. Type 3 material works fine through 50 MHz, and at 50 MHz, the difference in performance between Type 3 and Type 6 material cannot be measured.

On p 26, the date in note 8 should be corrected to read 1964.

My thanks to Dick Green, KA1LBW, who spent much time and effort working with me to locate the errors, and who provided the list of parts vendors. Lastly, please make note of my new address. —*John Grebenkemper, KI6WX (ex-KA3BLO), 19490 Miller Ct, Saratoga, CA 95070*

[Editor's Note: Photocopies of the corrected schematic diagram (p 23) for The Tandem Match are available free of charge from the Technical Department Secretary, ARRL, 225 Main St, Newington, CT 06111. Please identify your request as Corrected Tandem Match Schematic Diagram, QS-01/87.]

**Note:** All correspondence addressed to this column should bear the name, call sign and complete address of the sender. Please include a daytime telephone number at which you may be reached if necessary.

# Feedback

☐ In "Amateur Radio and the Blind, Part 1," Oct 1987 *QST*, please make this change to the sidebar on p 28, left-hand column, fifth entry. The correct spelling of Bill's last name is Gerrey, and his call sign is WA6NPC. (Tnx to Ricardo J. Alfano II, W6FWX, for this information.)

☐ Author Steve Stuntz provides us with the following corrections to his article, "A Packet Terminal for Atari Computers," Nov 1987 *QST*, p 17, Fig 4. On the DB-25 connector, the TRANSMIT DATA line should be shown connected to pin 2 (not pin 1); the RECEIVE DATA line connects to pin 3 (not pin 2).

# Strays

**I would like to get in touch with...**

☐ anyone with a schematic/parts list for a Knight KG663 regulated power supply. Lyle Seehorn, W7YKA, 3625 SW 328th St, Federal Way, WA 98023.

☐ anyone with a schematic and parts list for a Midland Model 13-500 2-meter rig. Harold McCullen, K8LNR, 2215 N Charles St, Saginaw, MI 48602.

# Test and Measurement Bibliography

The following articles on test and measurement are from *QEX* and *QST* between 1995 and April 2013. The list was compiled by Alan Bloom, N1AL.

Ahlstron, James, N2ADR, "Using a Digital Transceiver as a Vector Network Analyzer," QEX, Jan/Feb 2013, pp 3-6.

Alldread, Thomas M., VA7TA, "NimbleSig III - Part 1," QEX, Jan/Feb 2009, pp 3-20.
Alldread, Thomas M., VA7TA, "NimbleSig III - Part 2," QEX, Mar/Apr 2009, pp 16-24.
Alldread, Thomas M., VA7TA, "NimbleSig III - Part 3," QEX, May/June 2009, pp 10-20.

Baier, Thomas C., DG6SAQ, "A Low Budget Vector Network Analyzer for AF to UHF," QEX, Mar/Apr 2007, pp 46-54.

Baier, Thomas C., DG8SAQ, "A Small, Simple USB-Powered Vector Network Analyzer Covering 1 kHz to 1.3 GHz," QEX, Jan/Feb 2009, pp 32-36.
Baier, Thomas C., DG8SAQ, "A Simple S-Parameter Test Set for the VNWA2 Vector Network Analyzer," QEX, May/June 2009, pp 29-32.

Basilier, Erik, WU7O, "Build a Power Attenuator," QST, November 1996.

Bird, Trevor S., "Definition and Misuse of Return Loss," QEX, Sept/Oct 2010, pp 38-39.

Bloom, Alan, N1AL, "A Modern GDO - The 'Gate" Dip Oscillator," QST, May 2003, pp 54-58.

Bloom, Alan, N1AL, "Power Factor" (Technical Correspondence), QST, April 2004, pp 82-83.

Bloom, Alan, N1AL, "Some Thoughts on Crystal Parameter Measurement" (Letter), QEX, Sept/Oct 2008, pp 41-42.

Bloom, Alan, N1AL, "A Transistor Tester in a Tin," QST, January 2012, pp 30-31.

Bloom, Alan, N1AL, "A Spice Bottle Logic Probe," QST, June 2012, pp 41-43.

Copyright © 2013 American Radio Relay League, Inc. All rights reserved.

Bloom, Jon, KE3Z, "Measuring System Response with DSP," QEX, February 1995.

Bloom, Jon, KE3Z, "Measuring 9600-Baud Radio BER Performance," QEX, March 1995.

Bowker, Dave, K1FK, "An RF Phase Meter," QEX, November/December 2010, pp 25-31.

Bowman, Michael P., KG2MG, "A Simple Path to Complex Impedance," QEX, Sept/Oct 2010, pp 3-15.

Bradley, Mark, K6TAF, "What Can You Do eith a Dip Meter?," QST, May 2002, pp 65-68.

Bramwell, Denton, K7OWJ, "AN RF Step Attenuator," QST, June 1995.

Bramwell, Denton, K7OWJ, "The Microwatter," QST, June 1997.

Bryce, Mike, WB8VGE, "A Universal Frequency Calibrator," QST, Nov 2009, pp 35-37.

Christensen, Paul, W9AC, "An SWR Null Meter," QST, Feb 2010, pp 30-32.

Cicchinelli, Larry, K3PTO, "A Homebrew Logic Analyzer," QEX, Jan/Feb 2000, pp 37-46.

Cicchinelli, Larry, K3PTO, "A PC-Based Digital Storage Oscilloscope and Logic Analyzer," QEX, July/Aug 2003, pp 13-29.

Cooper, Sid, K2QHE, "A Compensated, Modular RF Voltmeter," QEX, Mar/Apr 2001, pp 26-34.

Danzer, Paul, N1II, "A Simple Transformer to Measure Your Antenna Current," QST, Sept 2009, p 35.

Daretti, Andrew, IZ2OUK, "A Simple and Effective RF Power Reference," QEX, Sept/Oct 2010, pp 32-34.

Dildine, Bob, W6SFH, "A Sampling Down Converter for Low Frequency Oscilloscopes," QST, January 2012, pp 39-43.

Copyright © 2013 American Radio Relay League, Inc. All rights reserved.

Eichel, Klaus H., DL6SES, "Avoiding Pitfalls During Measurements of High-Performance HF Receivers" (Tech Notes), QEX, Jan/Feb 2004, pp 59-60.

Evans, Randy, KJ6PO, "A Laboratory-Grade 10-MHz Frequency Standard," QEX, May/Jun 2002, pp 13-21.

Fernandes, Rubens R., ex-PY2QE, "A Frequency Counter for the Experimenter," QEX, May/June 2010, pp 10-15.

Ford, Jim, N6JF, "Build a Return Loss Bridge," QST, September 1997, p34-36.

Forrer, Johann B., KC7WW, "A Low-Cost HF Channel Simulator for Digital Systems," QEX, May/June 2000, pp 13-22.

Franke, John M., WA4WDL, "Double Balanced Beat Indicator," QEX, Jan/Feb 2012.

Geissinger, Gary, WAØSPM, "Building Inexpensive HF Power Attenuators," QST, November 2012, pp 32-33.

Gilbert, Steve, K1SG, "An Expanded Scale Voltmeter for 120 or 240 V ac - With a Bonus," QST, November 2010, pp 43-44.

Glenn, Frederick, K9SO, "Repair and Calibrate Those Wattmeter 'Slugs' with Confidence,", QST, Nov 2009, pp 50-52.

Grant, Ken, VE3FIT, "A Versatile Two-Tone Audio Generator for SSB Testing," QEX, Mar/Apr 2009, pp 25-28.

Green, Rod, VK6KRG, "Testing Receivers - Some Thoughts," QEX, Mar/Apr 2004, pp 53-55.

Green, Sam, WØPCE, "Fully Automated DDS Sweep Generator Measurement System," QEX, Nov/Dec 2008, pp 13-22.
Green, Sam, WØPCE, "Fully Automated DDS Sweep Generator Measurement System - Take 2," QEX, Sept/Oct 2012, pp 14-24.

Green, Sam, WØPCE, "An Oscilloscope Camera Mount," QEX, May/June 2010, pp 37-40.

Copyright © 2013 American Radio Relay League, Inc. All rights reserved.

Hageman, Steve, "Build Your Own Network Analyzer, Part 1," QST, January 1998.
Hageman, Steve, "Build Your Own Network Analyzer, Part 2," QST, February 1998.

Hageman, Steve, "Build this 250-MHz Synthesized Signal Source," QEX, Jan/Feb 2000, pp 3-15.

Hageman, Steve, "Build a Data Acquisition System for your Computer" (Tech Notes), QEX, Jan/Feb 2001, pp 52-57.

Hageman, Steve, "Build a 250-MHz Network Analyzer," QEX, Mar/Apr 2002, pp 3-10.

Hardcastle, Jack, G3JIR, "Quartz Crystal Parameter Measurement," QEX, Jan/Feb 2002, pp 7-11.

Hare, Ed, KA1CV, "Swept Receiver Dynamic Range Testing in the ARRL Laboratory," QEX, June 1996.

Hayward, Roger, KA7EXM, “A PIC-based HF/VHF Power Meter,” QEX, May/Jun 2005, pp 3-10.
Hayward, Roger, KA7EXM, Applications Using the PIC-Power Meter," QST, June 2005, pp 28-30.

Hayward, Wes, W7ZOI, et al, "A Spectrum Analyzer for the Radio Amateur, Part 1" QST, August 1998.
Hayward, Wes, W7ZOI, et al, "A Spectrum Analyzer for the Radio Amateur, Part 2" QST, September 1998.

Hayward, Wes, W7ZOI, et al, "A Tracking Signal Generator for Use with a Spectrum Analyzer," QST, November 1999, pp 50-52.

Hayward, Wes, W7ZOI, et al, "Simple RF-Power Measurement," QST, June 2001, pp 38-43.

Hayward, Wes, W7ZOI, "Oscillator Noise Evaluation with a Crystal Notch Filter," QEX, July/Aug 2008, pp 6-12.

Copyright © 2013 American Radio Relay League, Inc. All rights reserved.

Hontoria, James, W1JGH, "A 10 MHz to 6 GHz Power Meter," QEX, July/Aug 2011, pp 17-21.

Huyett, Martin, KØBXB, "Antenna Measurement for the Ham on a Budget," QST, Dec 2010, pp 38-39.

Isley, Cary T., "A Simple Frequency Deviation Meter Using Time-Domain Techniques," QEX, August 1995.

Jones, Bill, K8CU, "Using the HP Z3801A GPS Frequency Standard," QEX, Nov/Dec 2002, pp 49-53.

Karlsen, Kjell, LA2NI, "A Method of Measuring Phase Noise in Oscillators," QEX, Nov/Dec 2004, pp 55-59.

Kaune, Bill, W7IEQ, "A Modern Directional Power/SWR Meter," QST, January 2011, pp 39-43.

Kelley, Bert, AA4FB, "A PIC SWR Meter," QST, December 1999, pp 40-43.

Kincaid, Dwayne L., "The DWM4: A Microprocessor Controller Multichannel Wattmeter for HF, VHF and UHF," QST, July 1999, pp 42-46.

Koehler, Jim A., VA7DIJ, "An L-Q Meter", QEX, Mar/Apr 2006, pp25-34.

Koehler, Jim A., VE5FP, "Some Thoughts on Crystal Parameter Measurement", QEX, July/Aug 2008, pp36-41.

Koehler, Jim, VE5FP, "A DDS-Based Signal Generator," QEX, Nov/Dec 2009, pp 3-13.

Koehler, Jim, VE5FP, "The Shunt Method for Crystal Parameter Measurement," QEX, July/Aug 2010, pp 16-25.

Kopski, Bob, K3NHI, "An Advanced VHF Wattmeter," QEX, May/Jun 2002, pp 3-8.

Kopski, Bob, K3NHI, "A Simple RF Power Calibrator," QEX, Jan/Feb 2004, pp 51-54.

Littlefield, Rick, K1BQT, "A Wide-Range RF-Survey Meter," QST, August 2000, pp 42-44.

Copyright © 2013 American Radio Relay League, Inc. All rights reserved.

Lyndon, Dave, AK4AA, "Accurate Measurement of Small Inductances," QEX, Jan/Feb 2004, pp 47-50.

McDermott, Tom, N5EG, et al, "A Low-Cost 100 MHz Vector Network Analyzer with USB Interface," QEX, Jul/Aug 2004.

Miller, Bob, KE6F, "Atomic Frequency Reference for Your Shack," QEX, Sept/Oct 2009, pp 35-44.

Nash, Bob, KF6CDO, "An Event per Unit Time Measurement System for Rubidium Frequency Standards," QEX, November/December 2010, pp 3-17.

Nellis, Mert, WØUFO, "Build an Ampere-Hour Meter for Portable Operations," QST, May 2010, pp 44-45.

Nichols, Eric P., KL7AJ, "Keeping Current with Antenna Performance," QST, Feb 2009, pp 34-36.

Noakes, John D., VE7NI, "The 'No Fibin' RF Field Strength Meter," QST, August 2002, pp 28-29.

Noskowicz, Steve, K9DCI, "Keeping Current with Antenna Performance" (Technical Correspondence), QST, July 2010, pp 50-51. [see Nichols above]

Ostapchuk, Pete, N9SFX, "A Rugged, Compact Attenuator," QST, May 1998.

Pivnichny, John, N2DCH, "A Signal Generator for the VHF Operator," QST, October 2010, pp 34-36.

Pontius, Bruce E., NØADL, "Measurement of Signal-Source Phase Noise with Low-Cost Equipment," QEX, May/Jun 1998.

Rech, Henry J., "Receiver Performance Measurement and Front End Selectivity," QEX, Sept/Oct 2008, pp 26-30.

Copyright © 2013 American Radio Relay League, Inc. All rights reserved.

Reed, John C., "Spectral Measurements the Hard Way," QEX, July 1995.

Richardson, Gary, AA7VM, "A Low Cost DDS Function Generator," QST, November 2005, pp 40-42.

Rogers, Robert J. WA1PIO, "Mystery Inductor Box," QST, January 2011, pp 30-31.

Rohde, Ulrich L., KA2WEU, "Theory of Intermodulation and Reciprocal Mixing: Practice, Definitions and Measurements in Devices and Systems, Part I," QEX, Nov/Dec 2002, pp 3-15.
Rohde, Ulrich L., KA2WEU, "Theory of Intermodulation and Reciprocal Mixing: Practice, Definitions and Measurements in Devices and Systems, Part II," QEX, Jan/Feb 2003, pp 21-31.

Roos, John C., K6IQL, "Converting a Vintage 5 MHz Frequency Standard to 10 MHz with a Low Spurious Frequency Doubler," QEX, March/April 2011, pp 19-35.

Rumley, Stuart, KI6QP, "A Precision Two-Tone RF Generator for IMD Measurements," QEX, April 1995.

Sabin, William E., WØIYH, "The Lumped-Element Directional Coupler," QEX, March 1995.

Shera, Brooks, W5OJM, "A GPS-Based Frequency Standard," QST, July 1998.

Shrader, Bob, W6BNB, "Measuring Radio Frequencies," QST, Sept 2009, pp 36-39.

Silver, H. Ward , NØAX, "Test Equipment for the Ham Shack." QST, May 2005, pp 36-39.

Skelton, Ron, W6WO, "Measuring HF Balun Performance," QEX, November/December 2010, pp 39-41.

Smith, Doug, KF6DX, "Improved Dynamic-Range Testing," QEX Jul/Aug 2002, pp 46-52.

Smith, Doug, KF6DX, "Quantifying Measurement Uncertainty" (Tech Notes), QEX Jan/Feb 2006, pp54-56.

Smits, Harke, PAØHRK, "A Noise/Gain Analyzer," QEX, Nov/Dec 1999, pp 5-10.

Copyright © 2013 American Radio Relay League, Inc. All rights reserved.

Sparks, Steve L., N5SV, "An RF Current Probe for Amateur Use," QST, February 1999, p 34.

Stanley, John O., K4ERO, "The Beauty of Spectrum Analysis - Part 1", QST, June 2008, pp35-38.
Stanley, John O., K4ERO, "The Beauty of Spectrum Analysis - Part 2", QST, July 2008, pp33-35.

Stanley, Wayne J., W4RDG, "Check the Calibration of your Maxwell Bridge with an Inductance Simulator" (Technical Correspondence), QST, March 1997.

Stanley, Wayne J., W4RDG, "A Calibration Source for DC and AC Voltmeters," QEX, July/Aug 2000, pp 51-54.

Steber, George R., WB9LVI, “LMS Impedance Bridge,” QEX, Sep/Oct 2005, pp 41-47.
Steber, George R., WB9LVI, "A Low Cost Automatic Impedance Bridge," QST, October 2005, pp. 36-39.

Steber, George R., WB9LVI, "I-V Curve Tracing With A PC," QEX, July/August 2006, pp. 3-9.

Steber, George R., WB9LVI, "Experimenter's RF Spectrum Analyzer," QST, October 2008, pp 36-40.

Steinbaugh, Gary, AF8L,"Pizzicato Pulse Generator", QEX, Mar/Apr 2006, pp3-8. [Time-domain reflectometer]

Steinbaugh, Gary, AF8L, "A Cybernetic Sinusoidal Synthesizer," QEX March/April 2009, pp 3-5.
Steinbaugh, Gary, AF8L, "A Cybernetic Sinusoidal Synthesizer: Part 2," QEX May/June 2009, pp 21-28.
Steinbaugh, Gary, AF8L, "A Cybernetic Sinusoidal Synthesizer: Part 3," QEX July/Aug 2009, pp 3-7..
Steinbaugh, Gary, AF8L, "A Cybernetic Sinusoidal Synthesizer: Part 4," QEX Sept/Oct 2009, pp 21-30.

Steinbaugh, Gary, AF8L, "An Inexpensive Laboratory-Quality RF Wattmeter," QEX, May/June 2010, pp 26-32.

Copyright © 2013 American Radio Relay League, Inc. All rights reserved.

Terwilliger, Curt, W6XJ, "Building a Modern Signal Tracer," QST, February 2011, pp 41-44.

Thompson, Tom, WØIVJ, "A High Power RF Sampler" (Technical Correspondence), QST, May 2011, pp 52-54.

Tipton, Ron, ex-K5UJC, "Build this Simple, High-Resolution DC Voltmeter," QEX, Sept/Oct 2001, pp 38-44.

Toal, Ed, N9MW, "A Peak Reading RF Power Meter", QST, March 2010, pp 41-43.

VanRemmen, Bill, KA2WFJ, "A Pocket-Size, Direct-Reading VHF SWR Meter," QST, February 1997.

van der List, Jos F. M., PAØJOZ, "Experiments with Phase-Noise Measurements," QEX, Jan/Feb 1999, pp 31-41.

von Valtier, Eric, K8LV, "Dual Directional Wattmeters," QEX, May/June 2006, pp 23-33.

Wade, Paul, N1BWT, "A UHF+ VSWR Bridge," QEX, February 1995.

Wade, Paul, N1BWT, "Noise Measurement and Generation," QEX, November 1996.

Wade, Paul, W1GHZ, "Microwave System Test" (Microwavelengths), QST, Aug 2010, pp 96-97.

Wade, Paul, W1GHZ, "Antenna Measurements - the Antenna Range" (Microwavelengths), QST, October 2012, pp 63-64.
Wade, Paul, W1GHZ, "Antenna Measurements: Methods and Equipment," (Microwavelengths), QST, January 2013, pp 66-67.

Whiteside, Steve, N2PON, "Pocket dBm RF Power Meter," QST, August 2008, pp 65-67.

Whiteside, Steve, N2PON, "A Linear Scale Milliohm Meter," QEX, July/Aug 2012, pp 33-38.

Copyright © 2013 American Radio Relay League, Inc. All rights reserved.

Zimmerman, Robert K., VE3RKZ, "An Inexpensive HF Power Meter with a Linear dB Scale," QEX, Mar/Apr 2010, pp 44-46.

Copyright © 2013 American Radio Relay League, Inc. All rights reserved.

# ARRL Periodicals Archive – Search Results

## A membership benefit of ARRL and the ARRL Technical Information Service

**ARRL Members:** You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

**Need a higher quality reprint or scan?** Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

**QST on CD-ROM**: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

**Non-Members:** Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.


### Copyright/Reprint Notice

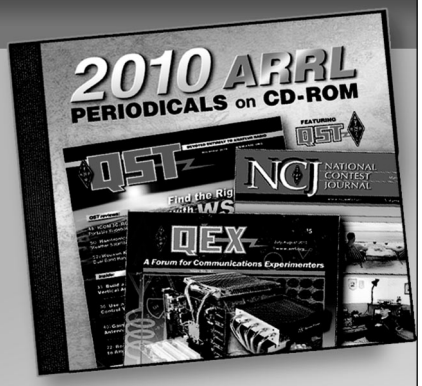
In general, all ARRL content is copyrighted. ARRL articles, pages, or documents--printed and online--are not in the public domain. Therefore, they may not be freely distributed or copied. Additionally, no part of this document may be copied, sold to third parties, or otherwise commercially exploited without the explicit prior written consent of ARRL. You cannot post this document to a Web site or otherwise distribute it to others through any electronic medium.

For permission to quote or reprint material from ARRL, send a request including the issue date, a description of the material requested, and a description of where you intend to use the reprinted material to the ARRL Editorial & Production Department: permission@arrl.org.


**QST Issue:** Jan 1974
**Title:** Art of Dipping, The
**Author:** Benjamin Clark, WB4OBZ

Click Here to Report a Problem with this File


ADVERTISEMENT




**2010 ARRL Periodicals**
on CD-ROM

**ARRL's popular journals are available on a compact, fully-searchable CD-ROM. Every word and photo published throughout 2010 is included!**

- ***QST*** **The official membership journal of ARRL**
- ***NCJ*** **National Contest Journal**
- ***QEX*** **Forum for Communications Experimenters**

**SEARCH** the full text of every article by entering titles, call signs, names—almost any word. **SEE** every word, photo (including color images), drawing and table in technical and general-interest features, columns and product reviews, plus all advertisements. **PRINT** what you see, or copy it into other applications.

**System Requirements**: Microsoft Windows™ and Macintosh systems, using the industry standard Adobe® Acrobat® Reader® software. The Acrobat Reader is a free download at **www.adobe.com**.

**2010 ARRL Periodicals on CD-ROM**

ARRL Order No. 2001
**Only $24.95***
*plus shipping and handling

Additional sets available:
2009 Ed., ARRL Order No. 1486, $24.95
2008 Ed., ARRL Order No. 9406, $24.95
2007 Ed., ARRL Order No. 1204, $19.95
2006 Ed., ARRL Order No. 9841, $19.95
2005 Ed., ARRL Order No. 9574, $19.95
2004 Ed., ARRL Order No. 9396, $19.95
2003 Ed., ARRL Order No. 9124, $19.95
2002 Ed., ARRL Order No. 8802, $19.95
2001 Ed., ARRL Order No. 8632, $19.95

ARRL The national association for AMATEUR RADIO™
SHOP DIRECT *or call for a dealer near you.*
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)

Jan 1974 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

• *Beginner and Novice*

# The Art of Dipping

## Twenty Ways to Use Your Dipper

BY BENJAMIN CLARK,* WB4OBZ

THE GRID-DIP oscillator has long been a standard piece of test equipment in the ham shack. All recent editions of the *Handbook* have carried a circuit or two and discussed its uses. Several articles each year in *QST* treat the dipper, giving new designs or new ways of using this useful instrument. This article is an attempt to summarize uses of the dipper, in the hope that it may serve as a single reference.

To obtain the fullest use of the dipper three accessories are needed, a "standard" capacitor, a "standard" inductor and a calibrated all-band receiver. A capacitor of 100 pF and an inductor of 5 μH are recommended in the measurement section of the *Handbook*, and these values handle most any need.

* Mepkin Abbey, Route 3, Box 357, Moncks Corner, SC 29461.

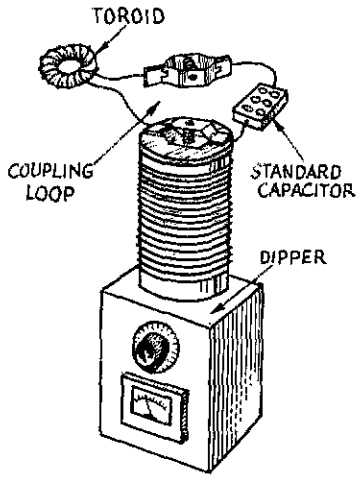



Fig. 1 — One way to measure toroids. The same technique can be used with shielded coils.

### *Resonant Frequency*

The original use of the dipper was to measure the resonant frequency of a tuned circuit, and this use has been sufficiently covered that it merely needs to be mentioned here. One couples the dipper to the circuit to be measured, tunes the dipper until the meter dips, indicating resonance. The only precautions to be observed are to use the loosest coupling that will give a good dip, and to be sure that the dipper is indicating at the resonant frequency of the circuit to be measured and not at the resonant frequency of a circuit within the dipper itself. It's a good idea with any dipper, commercial or homemade, to check all ranges with the unit not coupled to anything and make note of any dips which occur. These are internal resonances within the dipper itself and knowing where they are can save much time and frustration when the unit is in actual use. One expects commercially built dippers to be free of such resonances, but a homemade unit may not be.

The calibration of the ordinary dipper is only approximate, close enough for most practical purposes, but not for all. Furthermore hand capacitance and the effect of the circuit to which the dipper is coupled will detune the oscillator somewhat. When greater accuracy is needed, one can make use of a frequency counter, or if this is not available, a calibrated frequency meter or receiver.

All dippers can be used as absorption frequency meters, for an absorption frequency meter is nothing more than a calibrated tuned circuit with some sort of resonance indicator. In every dipper, the current that passes through the meter is rectified rf from the tuned circuit, whether rectified by the grid circuit of a vacuum tube, by the gate junction of a JFET, or by a separate rectifier connected across the tuned circuit. It will rectify equally well whether the rf is generated by the dipper oscillator or is picked up from outside. With tube-type dippers, the plate voltage must be removed without cutting the heater or filament supply, to allow the grid-cathode circuit to act as a diode rectifier. An adjustable plate voltage is even

Jan 1974 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

more helpful, as this will permit the oscillator circuit to act as a $Q$ multiplier when the voltage is adjusted to just below the point of oscillation, giving a sharper and more sensitive peak when the tuned circuit picks up rf to which it is resonant.

### *Inductance*

With a capacitor standard it becomes easy to measure the inductance of a coil. Simply connect the coil in parallel with the capacitor and measure the resonant frequency. Then if you feel like doing some figuring, you can use the formula for finding resonant frequencies to determine the inductance, or you can do it the easy way by using an L/C/F calculator.[1]

The usual way of coupling to a coil inductively will not work as well when the coil to be tested is a toroid or is shielded, precisely because these types of coils are intended to have as little external field as possible. There are three ways that one can couple them to the dipper for measurement, however. If the dipper has the hot end of the coil connected to an external terminal, a small capacitor, say 5 pF, can be connected between that terminal and one side of the coil and capacitor standard combination, the other side being connected to the dipper chassis.

Another method for coupling to toroids and shielded coils makes use of the fact that the loop formed by the leads from the capacitor standard to the coil forms one turn of the coil, and can be coupled inductively to the dipper. The coupling is quite loose, but the high $Q$ of the unloaded coil will usually make it adequate to get a good dip at resonance. See Fig. 1.

A third method for toroids is to wind links around the toroid coil and the dipper coil and connect them together as described in the April, 1973 issue of *QST* page 60, and shown in Fig. 2.

In the design of inductively coupled bandpass filters, one often needs to know the mutual inductance or coefficient of coupling of a pair of coils. The *Handbook* gives the following method: measure the inductance of one of the coils with the other first open, then shorted. The coefficient of coupling, ($k$) is found from the formula:

[1] LaPlaca, "Using the ARRL L/C/F Calculator," *QST*, December, 1973.

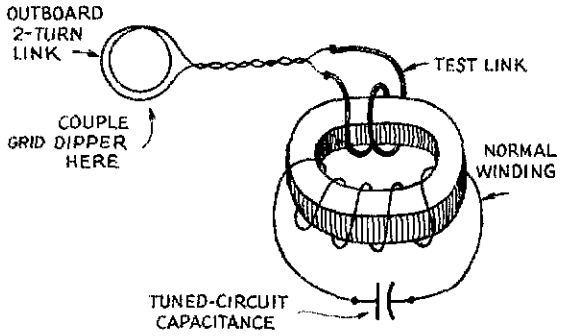



Fig. 2 — Another method for testing toroids.

$$k = \sqrt{1 - \frac{Ls}{Lo}}$$

where $Ls$ is the inductance measured with the other coil shorted, $Lo$ is the value measured when the short is removed. See Fig. 3 a and b. With known capacitors in use, measuring the frequency will enable you to determine the inductance.

Mutual inductance can be measured by finding the inductance of the two coils connected in series, noting the value, then reversing the connection to one of the coils and measuring the inductance again. Half the difference between the two measurements is the mutual inductance. Mutual inductance ($M$) and coefficient of coupling ($k$) are related by the formula:

$$k = \frac{M}{\sqrt{L1L2}}$$

For accurate measurement of mutual inductance or coefficient of coupling, something better than the dipper calibration may be needed, and may be obtained by using the frequency counter method described earlier. If one has a counter, or a receiver covering the required range, fine. But if your receiver is of the amateur-bands-only variety, your capacitor standard may not be able to tune the coil within your receiver's range. In that case, use any capacitor that will do

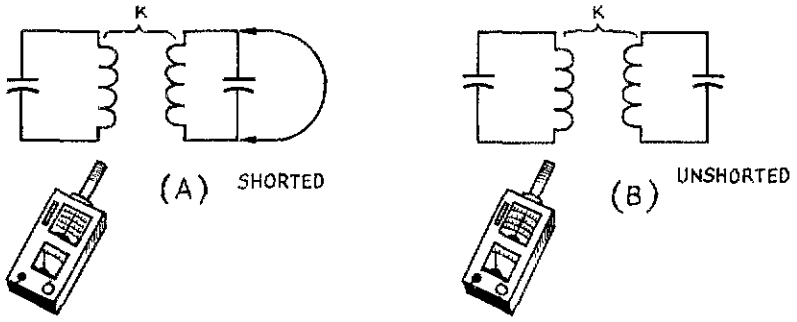



Fig. 3 — This shows how to determine the value of an unknown inductance as described in the text.

Jan 1974 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

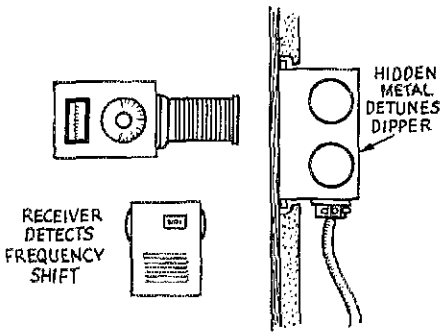



Fig. 4 — Locating hidden metals with a dipper.

so, and compute the coefficient of coupling as shown above, but using another formula that does not require that one determine the actual value of the inductance:

$$k = \sqrt{1 - \frac{fo^2}{fs}}$$

where *fo* and *fs* are the frequencies measured with the other coil open and shorted respectively.

### *Capacitance*

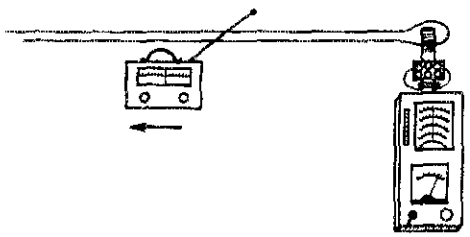
Capacitance is measured just like inductance except, of course, that you connect the inductance standard in parallel with the capacitor to be measured.

### *Transmission Lines*

In constructing antennas, a quarter-wave section of line at the operating frequency is often needed. If one-fourth the wavelength in free space were required, the length could simply be computed by formula. But the electrical length of a line is always greater than the measured length; therefore the velocity factor of the coax should be taken into account. This will assure your getting the proper line length to within a few inches. To get a quarter-wave line, attach a loop to one end, leave the other end open, and dip. Trim until the dip is at the desired frequency. When the dip is where you want it, you have your quarter-wave line (or any odd multiple of a quarter wave, for that matter).

For a half-wave line, do the same as described above, except that the far end should be shorted, and a small coupling loop for the dipper should be used at the near end of the line. If the line is of reasonably good quality, the dip should be quite sharp in both cases.

Fig. 5 — A method for tracing hidden pipes or wiring using a bc receiver and a dipper.



To measure the velocity factor of a line, simply reverse the procedure: find the frequency at which the length of line is resonant (far end open), measure the length, and use the formula:

$$V = \frac{fl}{246}$$

where *f* is the resonant frequency, *l* is the length in feet, and *V* is the velocity factor. For long lines, greater than a quarter wavelength at the lowest frequency reached by the dipper (rarely the case in an amateur installation), a line terminated in an open circuit will be resonant at any odd multiple of a quarter wavelength. In this case, however, it may be necessary to check at several frequencies to be sure that one has the correct number of quarter wavelengths.

### *Miscellaneous*

Simply because it is an oscillator with a wide tuning range, the dipper can serve as a signal generator, both for approximate alignment of receivers and for supplying rf for antenna measurements.

A dipper can also act as a neutralization indicator. When the dipper is coupled to the plate tank of a transmitter stage, tuning the grid tank through resonance will cause a dip if the stage is not neutralized. This method has the advantage of not requiring that the transmitter be turned on to make the check. A dipper can also be used in the absorption-frequency-meter mode to detect rf in the plate circuit when the grid is driven.

A troublesome problem in transmitters, both tube and solid state, is that of parasitic oscillations. The dipper can often be most helpful in determining the frequency of the parasitics and in identifying the resonances responsible.

In the oscillating mode of the dipper, particularly if regeneration is adjustable, a dipper can serve as a field-strength indicator. A short antenna, about 20 inches of stiff wire connected to the hot end of the dipper coil will serve to pick up the signal.

A dipper, when its signal is picked up on a communications receiver, can be used to locate metal. As the coil is brought close to the metal, the dipper oscillator will be detuned, and the change in pitch of the beat note in the receiver will enable one to locate the metal easily. See Fig. 4.

Hidden wiring or metal piping can also be traced by inducing an rf current in the metal, and following it while monitoring the dipper's signal in a bc receiver. (see Fig. 5.) In this case, however, use the weakest signal that will give adequate response, both to secure more accurate locations and to cut down on the interference possibilities to broadcast reception in your neighborhood. Whenever a small bc receiver is used for tests like these, it is best to try the techniques first on a piece of wire you can see and become familiar with the method and the behavior of the equipment, as otherwise one can be misled by the avc action of the receiver.

*(Continued on page 33)*

Jan 1974 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

### *Testing and Operation*

The amplifier is unconditionally stable, with no parasitics. To verify this, a zero bias check for stability was made. This involved shorting out the Zener diode in the cathode return lead, reducing bias to essentially zero volts. Plate voltage was applied, allowing the tube to dissipate about 885 watts. The input and output circuits were then tuned through their ranges with no loads attached. There was no sign of output on the relative output meter and no change in the plate and grid currents. As with most cathode-driven amplifiers, there is a slight interaction between grid and plate currents during normal tune-up under rf-applied conditions. This should not be misconstrued as amplifier *instability*.

Tolerances of the Zener diode used in the cathode return line will result in values of bias voltage and idling plate currents other than those listed in Table I. The 1N3311, a 20-percent tolerance unit, is rated at 12 volts nominal but actually operates at 10 volts in this amplifier (within the 20-percent tolerance).

All testing and actual operation of this amplifier was conducted with a Raytrack high-voltage power supply used in conjunction with the authors 6-meter amplifier. The power supply control and output cable harness was moved from one amplifier to the other, depending on the desired frequency of operation.

Drive requirements were measured for plate power-input levels of 1000 and 1600 watts with a Bird model 43 Thru Line Wattmeter and a plug of known accuracy. Output power was measured simultaneously with drive requirements at the 1000 and 1600 watt plate power input levels. A second Bird model 43 with a 1000-watt plug was used to measure amplifier output into a Bird 1000-watt Termaline load. A 2500-watt plug would be necessary to determine output power at the 2-kW input level, so I stopped at the 1000-watt output point and worked backwards to calculate apparent stage gain and efficiency.

Efficiency measurements also were made employing the "tube air-stream heat-differential" method. Several runs were made at 885 watts static dc and normal rf input. Apparent efficiencies of 62 to 67 percent were noted. These values were about 5-percent higher than the actual power output values given in Table I. Both efficiency measurement schemes serve to confirm that the amplifier is operating at the upper limit of the theoretical 50-60-percent efficiency range for typical Class AB2 amplifiers.

To commence routine operation, the variable capacitor in the input circuit should be set at the point where lowest input VSWR was obtained during the "cold tube" initial tube-up. The ability of the plate tank to resonate at 144-145 MHz with the top cover in place should be verified with a grid-dip meter, via a one-turn link attached to the rf output connector. Top and bottom covers are then secured. As with all cathode driven amplifiers, excitation should never be applied when the tube heater is activated and plate voltage is removed.

Table I

*Performance Data*

| | | |
|---|---|---|
| Power input, watts | 1000 | 1600 |
| Plate voltage | 2600 | 2450 |
| Plate current (single tone) | 385 mA | 660 mA |
| Plate current (idling) | 50 mA | 50 mA |
| Grid bias | -10 V | -10 V |
| Grid current (single tone) | 35 mA | 54 mA |
| Drive power, watts | 18 | 41 |
| Efficiency (apparent) | 59.5 % | 61.8 % |
| Power gain (apparent) | 15.2 dB | 13.9 dB |
| Power output, watts | 595 | 1000 |

Next, turn on the tube heater and blower simultaneously, allowing 90 seconds for warm-up. Plate potential between 2400-3000 volts then may be applied and its presence verified on the multimeter. The power supply should be able to deliver 800 mA or so. With the VOX relay actuated, resting current should be indicated on the cathode meter. A small amount of drive is applied and the plate tank circuit tuned for an indication of maximum relative power output. The cathode circuit can now be resonated, tuning for minimum reflected power on the reflectometer, and not for maximum drive power transfer. Tuning and loading of the plate-tank circuit follows the standard sequence for any cathode driven amplifier. Resonance is accompanied by a moderate dip in plate/cathode current, a rise in grid current and a considerable increase in relative power output. Plate-current dip is not absolutely coincident with maximum power output but it is very close. Tuning and output-loading adjustments should be for maximum efficiency and output as indicated on the output meter. Final adjustment for lowest VSWR at amplifier input should be done when the desired plate input-power level has been reached.

### *Acknowledgements*

*I would like to express my gratitude to my* colleagues at MIT Lincoln Laboratory for their assistance in this project. Special thanks go to Ted Simmington, W1JOT; Lew Collins, K4GGI and Leo Wilber, W1MV, for their excellent comments on the construction portion of this article, and to Mr. Eino O. Gronroos, for his objective review of the technical manuscript. QST

#### References

5) Cool Amp Silver Plating Powder, part No. 1233-500, available from The Cool Amp Co., 8603 S. W. 17th Ave., Portland, OR 97219.

6) Tube socket grid clips, part No. 149-842, and Teflon chimney, part No. SK-2216, available from Eimac Division of Varian, 301 Industrial Way, San Carlos, CA 94070.

7) *The Radio Amateur's VHF Manual*, 2nd Edition, p. 288. (This Edition was erroneously labeled as the 11th at the time of printing.)

## The Art of Dipping

*(Continued from page 18)*

These twenty reasons are not all the possibilities, but seem to be among the most *important, and illustrate why a dipper is a handy* thing to have around the ham workshop. QST

# ARRL Periodicals Archive – Search Results

## A membership benefit of ARRL and the ARRL Technical Information Service

**ARRL Members:** You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

**Need a higher quality reprint or scan?** Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

**QST on CD-ROM**: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

**Non-Members:** Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.

### Copyright/Reprint Notice

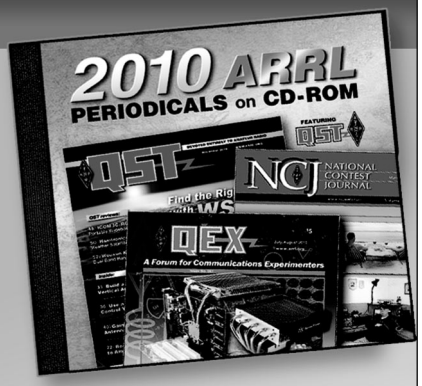
In general, all ARRL content is copyrighted. ARRL articles, pages, or documents--printed and online--are not in the public domain. Therefore, they may not be freely distributed or copied. Additionally, no part of this document may be copied, sold to third parties, or otherwise commercially exploited without the explicit prior written consent of ARRL. You cannot post this document to a Web site or otherwise distribute it to others through any electronic medium.

For permission to quote or reprint material from ARRL, send a request including the issue date, a description of the material requested, and a description of where you intend to use the reprinted material to the ARRL Editorial & Production Department: permission@arrl.org.

**QST Issue:** Jan 1987
**Title:** Tandem Match--An Accurate Directional Wattmeter, The
**Author:** John Grebenkemper, KA3BLO

Click Here to Report a Problem with this File

ADVERTISEMENT




**2010 ARRL Periodicals**
on CD-ROM

**ARRL's popular journals are available on a compact, fully-searchable CD-ROM. Every word and photo published throughout 2010 is included!**

- ***QST*** **The official membership journal of ARRL**
- ***NCJ*** **National Contest Journal**
- ***QEX*** **Forum for Communications Experimenters**

**SEARCH** the full text of every article by entering titles, call signs, names—almost any word. **SEE** every word, photo (including color images), drawing and table in technical and general-interest features, columns and product reviews, plus all advertisements. **PRINT** what you see, or copy it into other applications.

**System Requirements**: Microsoft Windows™ and Macintosh systems, using the industry standard Adobe® Acrobat® Reader® software. The Acrobat Reader is a free download at **www.adobe.com**.

**2010 ARRL Periodicals on CD-ROM**

ARRL Order No. 2001
**Only $24.95***
*plus shipping and handling

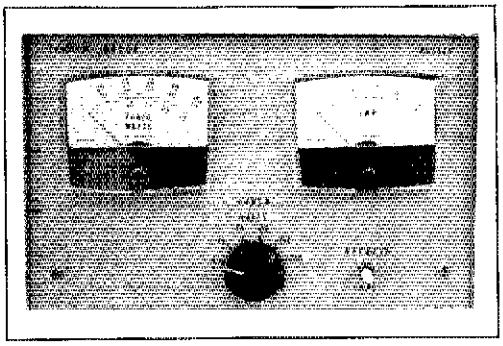
Additional sets available:
2009 Ed., ARRL Order No. 1486, $24.95
2008 Ed., ARRL Order No. 9406, $24.95
2007 Ed., ARRL Order No. 1204, $19.95
2006 Ed., ARRL Order No. 9841, $19.95
2005 Ed., ARRL Order No. 9574, $19.95
2004 Ed., ARRL Order No. 9396, $19.95
2003 Ed., ARRL Order No. 9124, $19.95
2002 Ed., ARRL Order No. 8802, $19.95
2001 Ed., ARRL Order No. 8632, $19.95

ARRL The national association for AMATEUR RADIO™
SHOP DIRECT *or call for a dealer near you.*
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)

Jan 1987 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

# The Tandem Match—An Accurate Directional Wattmeter

## This accurate directional wattmeter measures power from 1.5 to 1500 W and SWR over a range from 1:1 to 50:1.



By John Grebenkemper, KA3BLO
Tandem Computers, Inc
2550 Walsh Ave
Santa Clara, CA 94051

This article describes construction of a new type of directional wattmeter that can accurately measure power output and *standing wave ratio* (SWR) over a wide range of power. I originally started working on this project because I was unhappy with the performance of affordable directional wattmeters. The performance of the Tandem Match is as good as, or better than, any commercially available directional wattmeter designed for amateur use.

Most SWR meters are very inaccurate at low power levels because of non-linearities of the detector diodes. This design uses a compensating circuit to cancel the diode nonlinearity. A direct SWR readout provides a handy way of adjusting an antenna matching network for minimum SWR to the transmitter. There is no need for mental gymnastics to guess what the minimum SWR is in the event that transmitter output varies substantially with SWR.

### Design Principles

Directional wattmeters for Amateur Radio use consist of three basic elements: a directional coupler, a detector, and a signal processing and display circuit. The directional coupler separates the signal into its forward and reflected components. The ideal directional coupler would provide a signal proportional to the forward and reflected voltage independent of frequency, and could then be used to measure forward and reflected power over a wide frequency range. The best designs today will work over two decades of frequency. If the unit is to operate without being switched, the directional coupler must provide both the forward and reflected voltages simultaneously.

The detector circuit provides a dc output voltage proportional to the ac input voltage. Most directional wattmeters use a single germanium diode as the detector element. A germanium diode is used to minimize the diode nonlinearity at low power levels. The diode nonlinearity will still cause measurement errors in the SWR unless it is compensated before the display circuit. Most directional wattmeters do not work well at low power levels because of this diode nonlinearity.

The signal processing and display circuit computes and displays the SWR. There are a number of ways to perform this function. Meters that display only the forward and reflected power require the operator to compute the SWR separately. Many instruments require that the operator adjust the meter to a reference level while measuring forward power, then switch to measure reflected power with the SWR displayed on a special scale. Several units have been described that directly compute the SWR using analog signal processing circuits.[1-4]

Let's take a brief look at several popular circuits for accomplishing the above functions. I will also compare and describe the circuits used in the Tandem Match. The design specifications used in the Tandem Match are shown in Table 1.

### Circuit Description

*Directional Coupler Circuits*

A directional coupler consists of an input port, an output port and a coupled port. The device takes a portion of the power flowing from the input port to the output port and directs it to the coupled port, but *none* of the power flowing from the output port to the input port is directed to the coupled port. There are several terms that define the performance of a directional coupler

- *Insertion loss* is the amount of power that is lost from the signal flowing from the input port to the output port.
- *Coupling factor* is the amount of power (or voltage) that appears at the coupled port relative to the amount of power (or voltage) flowing from the input port to the output port.
- *Isolation* is the amount of power (or voltage) that appears at the coupled port relative to the amount of power (or voltage) flowing from the output port to the input port.
- *Directivity* is the isolation less the coupling factor.

All of these factors important in determining the performance of a directional wattmeter. Insertion loss should be minimized so that the coupler doesn't dissipate a significant amount of the transmitted power. The flatness of the coupling factor as a function of frequency determines how accurately the directional wattmeter can determine the forward and reflected power over a range of frequencies. Directivity will dictate the minimum SWR that can be measured. A directional coupler with 20-dB directivity will measure a 1:1 SWR as 1.22:1, but one with 30-dB directivity will measure a 1:1 SWR as 1.07:1.

The most commonly used directional coupler for Amateur Radio is the design originally described by Bruene in a *QST* article.[5] See Fig 1. I built a coupler of this type in one of my early attempts to make an accurate directional wattmeter. My measurements of its performance showed that over the range of 3 to 30 MHz, the coupling factor was flat to within ±1 dB, and the directivity was approximately

[1]Notes appear on page 26.

Jan 1987 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

**Table 1**
**Performance Specifications for the Tandem Match**

| | |
|---|---|
| Power range: | 1.5 to 1500 W. |
| Frequency range: | 1.8 to 54 MHz. |
| Power accuracy: | Better than ± 10% (± 0.4 dB). |
| SWR accuracy: | Better than ± 5%. |
| Minimum SWR: | Less than 1.05:1. |
| Power display: | Linear, suitable for use with either an analog or digital meter. |
| SWR display: | Linear, suitable for use with either an analog or digital meter. |
| Calibration: | Requires only an accurate voltmeter. No calibrated RF source is required. |

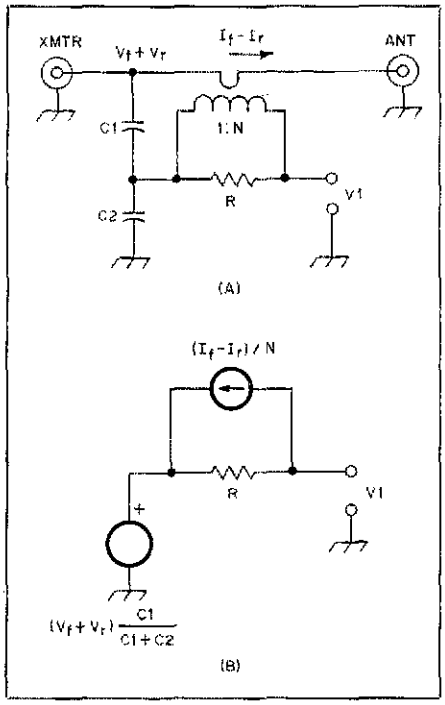



Fig 1—Simplified diagram of the Bruene directional coupler. At A, one of the unidirectional segments. At B, an equivalent circuit.

20 dB. Both of these factors limit the accuracy of this coupler, both for measuring power and determining low SWR. However, it is a relatively simple directional coupler, and it works well over a wide frequency range if high precision is not required.

The coupler used in the Tandem Match consists of a pair of toroidal transformers connected in tandem. This configuration has been used for years and has been described in several articles.[6,7] As long as the coupling factor is 20 dB or greater, this coupler can be used to separate both forward and reflected power. Fig 2 shows this coupler configuration.

I have made a number of measurements of the performance of this type of directional coupler. The particular configuration used in the Tandem Match works well over the frequency range of 1.8 to 54 MHz, with a nominal coupling factor of 30 dB. Over this range, the insertion loss is less than 0.1 dB. Figs 3 and 4 show plots of coupling factor and directivity as a function of frequency. Over the frequency range of 1.8 to 30 MHz the coupling factor is flat to within ± 0.1 dB, and increases to only ± 0.3 dB at 50 MHz. Directivity exceeds 35 dB from 1.8 to 30 MHz and exceeds 26 dB at 50 MHz.

The low-frequency limit of the directional coupler is determined by the inductance of the secondary windings of the transformers. Inductive reactance of the secondary windings should be greater than 150 Ω to avoid excessive insertion loss. The high-frequency limit of the directional coupler is determined by the length of wire used in winding the transformers. When this length approaches a significant fraction of a wavelength, the performance of the directional coupler deteriorates. A 20-dB coupling-factor version of this design should work up to 150 MHz because of the shorter length of wire required in the transformer secondaries.

*Detector Circuits*

Most amateur directional wattmeters use a single germanium diode as the detector. Germanium diodes are used because of their low forward voltage drop. However, this voltage drop is still significant, and an uncompensated diode detector will underestimate the detected power. A directional wattmeter that directly displays power will usually compensate for the diode nonlinearity by using a meter scale that takes these effects into consideration.

The effect of underestimating detected power becomes worse at lower power levels. Under these conditions, the ratio of the forward power to the reflected power will be overestimated because the reflected power is always less than the forward power. This results in an instrument that underestimates the SWR, particularly as the power level is reduced. A directional wattmeter can be checked for this effect by measuring the SWR at several power levels, as SWR should be independent of power level.

The Tandem Match uses a feedback circuit to compensate for diode nonlinearity. A simplified diagram of the detector is shown in Fig 5. When used with the 30-dB directional coupler, the output voltage of this circuit tracks the square root of the power over a range from 10 mW to 1.5 kW. A comparison between the

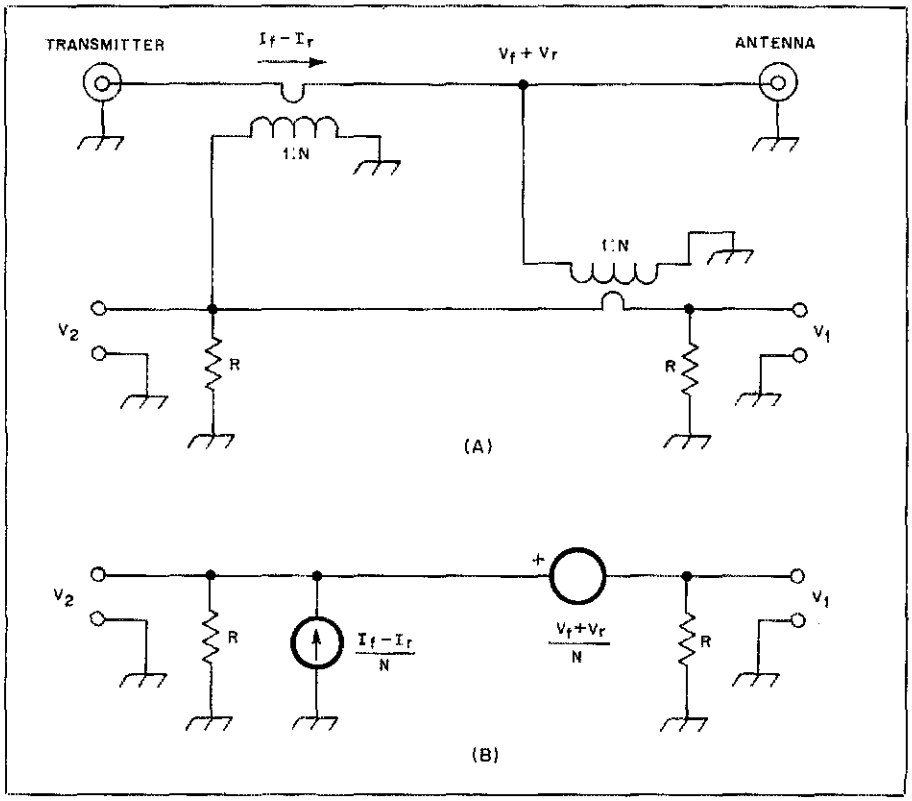



Fig 2—Simplified diagram of the Tandem Match directional coupler. At A, a schematic of the two transformers. At B, an equivalent circuit.

Jan 1987 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

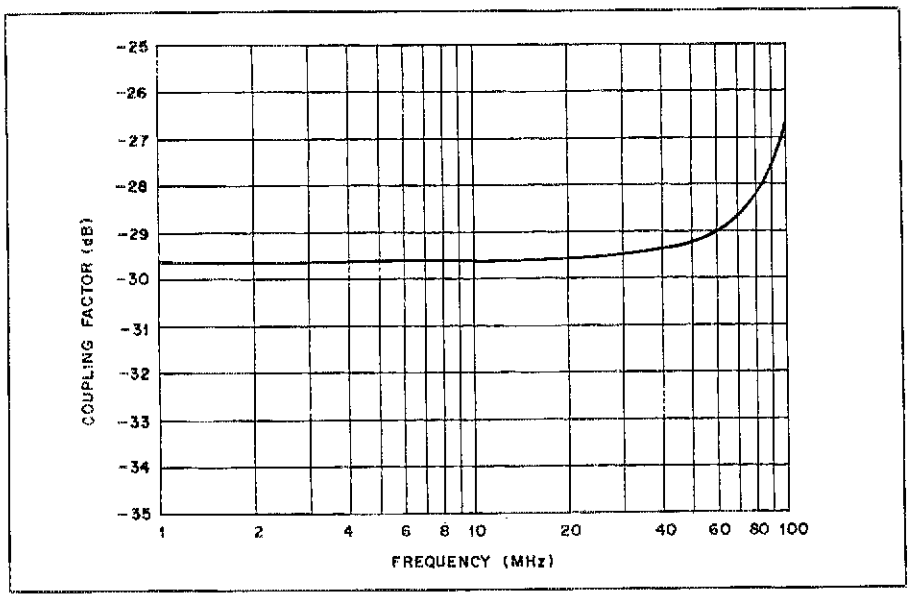



Fig 3—A plot of coupling factor for the Tandem Match directional coupler.

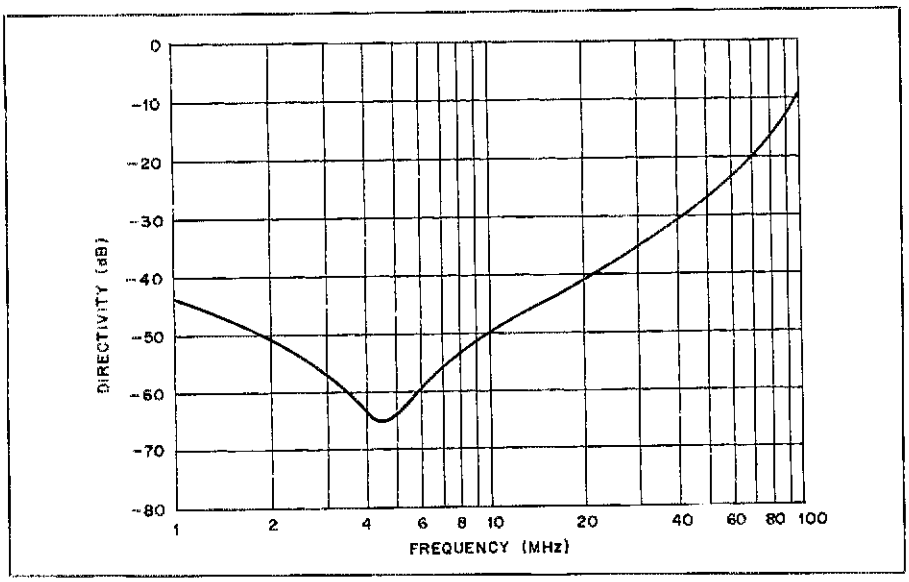



Fig 4—A plot of directivity for the Tandem Match directional coupler.

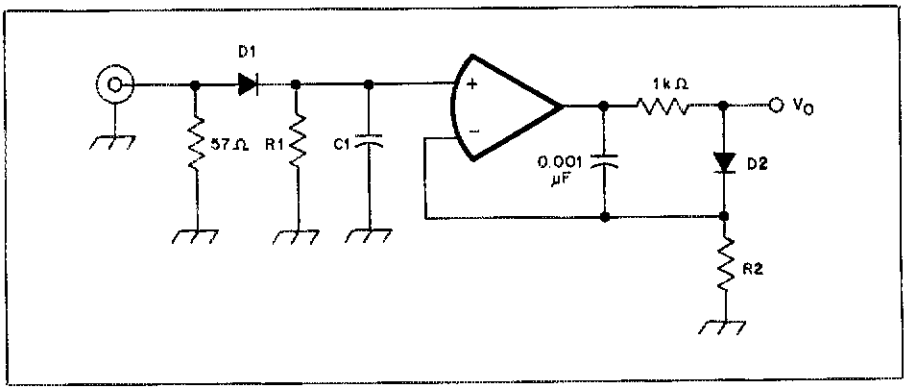



Fig 5—Simplified diagram of the detector circuit used in the Tandem Match. The output voltage, $V_o$ is approximately equal to the input voltage. D1 and D2 must be a matched pair. The op amp should have a low offset voltage (<1 mV), a low leakage current (<1 nA), and be stable over time and temperature. The resistor and capacitor in the feedback path assure that the op amp will be stable.

accuracy of this detector using Schottky barrier diodes and a detector using a germanium diode is shown in Fig 6. The compensated diode detector tracks the peak input voltage down to an input level of 30 mV, while the uncompensated diode detector shows significant errors at peak inputs of 1 V and less.

The circuit uses the voltage across the feedback diode D2 to compensate for the voltage drop across the detector diode, D1. The diodes must be a matched pair. The average current through D1 is determined by the detector-diode load resistor, R1. The peak current through this diode is several times larger than the average current; therefore, the current through D2 must be several times larger than the average current through D1 to compensate adequately for the peak voltage drop across D1. This is accomplished by making the feedback-diode load resistor, R2, several times smaller than R1. The voltage at the output of the compensated detector will be approximately the peak RF voltage at the input.

A computer simulation was used to optimize this ratio. The simulation assumed that the detector diodes were Schottky barrier types and that the detector-diode load resistor was 1 megohm. A ratio of R1 to R2 of 5:1 appeared to be near optimal. A larger ratio resulted in excessive peaking in the low power range, and a smaller ratio resulted in a faster roll off at the lower power levels. Experimental data tended to confirm these results.

### *Signal-Processing and Display Circuits*

The signal-processing circuitry calculates and displays the power measurement and SWR. A block diagram of the Tandem Match is shown in Fig 7. This circuit differs from conventional directional wattmeters in several respects. Most directional wattmeters display the forward power. This measurement is very close to the power delivered into the transmission line as long as the SWR is low. However, as the SWR increases, the forward power becomes an increasingly poor measure of the power being delivered to the load. At an SWR of 3:1, a forward power reading of 100 W implies that only 75 W is being delivered to the load, assuming that the transmission line has no losses. (The reflected power reading will be 25 W.) The Tandem Match directly displays the power being delivered *into* the transmission line. This is the quantity which must be optimized to result in maximum radiated power.

The Tandem Match also directly computes and displays the SWR of the transmission line on a linear scale. The direct display of SWR provides a convenient method of adjusting an antenna matching network. The antenna matching network is adjusted to minimize the displayed SWR. The displayed SWR is not affected by changes in the transmitter output power and the antenna system can be

Jan 1987 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

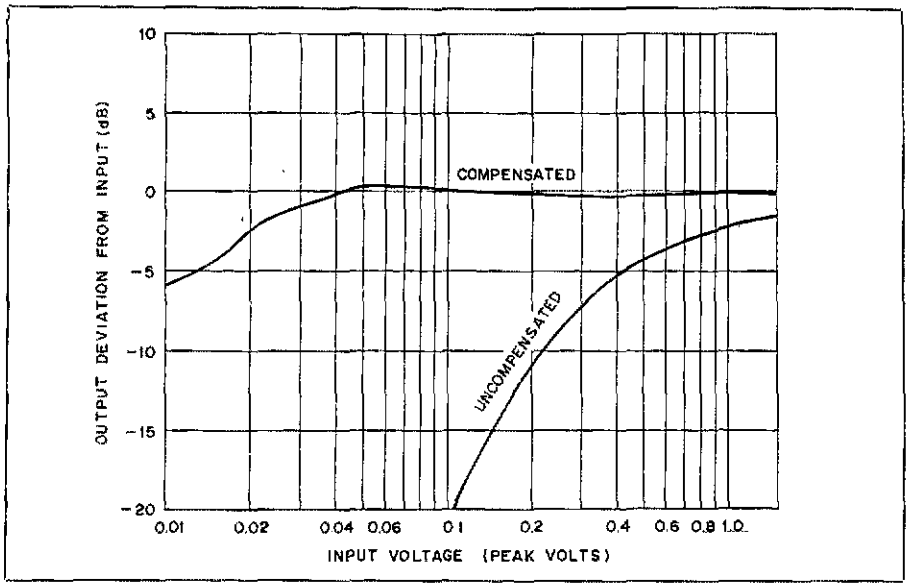



Fig 6—Comparison of accuracy for a compensated and uncompensated diode detector. The top line shows measurement error as a function of peak RF input voltage for the compensated detector using a matched pair of Schottky barrier diodes. The bottom line shows measurement error for an uncompensated 1N34A germanium diode detector. Measurements were made at a frequency of 10 MHz.

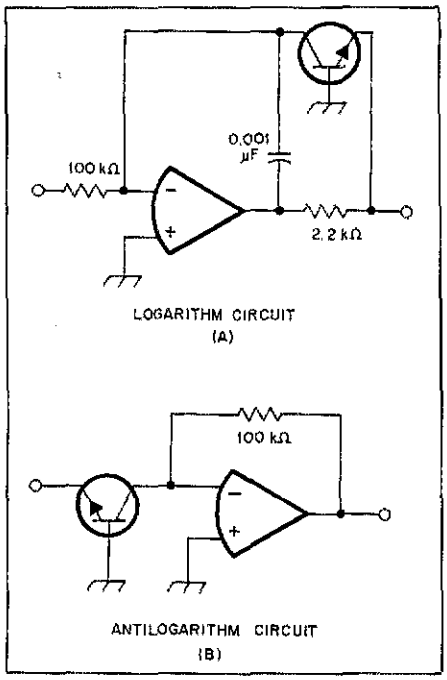



Fig 8—Simplified diagrams of the logarithm circuit at A and the antilogarithm circuit at B.

tuned with an output power of only a few watts. People are amazed at how easy it is to tune an antenna system using the direct SWR display from the Tandem Match.

The heart of the signal-processing circuit is the analog logarithm and antilogarithm circuits shown in Fig 8. The circuit is based on the fact that the collector current in a silicon transistor is proportional to the exponential (antilogarithm) of its base-emitter voltage over a range of collector currents from a few nanoamperes to a few milliamperes when the collector base voltage is zero.[3] Variations of this circuit are used in the squaring circuits to convert voltage to power and in the divider circuit used to compute the SWR. With good op amps, this circuit will work well for input voltages of less than 100 mV to greater than 10 V.

## Construction

Up to this point, we have examined some of the circuitry of the Tandem Match. The

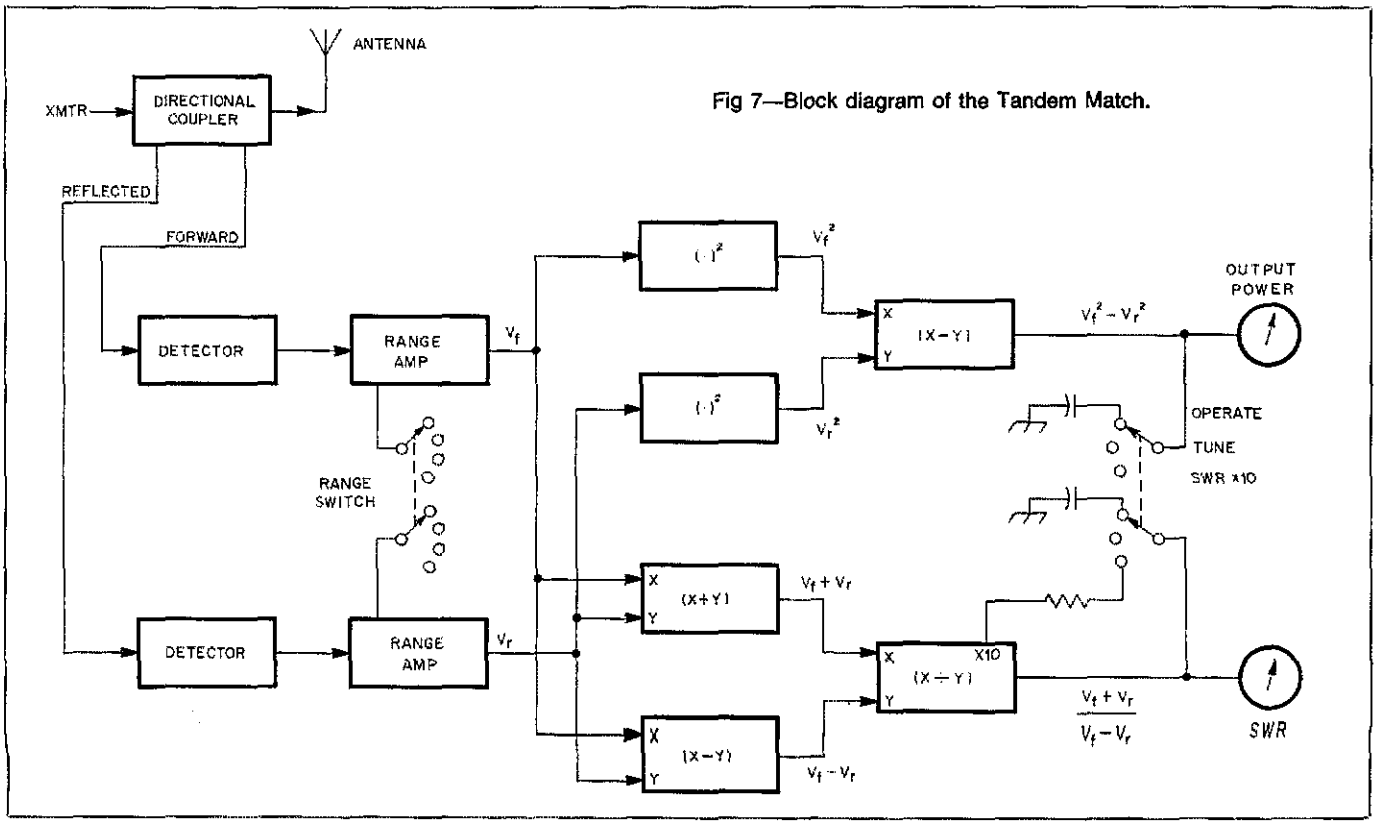



Fig 7—Block diagram of the Tandem Match.

Jan 1987 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

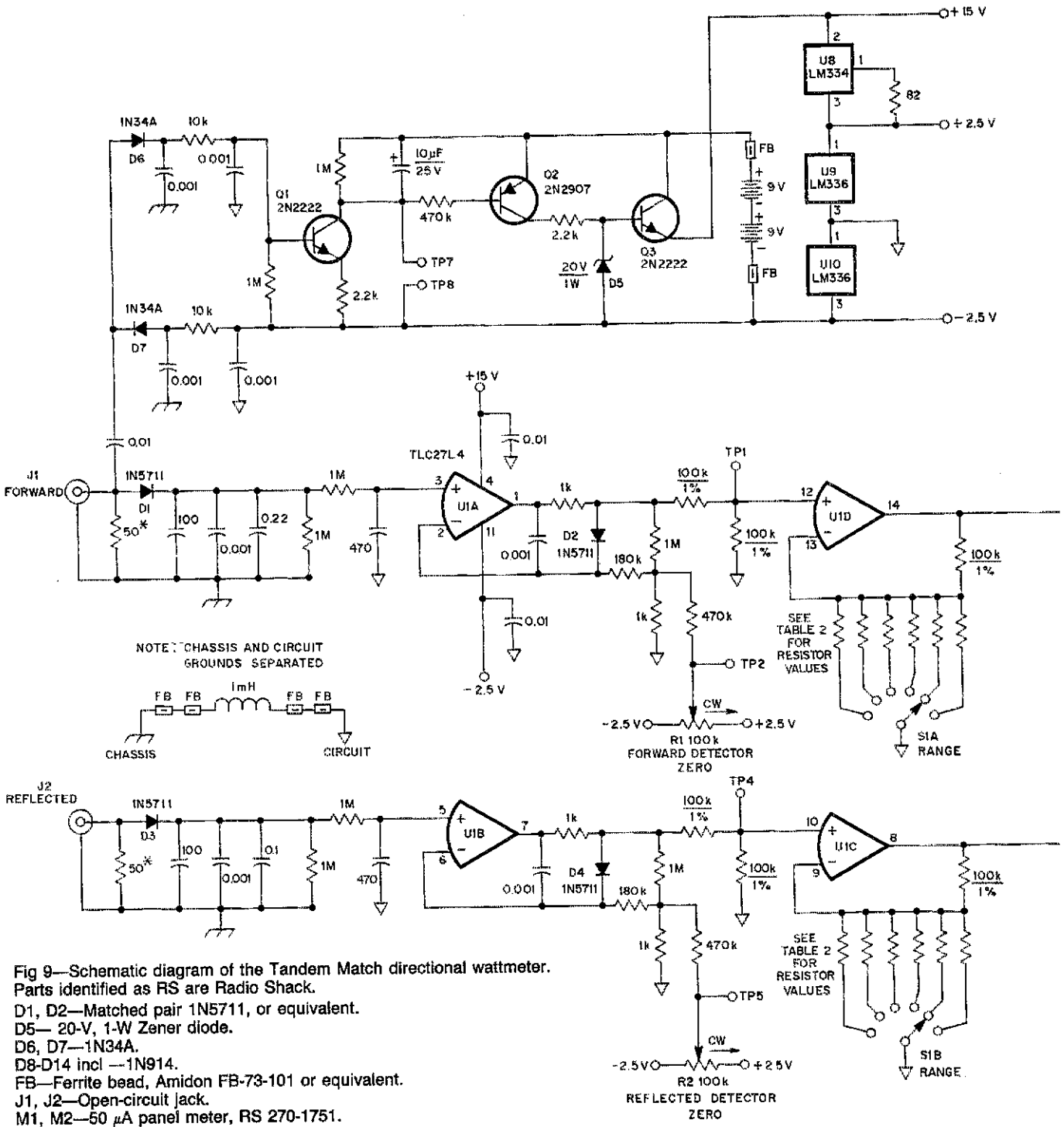
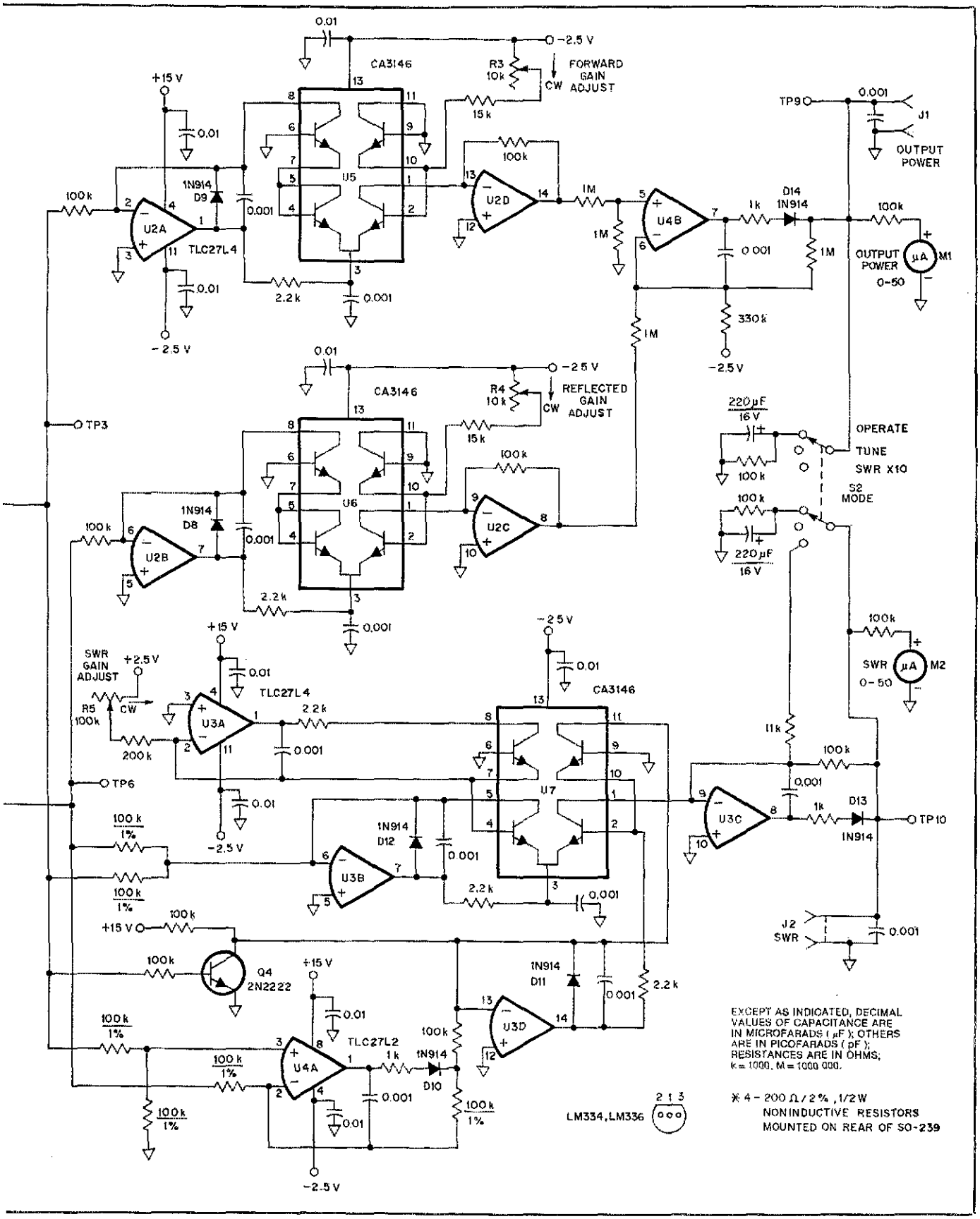



Fig 9—Schematic diagram of the Tandem Match directional wattmeter. Parts identified as RS are Radio Shack.
D1, D2—Matched pair 1N5711, or equivalent.
D5— 20-V, 1-W Zener diode.
D6, D7—1N34A.
D8-D14 incl —1N914.
FB—Ferrite bead, Amidon FB-73-101 or equivalent.
J1, J2—Open-circuit jack.
M1, M2—50 µA panel meter, RS 270-1751.
Q1, Q3, Q4—2N2222 or equivalent.
Q2—2N2907 or equivalent.
R1, R2, R5—100 kΩ, 10-turn cermet Trimpot®.
R3, R4—10 kΩ, 10-turn cermet Trimpot.
U1-U3 incl.—TLC27L4 or TLC27M4, RS 276-1750.
U4—TLC27L2 or TLC27M2, RS 276-1749.
U5-U7 incl —CA3146.
U8—LM334.
U9, U10—LM336.

rest of this article describes the construction and calibration of the unit. The schematic diagram for the Tandem Match is shown in Fig 9. The finished unit is capable of measuring power and SWR as accurately as any commercial directional wattmeter, yet requires only an accurate voltmeter for calibration. I have constructed several versions of both the directional coupler and the detector circuit. They have all performed within their specifications.

The Tandem Match is designed to oper-

Jan 1987 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved




ate from batteries and draw very little power. Much of the circuitry is of high impedance, and care must be taken to isolate it from RF fields. The unit should be housed in a metal case. Most of the problems in the initial unit were caused by RF getting into the op amp circuitry.

*Directional Coupler*

The directional coupler is constructed in its own small, (2¾ × 2¾ × 2¼-inch) aluminum box (see Fig 10). Two pairs of SO-239 connectors are mounted on opposite sides of the box. A piece of copper-clad PC board is run diagonally

Jan 1987 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

across the box to improve the directivity of the coupler. The RG-8X coaxial cable passes through holes in the PC board. The center conductor of the coaxial cable is connected to the center pin of the SO-239 connectors. The coaxial-cable shield is connected only at one end; the other end is open. This is very important—the shield must be connected only at one end or the directional coupler circuit will not work properly. The shield forms an electrostatic shield between the two sides of the transformer. Each transformer consists of 31 turns of no. 24 enamel wire on an Amidon T50-3 iron-powder core. The core is then slipped over the coaxial cable (including both the shield and the outer insulation.) The center conductor forms the primary of the transformer. The transformer is then connected as shown in Fig 10 with the wires running through separate holes in the copper-clad PC board. The secondary leads are twisted together and run to the SO-239 connector on the opposite side.

The directional coupler can be separated from the rest of the circuitry, if desired. In this event, run two coaxial cables from the directional coupler to the detector inputs to carry the forward and reflected power.

I have not used this directional coupler at power levels in excess of 100 W output and don't know if higher power levels will cause the enamel insulation to break down. If this is a problem, Teflon® insulated wire should work at power levels up to the legal limit. The wire size is not important, but the wire diameter, including insulation, must not exceed the size of no. 24 enamel-covered wire. I constructed one directional coupler using no. 28 Teflon-insulated wire, and it worked identically to one constructed with enameled wire.

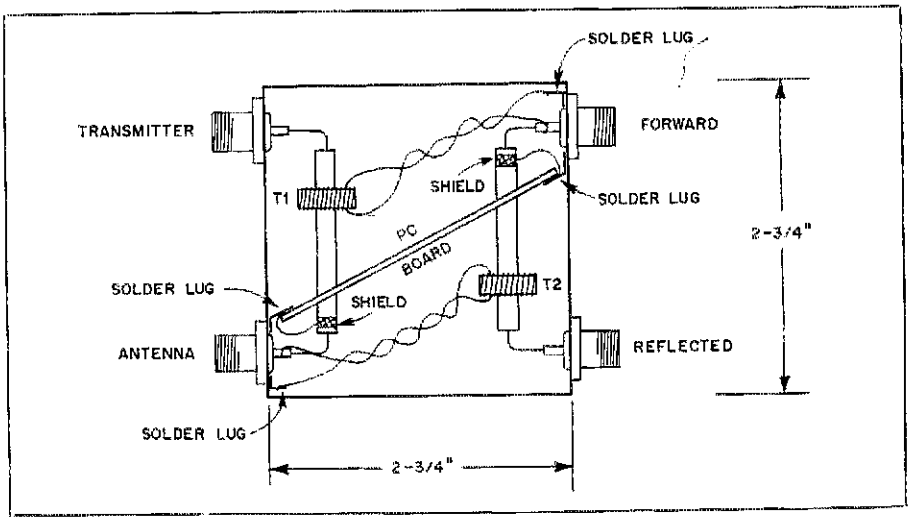



Fig 10—Construction details for the directional coupler.

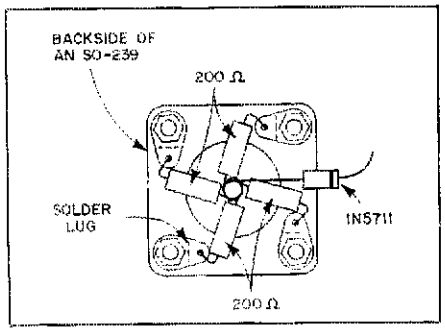



Fig 11—Mounting of parallel load resistors on SO-239 connector. Four 200-ohm resistors are mounted in parallel to provide a 50-ohm detector load.

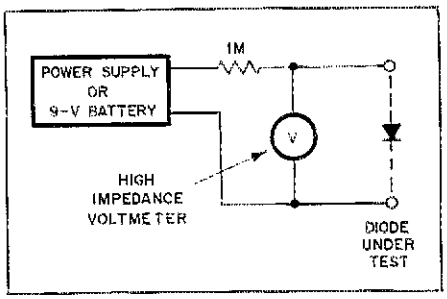



Fig 12—Diode matching test setup.

*Detector and Signal-Processing Circuits*

The detector and signal-processing circuits were constructed on a piece of copper-clad Vectorbord® . These circuits use two separate grounds. It is extremely important that these grounds be isolated as shown in the circuit diagram. Failure to do this will probably result in RF disrupting the circuit operation because of RF currents flowing on the outside shield of the coaxial cable. The separate grounds prevent this current from getting into the op amp circuitry.

The directional coupler requires good 50-ohm loads. Each load is constructed on the back of a female UHF chassis connector. The load consists of four 200-ohm resistors connected from the center conductor of the UHF connector to each of the four holes on the mounting flange, as shown in Fig 11. The detector diode is then run from the center conductor of the connector to the 100-pF and 1000-pF bypass capacitors mounted right next to the connector. This load and detector combination has been measured to be flat to beyond 500 MHz.

The detector diodes must be matched. This can be done at dc using the circuit shown in Fig 12. The meter must have a high input impedance, preferably 10 megohms or greater. Diodes which have a voltage drop within a few millivolts of each other can be considered matched. I found that diodes from the same batch were already matched.

I used 1N5711 Schottky barrier diodes because they were available. Any RF detector diode with a low forward voltage drop could be used. The important parameters are that the diodes must be a matched pair, the forward voltage drop of the diodes should be less than 300 mV, and the reverse breakdown voltage of the diodes must be greater than 30 V.

Germanium diodes could be used in this circuit, but there will be a degradation in performance. If germanium diodes are used, the resistance values for the detector-diode and feedback-diode load resistors should be reduced by a factor of 10.

The rest of the circuit layout is not critical. However, the lead lengths of the 0.001- and 0.01-μF bypass capacitors should be kept short. The capacitors provide additional bypassing for the op amp circuitry.

The 1N34A diodes form a voltage doubler which is used to detect the presence of a carrier. When forward power exceeds 1.5 W, Q3 is turned on and supplies power to the directional coupler. Power is held on for about 10 seconds after the carrier is last detected. Connecting TP7 to TP8 will force the unit on, even with no carrier present. Regulated references of +2.5 V and −2.5 V are generated by the LM334 and LM336. Using Zener diodes instead of these will significantly degrade the stability and accuracy of the directional wattmeter.

The four op amps in U1 compensate for the nonlinearity of the detector diodes. D1 and D2 must be a matched pair, as must D3 and D4. A RANGE switch selects the power range. The resistor values for the RANGE switch are shown in Table 2. I used a six-position switch because I had one handy. Full-scale input power gives an output at U1C or U1D of 7.07 V. The forward and reflected power detectors are zeroed with R1 and R2.

Forward and reflected voltages are squared by U2, U5 and U6 so that the output voltages are proportional to forward and reflected power. The gain constants are adjusted using R3 and R4 so that an input of 7.07 V to the squaring circuit gives an output of 5 V. The difference between these two voltages is used by U4B to yield an output that is proportional to the power deli-

Jan 1987 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

**Table 2**
**Range-Switch Resistor Values**

| *Full-Scale Power Level (Watts)* | *Range Resistor (1% Accuracy) (kilohms)* |
|---|---|
| 1 | 2.32 |
| 2 | 3.24 |
| 3 | 4.02 |
| 5 | 5.23 |
| 10 | 7.68 |
| 15 | 9.53 |
| 20 | 11.0 |
| 25 | 12.7 |
| 30 | 14.0 |
| 50 | 18.7 |
| 100 | 28.7 |
| 150 | 37.4 |
| 200 | 46.4 |
| 250 | 54.9 |
| 300 | 63.4 |
| 500 | 100.0 |
| 1000 | 237.0 |
| 1500 | 649.0 |
| 2000 | open |

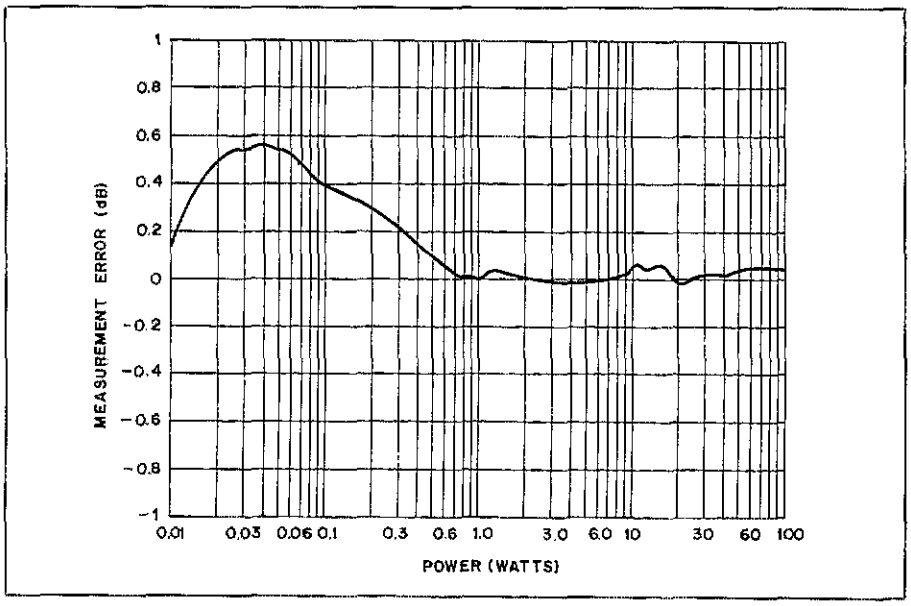



Fig 13—Power measurement accuracy for the Tandem Match. Measurement errors exist primarily in the low-power range.

vered into the transmission line. This voltage may be peak detected to hold and indicate the maximum power that occurs during CW or SSB transmissions.

Computation of SWR from the forward and reflected voltages is accomplished by U3, U4 and U7. When no carrier is present, Q4 forces the SWR reading to zero. This circuit considers no carrier present if the forward power is less than 1 to 2% of the full-scale setting of the RANGE switch. The SWR computation circuit gain is adjusted by R5. The SWR output varies from 1 V for an SWR of 1:1 to 5 V for an SWR of 5:1. The output may be peak detected to hold and indicate the maximum SWR reading during CW or SSB transmissions.

Transistor arrays (U5, U6 and U7) are used for the logarithm and antilogarithm circuits. Using transistor arrays guarantees that the transistors will be well matched. Individual transistors may be used, but the accuracy may be significantly degraded.

A three-position toggle switch is used to select the three operating modes. In the OPERATE mode, the power and SWR outputs are peak detected and held for a few seconds to allow the meters to be read during actual transmissions. In the TUNE mode, the meters display the instantaneous output power and SWR. This mode is used to adjust an antenna matching network. In the SWR × 10 mode, the SWR full-scale reading is increased to 50:1. This mode is handy for adjusting an antenna matching network when one has no idea what the correct setting should be.

An external meter can be used to read both power and SWR values. The output power range is 0 to 5 V, with 0 V representing no output power and 5 V representing the RANGE switch full-scale value. Output voltages above 5 V may be incorrect because of voltage limiting in some of the op amp circuits. The SWR range is also from 0 to 5 V, with 0 V representing no power present, 1 V representing an SWR of 1:1 and 5 V representing an SWR of 5:1. I have used a digital voltmeter to obtain more precise readings than possible with the analog meters. Alternatively, one could incorporate digital panel meters into this unit in place of the analog meters shown in the circuit diagram.

### Calibration

The directional wattmeter can be calibrated to its full accuracy using only an accurate voltmeter. A digital voltmeter is preferred, but a good analog meter should also work. All calibration is done with dc voltages. The directional coupler and detector circuits are inherently accurate if they were correctly built. To calibrate the wattmeter, use the following procedure:

1) Set the mode switch to TUNE and the RANGE switch to 100 W full scale or less.

2) Jumper TP7 to TP8. This turns the unit on.

3) Jumper TP1 to TP2. Adjust R1 for 0 V at TP3.

4) Jumper TP4 to TP5. Adjust R2 for 0 V at TP6.

5) Adjust R1 for 7.07 V at TP3.

6) Adjust R3 for 5.00 V at TP9 or full scale reading on M1.

7) Adjust R2 for 7.07 V at TP6.

8) Adjust R4 for 0 V at TP9 or zero reading on M1.

9) Adjust R2 for 4.71 V at TP6.

10) Adjust R5 for 5.00 V at TP10 or full scale reading on M2.

11) Set RANGE switch to most sensitive scale.

12) Remove jumpers from TP1 to TP2 and TP4 to TP5.

13) Adjust R1 for 0 V at TP3.

14) Adjust R2 for 0 V at TP6.

15) Remove jumper from TP7 to TP8.

This completes the calibration procedure. I have found that this calibration is as accurate as using expensive laboratory test equipment. The directional wattmeter should now be ready for use.

### Accuracy

Some writers claim that their new gadget performs miracles, but there is seldom any proof to back up their claims. For this article, I was determined to demonstrate that the Tandem Match meets its accuracy specifications when tested with calibrated laboratory test equipment.

I compared the performance of this wattmeter against other well-known directional couplers and laboratory test equipment. The Tandem Match is as good as any amateur directional wattmeter I have tested. I checked the power measurement accuracy against a Hewlett-Packard HP-436A power meter. The HP meter has a specified measurement error of less than ± 0.05 dB. The Tandem Match tracked the HP meter to within ± 0.5 dB over a power range of 10 mW to 100 W, and to within ± 0.1 dB over a power range of 1 W to 100 W. Fig 13 shows the measured performance. The unit was tested only to 100 W because I do not have a transmitter with a higher power rating.

The performance in measuring SWR was equally good. Using the HP power meter and a calibrated directional coupler to compute the SWR, the Tandem Match tracked the test equipment to within ± 5%

Jan 1987 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

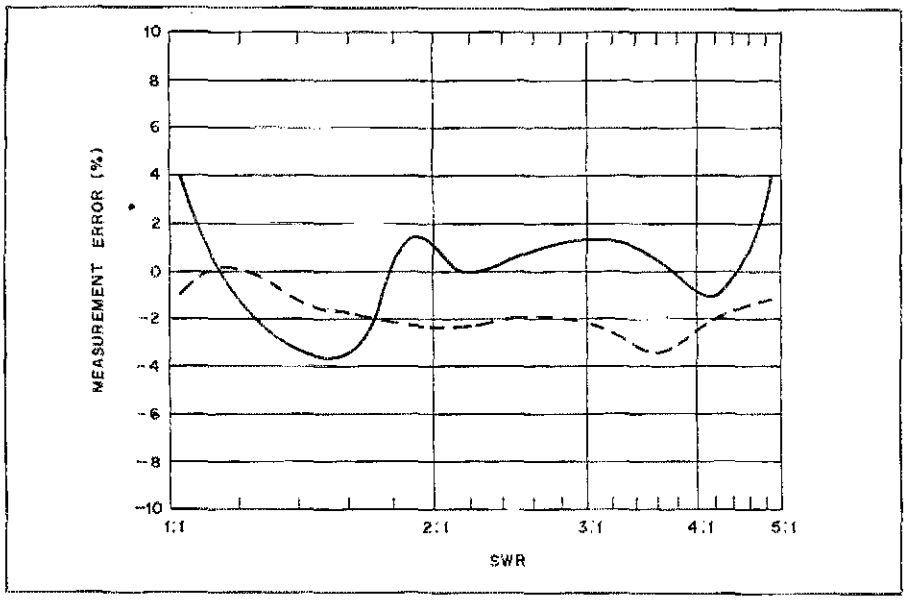



*Fig 14—SWR measurement accuracy for the Tandem Match over the range of SWR values from 1:1 to 5:1. Measurements were made at 10 MHz. The dashed line was measured at 8 W forward power; the solid line was measured at 100 W forward power.*

for SWR from 1:1 to 5:1. These measurements were made at power levels of 8 W and 100 W. Fig 14 reflects these results.

### Operation

Operation of the Tandem Match is straightforward. The unit is connected in the 50-ohm line between the transmitter and the antenna matching network, or antenna if no matching network is used. The RANGE switch should be set for a range greater than the output rating of the transmitter, and the mode switch should be set to TUNE. When the transmitter is keyed, the Tandem Match will automatically turn itself on and measure the power delivered to the antenna and the SWR on the transmission line. If no carrier is present, the OUTPUT POWER and SWR meters will indicate zero.

The OPERATE mode is used to measure the peak power delivered to the antenna and the SWR when CW or SSB is transmitted. The peak detectors in the unit are not ideal, and there could be about a 10% variation from the actual peak power and SWR. However, this mode is convenient for monitoring these parameters during actual transmissions.

The SWR × 10 mode is used to increase the SWR meter range to 50:1. This mode is sometimes necessary for initially adjusting a matching network if the SWR is very high. This range should be sufficient to cover any SWR value that occurs in amateur use. A 50-foot length of RG-8 that is left open circuited will display a measured SWR of 43:1 at 3.5 MHz exclusively because of the loss in the cable. Higher frequencies and longer cables will exhibit a smaller maximum SWR.

It is easy to use the Tandem Match to adjust a matching network. Adjust the transmitter for minimum output power, as long as it exceeds 1.5 W. With the carrier on and the mode switch set to TUNE or SWR × 10, adjust the matching network for minimum displayed SWR. The process is similar to dipping the plate current on a tube-type transmitter. Once the minimum SWR is obtained, the transmitter can be set to the proper operating mode and output power with the Tandem Match set to the OPERATE mode. The peak output power and SWR can then be monitored during operation.

### Design Variations

There are several ways in which this design could be changed to enhance operation. The most important would be to add VHF/UHF operation capability. This can be accomplished by designing a new directional coupler that will work above the HF range. The existing detector circuit will work to at least 500 MHz.

For those who desire a low-power directional wattmeter, the directional coupler can be changed to a 20-dB coupling factor by decreasing the turns ratio of the transformers to 10:1. A low-power unit should be capable of measuring output power from 1 mW to about 150 W, and the unit should switch on at about 150 mW. This change should also increase the maximum operating frequency to about 150 MHz. If operation on 160 meters is desired, it may be necessary to change the toroidal core material to obtain sufficient reactance at 1.8 MHz to minimize the insertion loss.

The unit can be made to operate with coaxial cable having characteristic impedances other than 50 ohms. It will be necessary to change the detector terminating resistors to result in the value of the characteristic impedance of the coaxial cable used. Because the detector circuit measures voltage, not power, the range resistors must also be changed to obtain the proper power levels.

The detector circuitry can be used to directly measure low RF power levels in 50-ohm circuits. Power is fed directly to the forward detector, and the power is read from the POWER OUTPUT meter. Over the power range from 10 μW to 1.5 W, the detector is quite linear.

### Acknowledgments

I would like to acknowledge the contribution of members of the Tandem Radio Amateur's Club who gave this unit a severe operational test during the 1985 Field Day and made several significant suggestions for improvements. I would also like to thank Mike Brown, WC6S, and Joe Castellano, W6SNV, for reviewing the manuscript and suggesting several clarifications.

### Notes

[1]D. L. Fayman, "A Simple Computing SWR Meter," *QST*, Jul 1973, pp 23-33.
[2]H. Perras, "Broadband Power-Tracking VSWR Bridge," *Ham Radio*, May 1979.
[3]V. G. Leenerts, "Automatic VSWR and Power Meter," *Ham Radio*, May 1980.
[4]A. L. Bailey, "The Antenna Lab, Parts 1 and 2," *Radio Communication*, Aug and Sep 1983.
[5]W. B. Bruene, "An Inside Picture of Directional Wattmeters," *QST*, Apr 1959.
[6]See note 2.
[7]W. M. Spaulding, "A Broadband Two-Port S-Parameter Test Set," *Hewlett-Packard Journal*, Nov 1984.
[8]J. F. Gibbons and H. S. Horn, "A Circuit With Logarithmic Response Over Nine Decades," *IEEE Transactions on Circuit Theory*, Vol CT-11, No. 3, pp 378-384, Sep 1984.

QEX

**QEX: THE ARRL EXPERIMENTERS' EXCHANGE AND AMSAT SATELLITE JOURNAL**

In what one publication can you find information on spread spectrum, packet radio and tips on testing a VHF/UHF antenna? *QEX* is the answer! *QEX* means Amateur Radio operators designing and experimenting with circuits and equipment to further the Amateur Radio frontier.

The December issue includes articles on:
- "Experimenting With Direct-Sequence Spread Spectrum," by Andre Kesteloot, N4ICK
- "HF Packet: Comments and Tips," by Gregory P. Latta, AA8V
- "A Two-Tone Generator With Very Low Distortion," by George Pelissier, F1BBI
- "Xerox 820-1 Compendium—Part 5," by AMRAD

Other features include: information on inductance and capacitance magnification, and the four basic components needed to test a VHF/UHF antenna.

*QEX* is edited by Paul Rinaldo, W4RI, and Maureen Thompson, KA1DYZ, and is published monthly. The special subscription rate for ARRL members is $6 for 12 issues; for nonmembers, $12. There are additional postage surcharges for mailing outside the US; write to Headquarters for details.

# Two-Tone Audio Generator



Exterior view of the two-tone audio generator.

## A TWO-TONE AUDIO GENERATOR

The audio frequency generator shown above makes a very convenient signal source for testing the linearity of a single-sideband transmitter. To be suitable for transmitter evaluation, a generator of this type must produce two non-harmonically related tones of equal amplitude. The level of harmonic and intermodulation distortion must be sufficiently low so as not to confuse the measurement. The frequencies used in this generator are 700 and 1900 Hz, both well inside the normal audio passband of an SSB transmitter. Spectral analysis and practical application with many different transmitters has shown this generator to meet all of the requirements mentioned above. While designed specifically for transmitter testing it is also useful any time a fixed-frequency, low-level audio tone is needed. Details on distortion measurement and the two-tone test can be found in Chapter 18.

**Circuit Details**

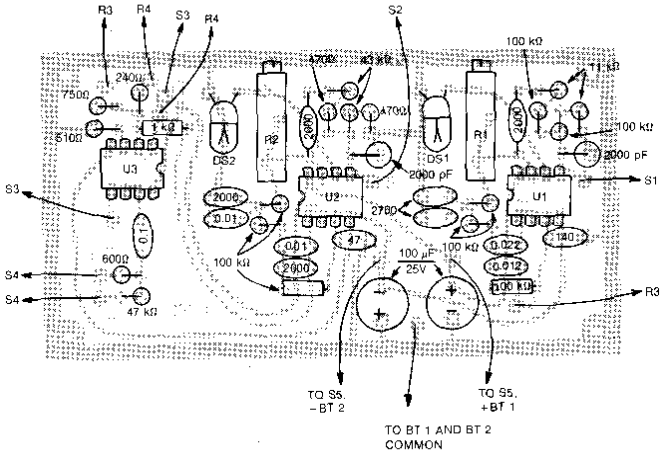
Each of the two tones is generated by a separate Wein bridge oscillator, U1B and U2B. The oscillators are followed by RC active low-pass filters, U1A and U2A. Because the filters require nonstandard capacitor values, provisions have been made on the circuit board for placing two capacitors in parallel in those cases where standard values cannot be used. The oscillator and filter capacitors should be polystyrene or Mylar film types if available. Two tones are combined at op amp U3A. This amplifier has a variable resistor, R4, in its feedback loop which serves as the output LEVEL control. While R4 varies both tones together, R3, the BALANCE control, allows the level of tone A to be changed without affecting the level of tone B. This is necessary because some transmitters do not have equal audio response at both frequencies. Following the summing amplifier is a step attenuator; S3 controls the output level in 10-dB steps. The use of two output level controls, R4 and S3, allows the output to cover a wide range and still be easy to set to a specific level.

The remaining op amp, U3B is connected as a voltage follower and serves to buffer the output while providing a high-impedance load for the step attenuator. Either high or low output impedance can be selected by S4. The values shown are suitable for most transmitters using either high- or low- impedance microphones.

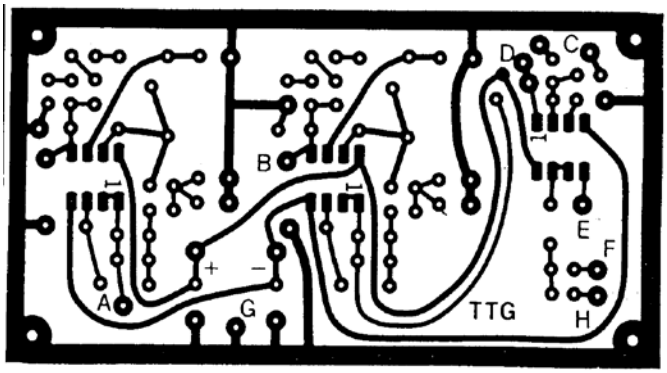
**Construction and Adjustment**

Component layout and wiring are not critical, and any type of construction can be used with good results. For those who wish to use a printed-circuit board, a parts placement guide is shown on the next page. Because the generator will normally be used near a transmitter, it should be enclosed in some type of metal case for shielding. Battery power was chosen to reduce the possibility of RF entering the unit through the ac line. With careful shielding and filtering, the builder should be able to use an ac power supply in place of the batteries.

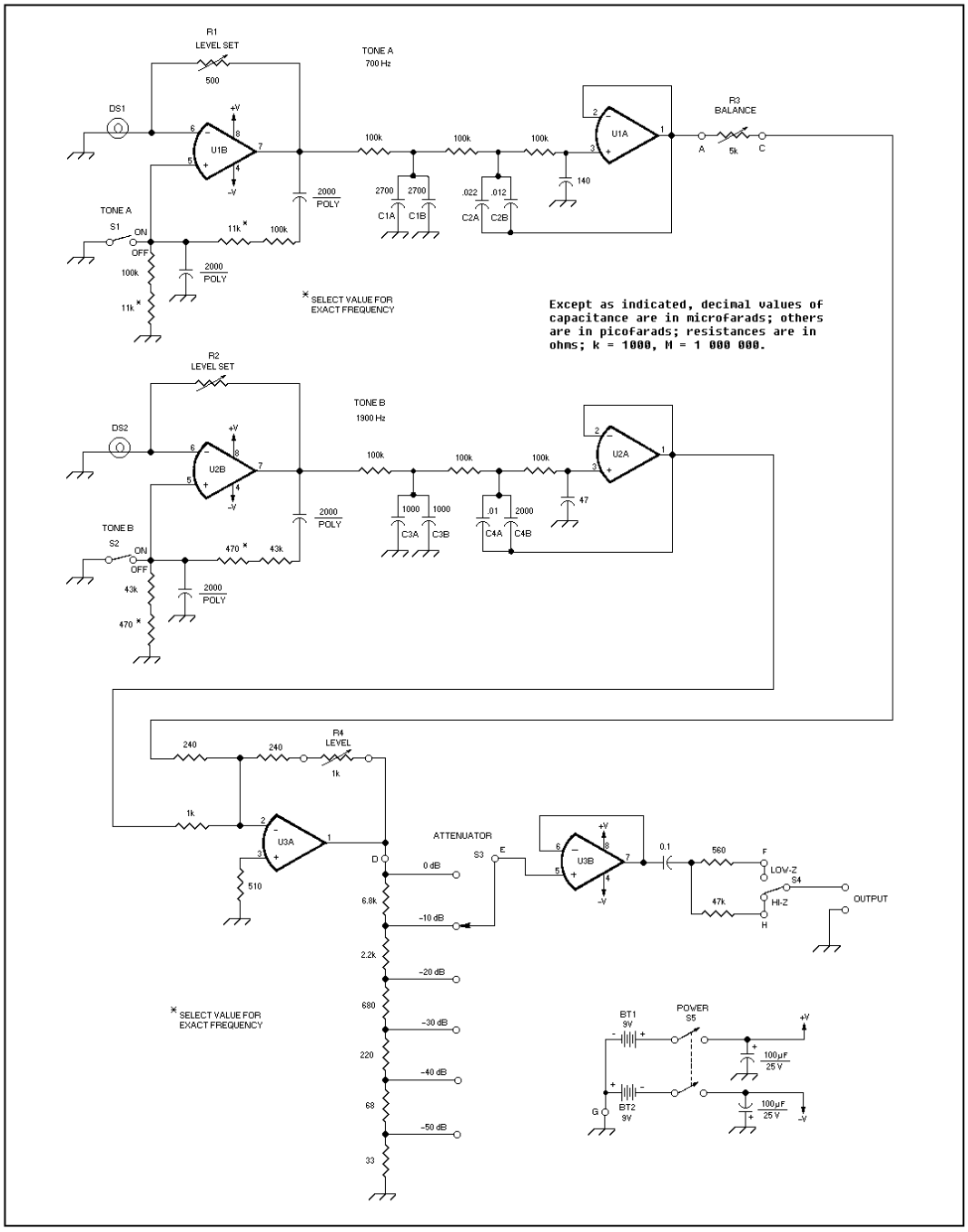
The only adjustment required before use is the setting of the oscillator feedback trimmers, R1 and R2. These should be set so that the output of each oscillator, measured at pin 7 of U1 and U2, is about 0.5 volt RMS. A VTVM or oscilloscope can be used for this measurement. If neither of these is available, the feedback should be adjusted to the minimum level that allows the oscillators to start reliably and stabilize quickly. When the oscillators are first turned on, they take a few seconds before they will have stable output amplitude. This is caused by the lamps, DS1 and DS2, used in the oscillator feedback circuit. This is normal and should cause no difficulty. The connection to the transmitter should be through a shielded cable.




Parts-placement diagram for the two-tone audio generator, shown from the component side of the board.




Printed-circuit board etching pattern, shown from the foil side of the board.




Schematic diagram of the two-tone audio generator. All resistors are 1/4-W carbon film.

BT1, BT2--9V Alkaline.
C1A, B--Total capacitance of 0.0054 µF, ±5%
C2A, B--Total capacitance of 0.034 µF, ±5%
C3A, B--Total capacitance of 0.002 µF, ±5%
C4A, B--Total capacitance of 0.012 µF, ±5%
DS1, DS2--12V, 25mA lamp.
R1, R2--500Ω, 10-turn trim potentiometer.
R3--500Ω, panel-mount potentiometer.
R4--1kΩ, panel-mount potentiometer.
S1, S2--SPST toggle switch.
S3--Single-pole, 6-position rotary switch.
S4--SPDT toggle switch.
S5--DPDT toggle switch.
U1, U2, U3--Dual JFET op amp, type LF353N or TL082.

# ARRL Periodicals Archive – Search Results

## A membership benefit of ARRL and the ARRL Technical Information Service

**ARRL Members:** You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

**Need a higher quality reprint or scan?** Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

**QST on CD-ROM**: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

**Non-Members:** Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.

### Copyright/Reprint Notice

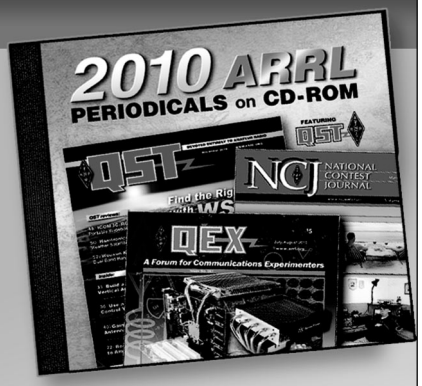
In general, all ARRL content is copyrighted. ARRL articles, pages, or documents--printed and online--are not in the public domain. Therefore, they may not be freely distributed or copied. Additionally, no part of this document may be copied, sold to third parties, or otherwise commercially exploited without the explicit prior written consent of ARRL. You cannot post this document to a Web site or otherwise distribute it to others through any electronic medium.

For permission to quote or reprint material from ARRL, send a request including the issue date, a description of the material requested, and a description of where you intend to use the reprinted material to the ARRL Editorial & Production Department: permission@arrl.org.

**QST Issue:** May 2002
**Title:** What Can You Do with a Dip Meter?
**Author:** Mark Bradley, K6TAF

Click Here to Report a Problem with this File

ADVERTISEMENT




**2010 ARRL Periodicals**
on CD-ROM

**ARRL's popular journals are available on a compact, fully-searchable CD-ROM. Every word and photo published throughout 2010 is included!**

- ***QST*** **The official membership journal of ARRL**
- ***NCJ*** **National Contest Journal**
- ***QEX*** **Forum for Communications Experimenters**

**SEARCH** the full text of every article by entering titles, call signs, names—almost any word. **SEE** every word, photo (including color images), drawing and table in technical and general-interest features, columns and product reviews, plus all advertisements. **PRINT** what you see, or copy it into other applications.

**System Requirements**: Microsoft Windows™ and Macintosh systems, using the industry standard Adobe® Acrobat® Reader® software. The Acrobat Reader is a free download at **www.adobe.com**.

**2010 ARRL Periodicals on CD-ROM**

ARRL Order No. 2001
**Only $24.95***
*plus shipping and handling

Additional sets available:
2009 Ed., ARRL Order No. 1486, $24.95
2008 Ed., ARRL Order No. 9406, $24.95
2007 Ed., ARRL Order No. 1204, $19.95
2006 Ed., ARRL Order No. 9841, $19.95
2005 Ed., ARRL Order No. 9574, $19.95
2004 Ed., ARRL Order No. 9396, $19.95
2003 Ed., ARRL Order No. 9124, $19.95
2002 Ed., ARRL Order No. 8802, $19.95
2001 Ed., ARRL Order No. 8632, $19.95

ARRL The national association for AMATEUR RADIO™
SHOP DIRECT *or call for a dealer near you.*
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)

May 2002 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved



By Mark Bradley, K6TAF

# What Can You Do with a Dip Meter?

## Quite a bit! The dipper is one piece of test equipment that can replace a whole shelf of expensive gear—if you know how to use it.

As radio amateurs we are often interested in resonance. What is the resonant frequency of that antenna I just put up? Is that trap resonant at the frequency I think it is? That crystal, the one with the strange markings, is it good for anything? Do I have an inductor in the junk box that will work in the next project? How do I find the value of those mica capacitors with the cryptic markings? Is that chunk of coax really a ¼ wavelength at the frequency I hope it is?

These are all questions that can be answered by using a dip meter or "dipper" to measure resonance—just one of the instrument's many uses. A dipper makes a very sensitive absorption wave meter for measuring a signal frequency. Since a dipper is an oscillator, I have used it as a signal source to troubleshoot receivers, as well.

All this versatility comes at a price; a dip meter is not a precision instrument. There are techniques to reduce errors to acceptable levels, which will be discussed later. In case you haven't guessed by now I am a big fan of dip meters—mine has allowed me to make many tests that would normally require an extensive array of laboratory equipment.

### What is a Dip Meter?

A dip meter is nothing more than an oscillator with the frequency-determining coil exposed, so that it may be coupled to other electrical circuits. A frequency control is included so the oscillator's approximate frequency is known and can be adjusted. A meter indicates the level of oscillation. Most dip meters come with a set of plug-in coils for wide frequency coverage in several ranges. Older vacuum tube units, in which the meter monitored the grid current of the tube to indicate the level of oscillation, were called grid dip meters. With the availability of high frequency transistors, dip meters went high-tech and battery operation became practical.

The typical dip meter is contained in a small case, with provisions for external plug-in coils. A dial to control the oscillator frequency will be conveniently located on the unit. The meter is located for easy reading while the frequency is being adjusted. Most dippers will also have a control to adjust the level of oscillator activity. This control allows the operator to keep the meter indication at a convenient level over a wide frequency range. If it is a solid-state unit, a battery is included in the case, while vacuum tube units will have an ac power supply that may be self-contained or separate.

Sometimes there is a switch to kill the oscillator to facilitate its use as an absorption wave meter. On others it is possible to turn the activity control down far enough to stop the oscillator. On the front panel there may be an audio output to listen to the modulation of a carrier. Figure 1 shows some common types of dip meters.



**Figure 1—Several common types of dip meters are shown with their plug-in coils that determine the oscillator's frequency.**

### Using the Dipper

When the coil of the dipper is placed near the resonant circuit under test, some of the energy from the oscillating dipper is coupled to the circuit. This coupling reaches a maximum when the frequency of the dipper and the resonant frequency of the circuit are the same. This coupled energy is supplied by the dipper's oscillator, which causes the amplitude of the oscillation to drop. Since the meter indicates oscillation level, a pronounced dip in the meter will be seen as the dipper is tuned through the resonant frequency of the circuit. The oscillator frequency at the minimum or bottom of the dip is the frequency of resonance of the circuit under test. The nice thing is that the circuit being tested does not have to be powered up to measure its resonant frequency.

Placing the axis of the dipper's coil adjacent and parallel to the axis of the coil in the circuit under test results in inductive coupling (Figure 2). This method gives the deepest and most easily found dip on the meter. The dipper's oscillator frequency is "pulled" by the additional load of the resonant circuit—this is one of the major sources of error in making dip meter measurements. Reading the dipper frequency with loose coupling

May 2002 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

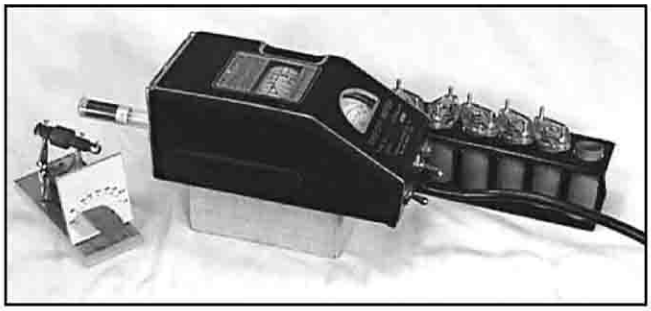


Figure 2—The dip meter's plug-in coil is aligned for inductive coupling with its axis parallel to the inductor of the resonant circuit.

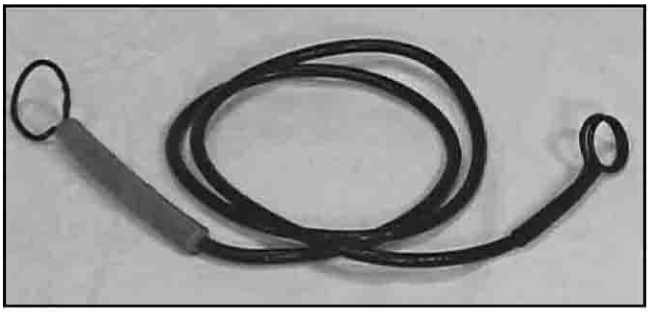


Figure 3—A coaxial cable "link" with a coil at each end is used to extend the reach of the dip meter's plug-in coil.

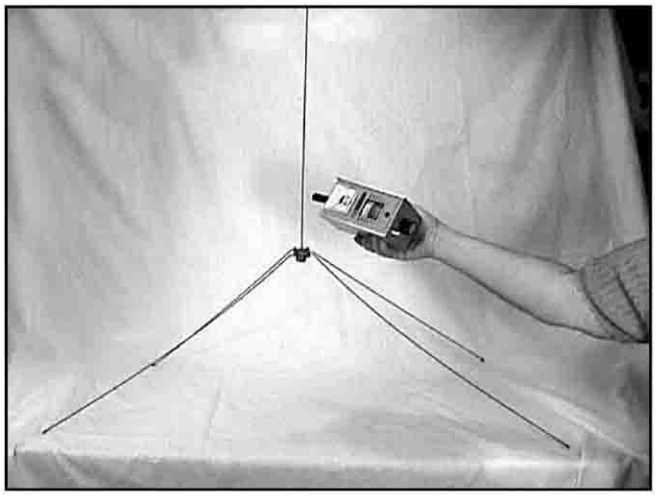


Figure 4—A dip meter may also be coupled to an antenna to determine its resonant frequency.

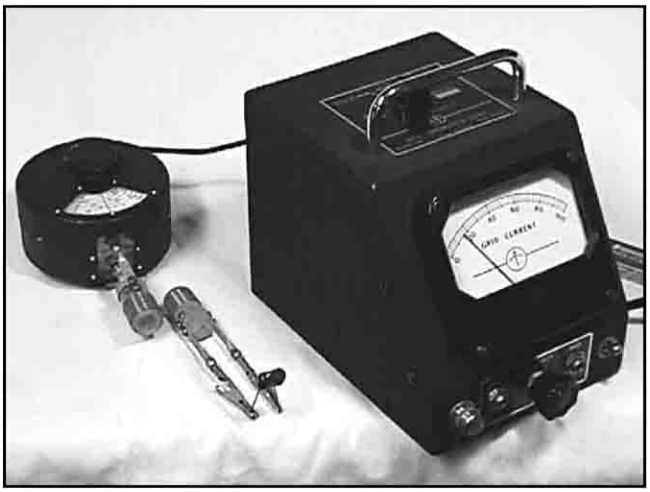


Figure 5—Tune the dip meter's frequency until the unit's meter reaches a minimum—or "dips." The plug-in coil is attached to the capacitor with slide-on alligator clips.

will reduce this error to acceptable levels. After the dip is found I decrease the coupling (move the two coils apart) and recheck the frequency of the dip.

A variation of inductive coupling is link coupling. This allows the dipper to be coupled to circuits in some very cramped places. The link I use is a 2-foot length of coax with a two-turn coil on each end (Figure 3). As the frequency of interest increases, links with fewer coil turns on each end should be used. Two turns can be used up to 70 MHz. Couple one link to the dipper and the other link to the circuit under test.

Capacitive coupling, in which the axis of the coil is perpendicular to the item under test, is useful when there is no inductor present or it is difficult to get to, such as with an antenna (Figure 4). Using capacitive coupling usually produces a shallow dip that is more difficult to see as the dipper is tuned.

### Finding the Resonant Frequency of an LC Circuit

Coupling the dipper's coil to the circuit under test, inductive coupling will produce an easily found dip as you tune the dipper through its frequency range (Figure 5). When you find a dip, move the coils apart to reduce coupling. If the depth of the dip does not decrease, you may find the dip is internal to the dipper. I usually move the coils apart so the dip is no more that 20% to 30% of the maximum meter reading. Loosening the coupling to this point prevents the circuit under test from pulling the dipper's oscillator too badly, and the resonant frequency may then be read off the dipper's dial with a fair degree of confidence.

Can't find a dip? The LC circuit could be outside the range of your dipper. It is helpful to have an idea where to expect resonance and to tune slowly. Occasionally I have found the coil under test to be open, or the resonating capacitor to be faulty, when I could not find a dip. A good way to gain some confidence in using your dipper is to make a parallel resonant circuit from a coil and capacitor. Support the circuit on a nonconductive surface. Practice coupling to the coil of the resonant circuit in every manner you can think of and note the characteristics of the dip.

### Finding the Value of an Unknown Inductor

Connecting a suitable capacitor of known value in parallel with the unknown inductor creates a resonant circuit. Using the dipper you can now find the resulting resonant frequency. I keep fixed-value 5, 20, 100 and 200 pF mica capacitors with my dipper just for making resonant circuits. I also have a calibrated 100 pF variable capacitor for doing quick checks on inductors.

Once resonance has been found, the value of the inductor can be found from the following equation:

$$L = \frac{1}{4\pi^2 f^2 C} \qquad \text{(Eq 1)}$$

where

$\pi \approx 3.1416$

f is in MHz

C is in µF, and

L will be calculated in µH

May 2002 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

Since I dislike doing the math, I have acquired two circular slide rules that will solve resonance problems. Some textbooks have resonance nomographs that can be used. [Chapter 6 *of The 2002 ARRL Handbook* contains a reactance chart that can be used for this purpose.—Ed.]

If you check the same inductor at different frequencies, you will get slightly different values due to the distributed capacitance of the inductor. If the inductor has a metal core, this will also cause inductance to vary with frequency. It is best to check inductors near the frequency of intended use.

### Finding the Value of an Unknown Capacitor

As with the unknown inductor, form a resonant circuit with the unknown capacitor using an inductor of known value. A good source of inductors is the plug-in coils that came with the dipper. As described in the preceding section you can find the inductance of the coils and use them as your inductance standards. To avoid soldering the plug-in coils to capacitors, I found that Mueller makes some alligator clips that slip over the pins of my dipper coils just fine (Figure 5).

Once you find the resonant frequency of the circuit formed with unknown capacitor and known inductor, calculate the value of the capacitor as follows:

$$C = \frac{1}{4\pi^2 f^2 L} \qquad \text{(Eq 2)}$$

where

$4\pi^2 \approx 39.48$

f is in MHz

L is in µH, and

C is calculated in µF

The frequency range of the dipper and the values of the known inductors limit the range of capacitance values that dippers can measure. The largest value of capacitance that can be measured is usually about 1 nF (1000 pF).

### Finding Q of an Inductor

The Q (or Quality Factor) of an inductor is a figure of merit for an inductor. For example, Q is an indication of how sharply a resonant circuit formed with this inductor will tune. There is a good explanation of Q in chapter 6 of *The 2002 ARRL Handbook*. Form a resonant circuit with the inductor to be tested and a mica capacitor. Since the Q of a mica capacitor will be in excess of 1200, the resultant Q of the resonant circuit will be almost totally dependent on the Q of the inductor.

An estimate of Q may be obtained in the following manner. After noting the frequency F and the depth of the dip at resonance, tune the dipper higher in frequency until the dip has been reduced by 30%—this is frequency $F_1$. Now tune lower back through the dip to where the dip has again been reduced by 30%. This frequency is $F_2$. Calculate Q using the following equation:

$$Q = \frac{F}{F_1 - F_2} \qquad \text{(Eq 3)}$$

To make a precise measurement of the frequencies involved, I track the frequency of the dipper with a calibrated receiver as discussed in the section on measuring crystals. Obviously the results are highly operator-dependent but are good enough to tell the difference between a coil with a Q of 20 and one of 50.

### Measuring Quarter- or Half-Wavelength Transmission Lines

The physical length of ¼ wavelength of a coaxial cable may be calculated using the following equation:

$$\frac{1}{4}L = \frac{246}{f}VF \qquad \text{(Eq 4)}$$

where

VF is the velocity factor of the coaxial cable (assumed to be 0.66).

f is in MHz, and

length is in feet

To prepare a ¼-wavelength section of cable, calculate the length of cable using Equation 4 (including the length of any connectors or adaptors), add a few percent and cut. Short one end with a loop and leave the other end open circuited. Couple the dip meter to the loop and look for the lowest frequency dip. This is the frequency at which the cable is approximately ¼ wavelength long.

It is slightly short, due to the detuning effect of the loop. Making the loop smaller will minimize the effect. Page 27-8 of *The ARRL Antenna Book* describes a more accurate method that replaces the loop with a series tuned circuit that resonates at the desired frequency. If you need a half wavelength section, you can use the ¼-wavelength technique at half the desired frequency.

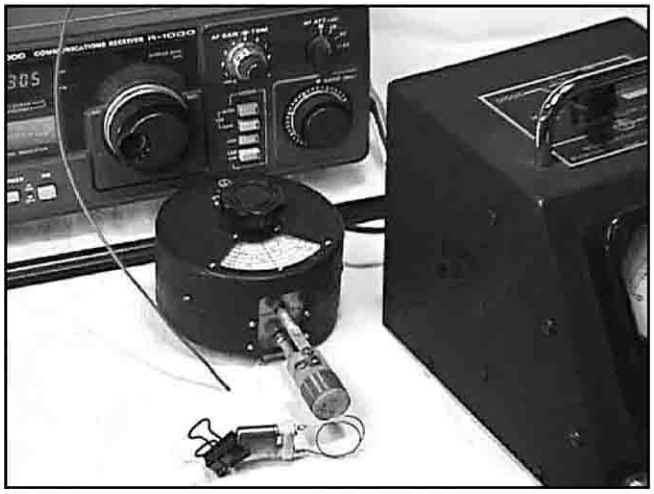


**Figure 6—A calibrated receiver is used to monitor the dip meter's exact frequency while a crystal is "dipped."**

### Measuring Crystals

A crystal's resonant frequency can be found by inductively coupling it to the dipper. I keep several different types of crystal sockets around with two turn loops soldered to them. It's then a simple matter to couple crystals to the dipper. The Q of a crystal is very high, so the dipper must be tuned slowly and watchfully. Because of the high Q, the dipper's frequency may be pulled significantly. For this reason I listen to the dipper on a receiver during the tests (Figure 6). A foot or two of wire lying in the area of the dipper and hooked to the antenna terminal of the receiver is enough coupling. Be sure the BFO of the receiver is on. The crystal frequency found by this method will not be exact but will usually be within 0.2%. The crystal's frequency can be specified only to operate in a circuit with a specified capacitance.

Sometimes a dip will be found at a frequency that doesn't make sense. You may be checking an overtone crystal. Check other harmonically related frequencies for a dip. Even crystals not intended for overtone operation will usually show some activity near their odd overtones.

May 2002 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

### As a Tuned Detector

Most dippers may be used as detectors by turning the oscillator completely off or by turning the activity control down to the point that oscillation just stops. In the first case the dipper will act as a diode detector and in the second case as a regenerative detector. The approach you use will be dependent on the features of dipper.

Many times I have found that a superheterodyne receiver was not functioning because the local oscillator was dead or off frequency. If you are suspicious of an oscillator, dig out the dipper. Couple the coil of the dipper to the oscillator coil. With the dipper in the detector mode, tune the dipper and look for an upward deflection of the dipper's meter as the frequency of the oscillator is found. If there is no upward deflection, the oscillator may not be doing its thing.

For those of us with vacuum tube power amplifiers, the dipper acting as a detector is an excellent indicator of parasitic oscillations that require neutralization. Follow the manufacturer's instructions and be aware that tube amplifiers use lethal high voltages.

Many times we would like to check the operation of a transmitter that has an integral antenna. Radio control models and garage door openers are some examples. A dipper acting as a detector can serve as a field strength meter to check the frequency and level. If your dipper has an audio output, you can confirm the carrier is being modulated, as well.

### As a Signal Source

Since a dipper is a tunable oscillator, it can be used as a signal source to align or troubleshoot a receiver. It will never replace the RF generator but when nothing else is available, it will do. To adjust the signal level, vary the activity control and the coupling to the dipper coil.

### Measuring Impedance

Heathkit, Millen and Eldico made impedance bridges designed to be driven by a dipper. Since these bridges have no means of compensating for reactance, measurements are best made at the frequency of resonance. The range of these bridges is around 10 to 400 ohms.

### Sources of Dippers

Eico, Heathkit, Millen and Measurements Corporation models show up at ham flea markets fairly often. Even some military surplus units are sometimes seen. Pricing seems to be from $3 to $50 depending on the condition and desirability of the particular model involved. If you are looking for a small useful project, why not build a dipper? *The 2002 ARRL Handbook* has construction information in chapter 26. Coil forms are available from Antique Electronic Supply.[1]

### Summary

I hope the information presented here will create some interest in dippers in general and will stimulate the discovery of other applications. For those of us who must pursue our amateur radio activities on a tight budget, the dipper represents great value for the dollar. The dipper is not inherently extremely accurate but with good technique and attention to detail, errors can be reduced to acceptable levels.

*All photos by the author.*

*Mark Bradley, K6TAF, of San Carlos, California, was first licensed in 1955. After college, he spent 19 years working for Ampex Corporation, developing various video tape recorders, instant replay machines and television cameras. He joined a start-up company, Acuson, in 1983 and spent the next 18 years working on various aspects of medical ultrasound imaging. On retirement last June he began home-brewing Amateur Radio projects.*

[1]Antique Electronic Supply, 6221 S Maple Ave, Tempe, AZ 85283, tel 480-820-5411, fax 800-706-6789 (US and Canada) or 480-820-4643; www.tubesandmore.com/; info@tubesandmore.com.

**55 Northam Ave**
**San Carlos, CA 94070**
**northam55@earthlink.net**

# NEW PRODUCTS

## HIGH-PERFORMANCE MICROPHONE PREAMP FROM TDL TECHNOLOGIES

◊ TDL's Model 401 high-performance mike preamp is designed to drive a PC sound card line input, but is useful in a variety of applications that require moderate to high gain and very low noise. The preamp uses a computer-optimized differential input stage and has built-in 9-V rechargeable batteries (and a cast aluminum case) to eliminate any possible interaction with the ac mains.

Other features include 20 to 60 dB gain, a dynamic range of 70 to 100 dB, selectable A, B or C weighting/response curves, three input connectors, a trickle charger for the internal batteries and more.

Price: $183. For more information or to download the Model 401's user manual in PDF format, contact TDL Technology, 5260 Cochise Tr, Las Cruces, NM 88012, tel 505-382-3173, fax 505-382-8810, **www.zianet.com/tdl**.

## MINIATURE SWITCHING POWER SUPPLY FOR FT-817 AND OTHER QRP RIGS

◊ Need a teeny, hash-free switching supply for your FT-817 (or other) QRP radio? Check out MFJ's new Model 4103. Measuring less than 4 × 3 × 2 inches and weighing in at just 10 ounces, the tiny switching supply provides up to 40 W of hash-free dc power (13.8 V dc at 2.9 A max).

On the input side, the supply requires 100 to 240 V ac at 47 to 63 Hz, making the '4103 usable nearly worldwide. Other features/specs include excellent line and load regulation, minimal ripple and noise and an MTBF of 100,000 hours.

Price: $39.95. For more information, contact your favorite Amateur Radio products dealer or MFJ, 300 Industrial Park Rd, Starkville, MS 39759; tel 800-647-1800, fax 662-323-6551, **mfj@mfjenterprises.com**; **www.mfjenterprises.com**.

Previous • Next New Products



# ARRL Periodicals Archive – Search Results

## A membership benefit of ARRL and the ARRL Technical Information Service

**ARRL Members:** You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

**Need a higher quality reprint or scan?** Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

**QST on CD-ROM**: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

**Non-Members:** Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.

### Copyright/Reprint Notice

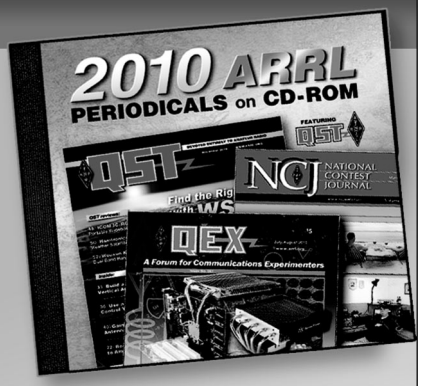
In general, all ARRL content is copyrighted. ARRL articles, pages, or documents--printed and online--are not in the public domain. Therefore, they may not be freely distributed or copied. Additionally, no part of this document may be copied, sold to third parties, or otherwise commercially exploited without the explicit prior written consent of ARRL. You cannot post this document to a Web site or otherwise distribute it to others through any electronic medium.

For permission to quote or reprint material from ARRL, send a request including the issue date, a description of the material requested, and a description of where you intend to use the reprinted material to the ARRL Editorial & Production Department: permission@arrl.org.

**QST Issue:** Jun 2002
**Title:** What Can You Do with a Dip Meter? (May 2002 QST, pp 65-68)

Click Here to Report a Problem with this File

ADVERTISEMENT




**2010 ARRL Periodicals**
on CD-ROM

**ARRL's popular journals are available on a compact, fully-searchable CD-ROM. Every word and photo published throughout 2010 is included!**

- ***QST*** **The official membership journal of ARRL**
- ***NCJ*** **National Contest Journal**
- ***QEX*** **Forum for Communications Experimenters**

**SEARCH** the full text of every article by entering titles, call signs, names—almost any word. **SEE** every word, photo (including color images), drawing and table in technical and general-interest features, columns and product reviews, plus all advertisements. **PRINT** what you see, or copy it into other applications.

**System Requirements**: Microsoft Windows™ and Macintosh systems, using the industry standard Adobe® Acrobat® Reader® software. The Acrobat Reader is a free download at **www.adobe.com**.

**2010 ARRL Periodicals on CD-ROM**

ARRL Order No. 2001
**Only $24.95***
*plus shipping and handling

Additional sets available:
2009 Ed., ARRL Order No. 1486, $24.95
2008 Ed., ARRL Order No. 9406, $24.95
2007 Ed., ARRL Order No. 1204, $19.95
2006 Ed., ARRL Order No. 9841, $19.95
2005 Ed., ARRL Order No. 9574, $19.95
2004 Ed., ARRL Order No. 9396, $19.95
2003 Ed., ARRL Order No. 9124, $19.95
2002 Ed., ARRL Order No. 8802, $19.95
2001 Ed., ARRL Order No. 8632, $19.95

ARRL The national association for AMATEUR RADIO™
SHOP DIRECT *or call for a dealer near you.*
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)

Jun 2002 *QST* - Copyright © 2011 American Radio Relay League, Inc. - All Rights Reserved

make the modification for you. This should be quite simple for anyone who works regularly with SMT boards.

### How to do it

You need a 1000-µF 16-V electrolytic capacitor (RadioShack #272-958) and some #30 insulated wire-wrap wire (RadioShack #278-502). First, clip the capacitor leads to about 1/8-inch on each end. (The fine wire will solder to these leads later.) Prepare two pieces of fine wire, about 3/4-inch long, by stripping 1/16-inch from one end and 1/8-inch from the other. The capacitor will be mounted with a double-faced foam-tape strip (RadioShack #64-2344). Cut a piece of foam strip roughly equal to the size of the capacitor.

Disassemble the 756PRO as follows: Remove all power and other cable connections to the rig. Turn the rig upside down and set its top on a soft cloth on your workbench. The front panel should face you.

Use a wrist strap to ground yourself to the rig and prevent any damage from static electricity. Remove the bottom cover by removing the six screws in the bottom and the two on each side. Loosen the two handle screws by four or five turns. Remove the cover and set it aside.

On the right side of the rig, you will see the main board. The new capacitor connects underneath the DSP unit, which is the large shiny metal shielded assembly on the main board. Carefully remove the DSP unit by lifting it straight up off the main board. You may need to rock it a bit to loosen it. Set the DSP unit aside.

Under the DSP unit are seven capacitors in two rows. Figure 1 shows the 1000-µF capacitor installed on the mounting tape. C316 is located about 1/2-inch below and left of that tape, with wires tack-soldered to each terminal. Notice that the negative side of this capacitor is toward the front panel.

First, tack-solder the 1/16-inch stripped end of a #30 wire to each terminal of C316. We recommend you tin this with a tiny ball of solder (1/64-inch), before touching it down to the capacitor tab. Do this quickly, to avoid overheating.

Next, place the foam tape on the board and install the 1000-µF capacitor, keeping its negative terminal toward the front panel. Finally, solder the #30 wires to the leads of the new capacitor. Verify the proper polarity.

After checking your work, carefully reinstall the DSP unit. Make sure it is carefully aligned before seating the connectors fully. Reinstall the bottom cover and screws. No adjustment or realignment is needed. Your 756PRO is ready to rock and roll—minus the rumble!




**Figure 1—Add a capacitor to cure audio rumble in ICOM IC-765PRO and PROII.**

You may still hear some vestiges of the noise-blanker signal with very, very strong signals, but this wasn't noticeable on the author's unit at normal audio levels. On CW, you will still hear some thumps on very strong signals, but we believe they're entering the audio chain at a different point. In any case, they're substantially reduced by this modification. Key clicks due to sharp filter response will still be present.

The authors would like to gratefully acknowledge the help and support of George, W5YR, and Adam, AB4OJ, for their help in corroborating the observations of the rumble phenomenon, and contributions to the diagnosis.

## ALUMINUM-FOIL ANTENNAS: THE "CATFISH" YAGI

*By Nate Williams, W9GXR, 6915 Prairie Dr, Middleton, WI 53562*

◊ I enjoyed the Doctor's answer to Bob, W6XS, on page 55 of the December 2001 *QST* regarding an aluminum-foil Yagi antenna. Here is the e-mail I sent Bob.

The November 1955 issue of *CQ* magazine carried an article on the "Catfish Special" Yagi antenna for 15 meters. This was the first Yagi I ever built and used. It was a two-element Yagi made up of bamboo poles covered with aluminum freezer foil. The thrust of the article was that the Japanese used bamboo poles painted with aluminum paint in early Yagis. The driven element was split, with a one or two-inch spacing and the Yagi was fed with 300-Ω twin lead. My Dad almost fell off our flat roof as we put this contraption on a 20-foot push-up mast, but did this antenna work out! Of course in the late 1950s, we experienced "once in a lifetime conditions" as far as propagation was concerned. Guys could work the world with AM running 20-W on 10 meters using a hunk of wire for an antenna. Concerning all the technical details, we weren't smart enough to know that it might not work, so we put the doggone thing up and started working DX. None of us had the sophisticated test equipment around today.

Try the "Catfish Special." It's cheap, fun and nobody will believe you are transmitting on freezer foil. Oh yes, the foil was held in place by tape along the elements. None of the dimensions are critical, use spacing between elements and element lengths that fit standard parameters for Yagi antennas.

Technical Correspondence items have not been tested by *QST* or the ARRL unless otherwise stated. Although we can't guarantee that a given idea will work for your situation, we make every effort to screen out harmful information.

Letters for this column may be sent to Technical Correspondence, ARRL, 225 Main St, Newington, CT 06111, or via e-mail to **tc@arrl.org**. The publishers of *QST* assume no responsibility for statements made herein by correspondents. QST

# FEEDBACK

◊ Through no fault of author K6TAF, some errors appear in "What Can you Do with a Dip Meter?" (May 2002 *QST*, pp 65-68). The equal sign was left out of both Eq 1 and Eq 2, and π was left out of Eq 1. Eq 1 should read:

$$L = \frac{1}{4\pi^2 f^2 C}$$

In Eq 3, the minus sign was left out of the expression $F_1 - F_2$.

And finally, the section of the article in the first full paragraph of the second column of page 67, from "Short each end of the cable…" to "at half the desired frequency" should read:

> Short one end with a loop and leave the other end open circuited. Couple the dip meter to the loop and look for the lowest frequency dip. This is the frequency at which the cable is approximately ¼ wavelength long. It is slightly short, due to the detuning effect of the loop. Making the loop smaller will minimize the effect. Page 27-8 of *The ARRL Antenna Book* describes a more accurate method that replaces the loop with a series tuned circuit that resonates at the desired frequency. If you need a 1/2-wavelength section, you can use the ¼-wavelength technique at half the desired frequency.

◊ There is a typographical error in the Web address given in "Radio Line of Sight Plot Server" (May 2002 *QST*, p 107). The correct address is **users.rcn.com/acreilly/los_form.html**. QST