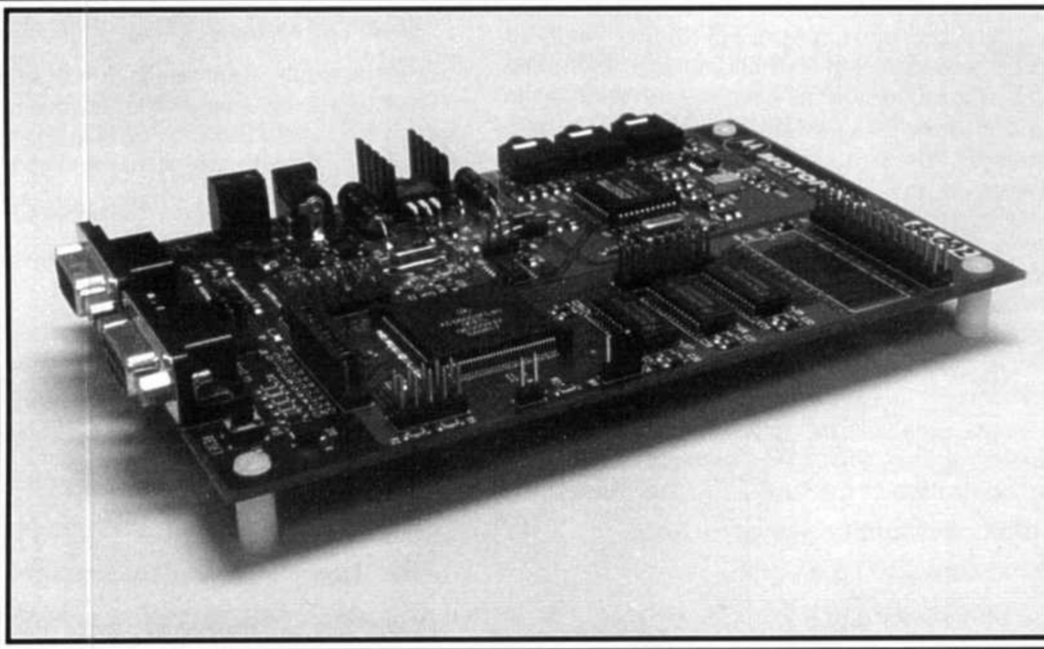
QEX

$1.75

ARRL Experimenter's Exchange

August 1995



High-Power, Low-Cost DSP Development

QEX: The ARRL
Experimenter's Exchange
American Radio Relay League
225 Main Street
Newington, CT USA 06111

QEX (ISSN: 0886-8093 USPS 011-424) is published monthly by the American Radio Relay League, Newington, CT USA.

Second-class postage paid at Hartford, Connecticut and additional mailing offices.

David Sumner, K1ZZ
Publisher

Jon Bloom, KE3Z
Editor

Lori Weinberg
Assistant Editor

Harold Price, NK6K
Zack Lau, KH6CP
Contributing Editors

Production Department

Mark J. Wilson, AA2Z
Publications Manager

Michelle Bloom, WB1ENT
Production Supervisor

Sue Fagan
Graphic Design Supervisor

Joe Costa
Technical Illustrator

Joe Shea
Production Assistant

Advertising Information Contact:

Brad Thomas, KC1EX, *Advertising Manager*
American Radio Relay League
203-594-0207 direct
203-594-0200 ARRL
203-594-0259 fax

Circulation Department

Debra Jahnke, *Manager*
Kathy Fay, N1GZO, *Deputy Manager*
Cathy Stepina, *QEX Circulation*

Offices

225 Main St, Newington, CT 06111-1494 USA
Telephone: 203-594-0200
Telex: 650215-5052 MCI
FAX: 203-594-0259 (24 hour direct line)
Electronic Mail: MCIMAILID: 215-5052
Internet:qex@arrl.org

Subscription rate for 12 issues:

In the US: ARRL Member $15, nonmember $27;

US, Canada and Mexico by First Class Mail: ARRL Member $28, nonmember $40;

Elsewhere by Surface Mail (4-8 week delivery): ARRL Member $20, nonmember $32;

Elsewhere by Airmail: ARRL Member $48, nonmember $60.

QEX subscription orders, changes of address, and reports of missing or damaged copies may be marked: *QEX* Circulation. Postmaster: Form 3579 requested. Send change of address to: American Radio Relay League, 225 Main St, Newington, CT 06111-1494.

Members are asked to include their membership control number or a label from their *QST* wrapper when applying.

About the Cover

KC7WW shows how the new Motorola 56002 evaluation module can be used for amateur DSP development.

ISSUE NO. 162

Features

Columns

August 1995 QEX Advertising Index

THE AMERICAN RADIO RELAY LEAGUE

The American Radio Relay League, Inc, is a noncommercial association of radio amateurs, organized for the promotion of interests in Amateur Radio communication and experimentation, for the establishment of networks to provide communications in the event of disasters or other emergencies, for the advancement of radio art and of the public welfare, for the representation of the radio amateur in legislative matters, and for the maintenance of fraternalism and a high standard of conduct.

ARRL is an incorporated association without capital stock chartered under the laws of the state of Connecticut, and is an exempt organization under Section 501(c)(3) of the Internal Revenue Code of 1986. Its affairs are governed by a Board of Directors, whose voting members are elected every two years by the general membership. The officers are elected or appointed by the Directors. The League is noncommercial, and no one who could gain financially from the shaping of its affairs is eligible for membership on its Board.

"Of, by, and for the radio amateur, "ARRL numbers within its ranks the vast majority of active amateurs in the nation and has a proud history of achievement as the standard-bearer in amateur affairs.

A bona fide interest in Amateur Radio is the only essential qualification of membership; an Amateur Radio license is not a prerequisite, although full voting membership is granted only to licensed amateurs in the US.

Membership inquiries and general correspondence should be addressed to the administrative headquarters at 225 Main Street, Newington, CT 06111 USA.

Telephone: 203-594-0200
Telex: 650215-5052 MCI
MCIMAIL (electronic mail system) ID: 215-5052
FAX: 203-594-0259 (24-hour direct line)

Officers

President: RODNEY STAFFORD, KB6ZV
5155 Shadow Estates, San Jose, CA 95135

Executive Vice President: DAVID SUMNER, K1ZZ

Purpose of *QEX*:

1) provide a medium for the exchange of ideas and information between Amateur Radio experimenters
2) document advanced technical work in the Amateur Radio field
3) support efforts to advance the state of the Amateur Radio art

All correspondence concerning *QEX* should be addressed to the American Radio Relay League, 225 Main Street, Newington, CT 06111 USA. Envelopes containing manuscripts and correspondence for publication in *QEX* should be marked: Editor, *QEX*.

Both theoretical and practical technical articles are welcomed. Manuscripts should be typed and doubled spaced. Please use the standard ARRL abbreviations found in recent editions of *The ARRL Handbook*. Photos should be glossy, black and white positive prints of good definition and contrast, and should be the same size or larger than the size that is to appear in *QEX*.

Empirically Speaking

DSP Tools Abound

As Johan Forrer, KC7WW, describes this month, the Motorola DSP56002 evaluation module (EVM) provides exciting opportunities for use and development of DSP in Amateur Radio. It's not the only player on the block, however. Texas Instruments has had TMS320-series evaluation modules available for some time. And here at ARRL HQ we just received an Analog Devices "EZ-KIT" evaluation module. This $89 unit is similar to the other evaluation modules in that it uses a state-of-the-art fixed-point DSP (the ADSP-2181) and includes all of the necessary hardware (CODECs, for example) and software to allow the user to develop and debug real applications. While the Motorola EVM may still be superior—the 24-bit architecture of the 56000 series is attractive—this Analog Devices EVM is truly inexpensive and capable of running serious DSP applications.

So, joy! Our cup runneth over with DSP development platforms. But there's a downside to this embarrassment of riches. That is that the development efforts of amateurs will be dispersed among the various architectures. Those favoring the TMS320 architecture will use the TAPR DSP-93 or the TI EVMs to write code that may not be easily ported to the Motorola or Analog Devices environments, and vice versa.

There's nothing particularly new about this, of course. If you want a particular application program for your desktop computer, you first have to find out if it's available for your Macintosh, IBM or whatever system. And the availability of different DSP architectures isn't a *bad* thing. It's just unfortunate that we are likely to end up often saying, "That's a neat program! Is it available for my DSP platform?"

One way to minimize the problem is for DSP developers to document the algorithms they are implementing, and to do so in sufficient detail that others can code it on different DSP platforms. Granted, some applications will take advantage of particular features of an architecture that make it difficult to port the program, but such cases should be in the minority.

As always, the pages of *QEX* are available for the documentation of algorithms and programming approaches. Please, let the world know how that neat new DSP application of yours works!

This Month in *QEX*

Measuring the deviation of an FM transmitter is a necessity for packet—and a good idea no matter what the modulation source. But unless you have a friend at the local two-way radio shop, you're unlikely to have access to a deviation meter. Cary T. Isley, W7KIM, provides a design for "A Simple Frequency Deviation Meter Using Time-Domain Techniques" that you can build.

Things are stirring on the 5760-MHz band, as numerous *QEX* articles have shown in the past year or two. More and more transverter, amplifier and antenna designs are cropping up, but one troublesome component remains: the transverter mixer. Paul Wade, N1BWT, addresses this problem with "A Dual Mixer for 5760 MHz with Filter and Amplifier."

DSP for Amateur Radio? Johan Forrer, KC7WW, shows how "Using the Motorola DSP56002EVM for Amateur Radio DSP Projects" not only puts a powerful, low-cost DSP engine in the experimenter's hands, but also gives access to a large set of existing amateur application software.

Finally, in his "Digital Communications" column, Harold Price, NK6K, discusses the Parallax Basic Stamp, a self-contained microcontroller that can be connected to a TNC to build packet remote sensing and control applications.—*KE3Z, email: jbloom @arrl.org*

A Simple Frequency Deviation Meter Using Time-Domain Techniques

No more excuses for packet overdeviation! This easy-to-use meter will let you set it right.

By Cary T. Isley, Jr, W7KIM

With the large amount of packet activity on VHF, attention has been frequently directed to the need for properly setting the frequency deviation of the associated transmitter or transceiver. This article describes a simple frequency deviation meter that provides a way to accurately measure deviation of both 2-meter and 70-cm transmitters and transceivers. It's particularly suitable for HTs. If your interest is primarily packet and you have a DMM, a simplified design approach may be used. A full-bore design can be used if you would also like to check deviation with voice modulation. In the full-bore design, an analog meter is incorporated to display the measured deviation.

Fig 1 depicts the basic principles of the design. A low-level RF signal (in this design, 2 W or less) is applied through a dummy load/attenuator to the input of a harmonic mixer. The LO signal is provided by a crystal-controlled oscillator at a frequency of 48 MHz. The nominal IF of 100 kHz can occur for an input signal of 144.1 MHz (beating with the third harmonic of the LO frequency) and at 432.1 MHz (beating with the ninth harmonic of the LO frequency). The IF signal is recovered by a low-pass filter. The IF signal is amplified by about 21 dB, then squared up by a comparator stage.

The output from the comparator is used to trigger a multivibrator that produces square pulses at a pulse repetition frequency (PRF) equal to the IF. With FM on the input signal, the IF and the associated PRF will vary accordingly. The pulse train from the multivibrator is applied to a low-pass filter with a cutoff frequency of about 10 kHz. This filter effectively removes the fundamental and harmonics of the pulse train, leaving the average of the pulse train and its time-varying component due to the original FM. Measuring the peak value of the ac component of this filtered signal gives a direct indication of the frequency deviation.

An offset circuit is included as part of the calibration procedure. This permits the operator to null out the dc component due to dc offsets or to the difference between the nominal IF (100 kHz) and the actual IF signal. A DMM (set to a dc voltage scale) is connected to the output of the offset circuit. With a signal from the RF source at either 144.1 MHz or 432.1 MHz as appropriate, the offset control is adjusted to produce a null reading on the DMM. The RF signal is then changed by 5 kHz and the gain control at the

1105 Whitehall Lane
Greenville, TX 75402

output of the offset circuit is adjusted to produce a dc voltage that can be directly related to a 5-kHz deviation. An ac voltage scale is used for the actual measurement. Since the ac voltage scale on a DMM is calibrated to indicate the RMS value of a sine wave, the calibration gain (with the 5-kHz offset) should be set to read $\sqrt{2}$ times the scale value selected to indicate 5 kHz of deviation. For example, using the 200-mV scale of the DMM and assuming that 50 mV corresponds to 5-kHz deviation, the calibration level would be set to 70.7 mV. During a measurement, a modulating tone with 5-kHz deviation would then read 50 mV on an ac voltage range of the DMM. It should be noted that this simplified measurement technique is only applicable with tone modulation. For voice modulation, some type of peak-reading voltmeter is needed. Such a meter is included in the full-bore upgrade.

The use of time-domain techniques to effect frequency demodulation is not new. In my ken, it goes back to the early 1950s. As I recall, a manufacturer incorporated this basic technique in a hi-fi FM receiver back then. It was also reported in the *Radio Handbook* as a frequency-demodulation technique. Apparently, the idea never took hold in spite of its stark simplicity. It can provide nearly distortionless recovery of the modulation if both the frequency deviation and the maximum modulating frequency are small compared to the IF—requirements satisfied in the design described here.

Fig 2 schematically depicts the basic design. RF from the transmitter or transceiver is applied to a nominal 50-Ω load comprised of four 220-Ω, ½-W resistors in parallel. This load will accommodate up to 2 W of applied RF. If this does not suit your needs, fabricate a suitable termination. The input load is followed by an attenuator, which reduces the input signal by about 22 dB. With the simple, untuned single-ended mixer used in this design, attenuation can be used to provide a moderate amount of isolation between the signal and LO sources. The LO load resistance is high enough to allow essentially all of the LO output to be applied to the mixer diode and to also provide a moderate attenuation of the LO signal going back into the RF source. The LO signal is provided by a packaged TTL unit that produces an output at 48 MHz. The harmonic mixer is a 1N34A diode. The mixer output is buffered by a voltage follower, U2A, which in turn drives a 5-pole Butterworth low-pass filter with a nominal cutoff frequency of 130 kHz. The filter output is amplified in U2B by approximately 21 dB. The signal out of U2B is applied to a voltage comparator, U3, which squares up the signal. Hysteresis is incorporated in the comparator to reduce the tendency to ring on signal transitions and to provide, in effect, a small amount of threshold bias that inhibits comparator triggering by input noise, so comparator action occurs only when a valid signal is present.

The comparator output is applied to the trigger input of a multivibrator, U4A; a positive-going transition from the comparator will trigger the multivibrator, which will then produce a 6-μs square pulse. The multivibrator output is applied to the input of an active low-pass filter. This filter is a 4-pole Butterworth design with a nominal cutoff frequency of 10 kHz. The dc level out of the filter is used to provide a valid-signal indication by switching on Q1, which in turn energizes LED-1. The filter output is also applied to an offset circuit, U5C, that provides, through adjustment of R24 and R25, a null capabilty for calibration. This could be done using only one pot, but I found that a 10-kΩ pot for coarse setting and a 1-kΩ pot for the fine adjustment offered a much less aggravating nulling procedure. The output of the offset circuit is applied to a gain pot, R27, that sets the level of the signal to be measured. This pot is buffered by a voltage follower, U4D, before application to the DMM.

The basic unit was assembled on a Radio Shack 276-168A prototype card. These cards are very useful for audio and most digital circuits but can introduce problems with RF and high-speed logic circuitry due to the lack of a ground plane. Unfortunately, I did indeed encounter ground-loop problems. Happily, I was able to resolve my difficulties through the judicious placement of bypass capacitors. In some cases, both electrolytics and disc caps in parallel were required. Note that the fabrication of this design was strictly experimental. Since I am a cir-

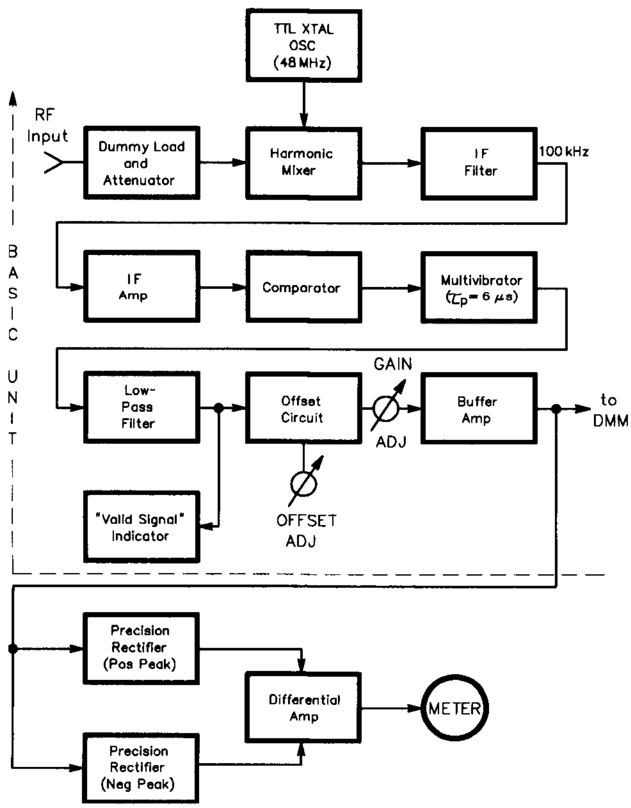


Fig 1—Frequency Deviation meter.

cuit dabbler and dilletante, I usually try to make do with the resources at hand and almost never move to the productizing step once the circuit is functioning. I would suggest that a circuit board with an adequate ground plane might be worthwhile, particularly if cleaning up ground-loop problems is not your forte!

If your interest is packet only, the basic unit is all you need. You can supply the ±5-V power requirements externally, or you may choose to incorporate the internal power supply included in the full-bore design. As noted above, if voice modulation is also to be measured, a method for actually determining the peak deviation must be provided. A peak-reading, precision rectifier can provide this. Fig 3 shows the additional circuitry needed to up-

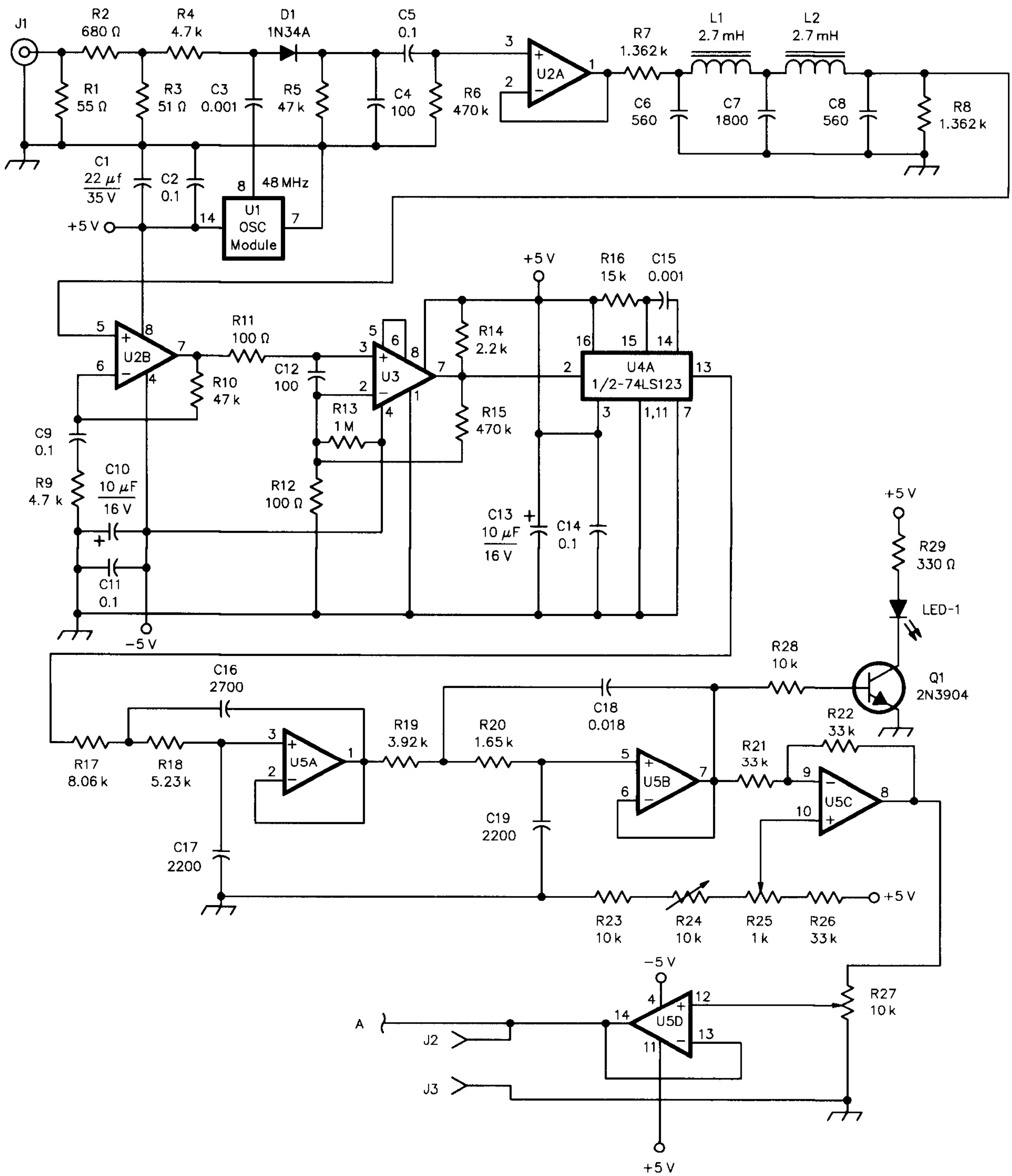


Fig 2—Basic unit.

grade the basic unit to the full-bore design. The output from the basic unit is applied to the inputs of a pair of peak-reading, precision rectifiers configured to determine the peak positive and negative excursions of the applied signal. The rectifier outputs are applied to the inputs of a differential amplifier whose output is proportional to the vector difference of the applied inputs. This output is applied to a 0 to 1-mA meter in a bridge circuit.

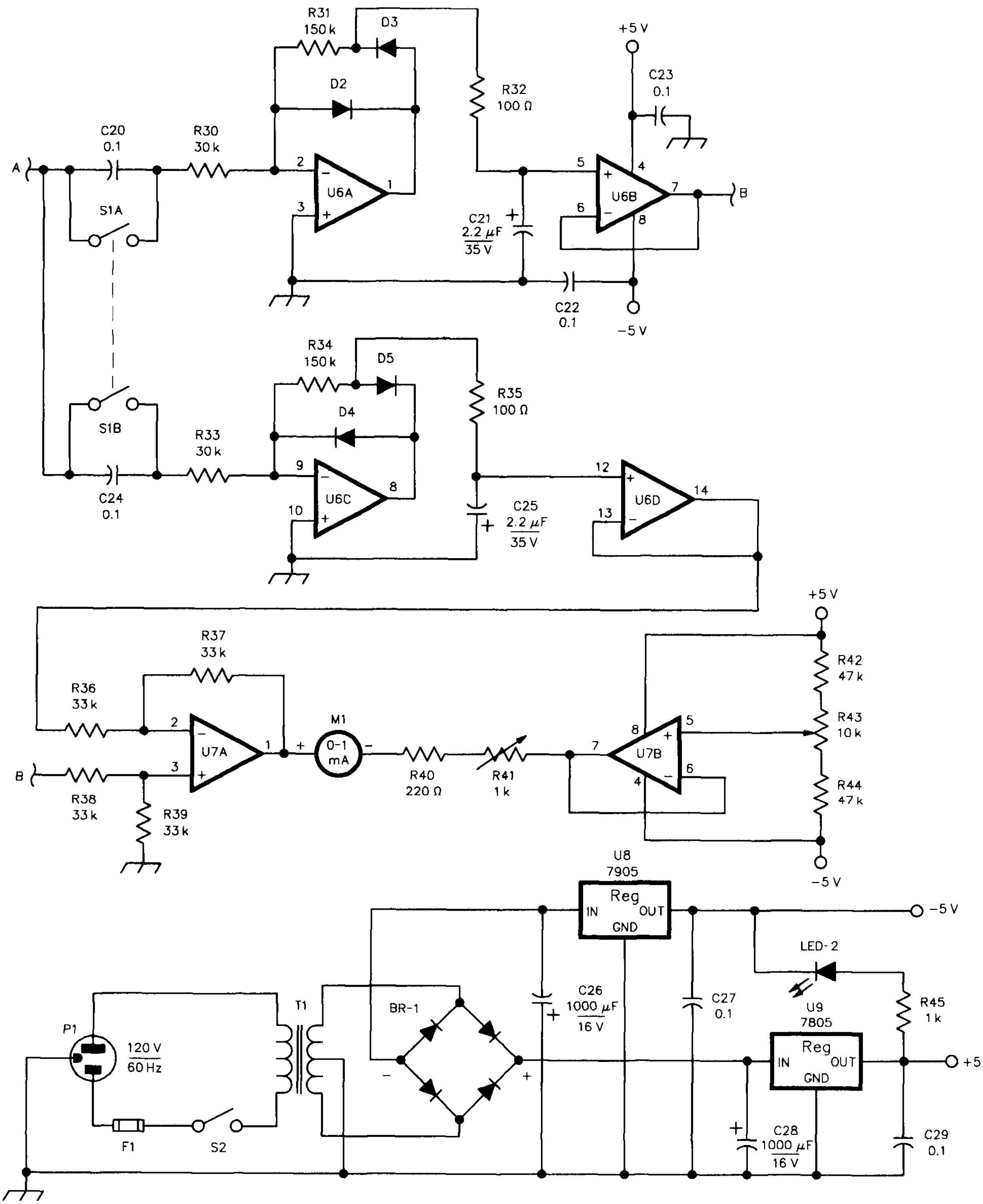


Fig 3—Full-bore upgrade.

The bridge is adjusted to null the meter under no-signal conditions, and the sensitivity of the bridge is set to provide the proper meter indication under calibration conditions.

The precision rectifiers and power-supply components (filter capacitors, rectifiers, and regulators) were installed on a Radio Shack 276-150 prototype board. For the precision rectifiers, I used two of the sections in a quad JFET chip. In my experimenting, I tried using one of the remaining sections as an ac amplifier for the DMM output. This circuit did not prove to be particularly useful, and I have not included it here. The components associated with the peak-reading meter were installed on a second 276-150 prototype board. No ground-loop problems were encountered with these two boards.

The internal meter may be calibrated with the following procedure. First, do the calibration of the basic unit as described above. Notice there is a DPST switch (S1) that, when closed, shorts the coupling capacitors, C20 and C24, so the inputs of the precision rectifiers can be dc coupled. After the basic calibration is done (with the DMM), the internal meter can be calibrated. Make sure that the meter bridge is balanced, then close S1. This action will cause a transient that will usually pin the meter, but recovery is rapid. Then introduce a 5-kHz offset in the RF source and adjust the meter sensitivity pot (R41) to obtain a half-scale reading on the meter. Change the offset to 5 kHz on the other side of the 100-kHz IF and you should obtain the same reading—or at least one very close to it. If not, you probably have a gain mismatch between the precision rectifiers or an imbalance in the differential amplifier. Since I had a large stock of 1% resistors in 30-kΩ and 150-kΩ values, I used them in the precision rectifiers. You can use regular 5% or 10% resistors as long as the respective pairs are matched. For example, matched 27-kΩ and 150-kΩ pairs should be satisfactory. I did use matched 33-kΩ resistors in the differential amplifier. After this step is satisfactorily accomplished, open S1; the calibration is now complete.

The meter I used is a Radio Shack 270-1754 that has a 1-mA movement. It was marketed for use as a voltmeter with a full scale of 15 V. As calibrated above, full scale will correspond to 5 kHz of deviation, so the reading should be divided by 3 to get the deviation in kilohertz. The internal meter will accommodate either packet tones or voice modulation.

I used the deviation meter to adjust TNC output levels into a Yaesu FT-470. I was initially alarmed by the large discrepancy between deviations for the mark and space tones, but a check with a test source showed performance of the deviation meter was okay. Then I realized, after looking at the schematic for the FT-470, that a high degree of preemphasis had been incorporated—almost 6 dB/octave over most of the audio range. Obviously, nothing can be done about that short of perhaps using a deemphasis network between the TNC and the transceiver. Barring this measure, just adjust for proper deviation, say 3.5 kHz, at the higher-frequency tone. I also checked deviation on my multimode rig, an ancient ICOM 260-A. With it, preemphasis was much less pronounced and appeared only in the upper audio range.

Purists will doubtless say that measuring frequency deviation at one particular frequency does not necessarily guarantee the same deviation at some other frequency in the band of interest. In the case of designs where the FM is internally applied to the synthesizer, this assertion may be true in a strict sense, although I doubt the set designer would want to see more than a few percent of variation over the particular band. In the case of multimode rigs, FM is usually developed in a fixed-frequency oscillator whose output is heterodyned with the output of the synthesizer to produce the output frequency. In this case, frequency deviation is obviously unaffected by a change in operating frequency.

As has been noted throughout this article, the design of the deviation meter was strictly experimental. As a result, I really can't provide the usual construction details. However, the design is simple and straightforward, so any experienced builder should be able to construct it without difficulty. In the following paragraphs, I'll describe how I assembled the unit in a cabinet.

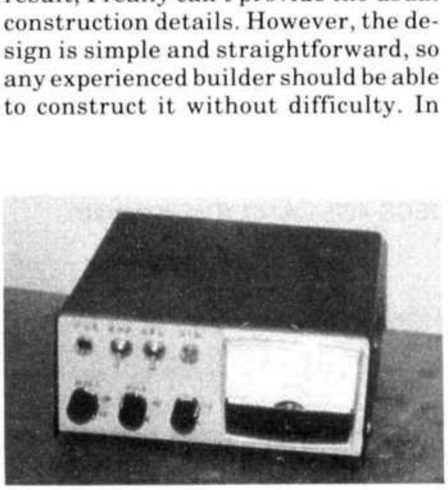


Fig 4

Figs 4 and 5 show the assembled full-bore unit. The circuit boards and the power-supply components are installed in a metal cabinet ($2^3/_4 \times 6^1/_4 \times 7^1/_4$-inch overall). The three pots associated with calibration (R24, R25 and R27), the LED indicators, two switches and the milliammeter are mounted on the front panel. One switch, S1, is used in the calibration procedure. The other switch is used to switch the DMM ac amplifier in and out of use. As I noted, this amplifier did not prove useful and has been omitted in the design described. The circuit boards are mounted on threaded, one-inch, aluminum spacers made by the E. F. Johnson Co. These spacers are very handy for mounting circuit boards. I keep them in plentiful supply. The power connector, the ac on/off switch, the line fuse, the RF input connector and the pin jacks for the DMM are installed on the rear panel. The power transformer, T1, is mounted vertically on the inside of the rear panel. If you elect to follow this type of layout, be certain all of the exposed 120-V points on the power connector, the fuse, and the on/off switch are adequately covered with heat-shrink tubing. The RF dummy load (four 220-Ω, ½-W resistors in parallel) is mounted with R2 directly on the BNC connector, J1. The other end of R2 is connected through a short length of RG-174 to R3, installed on the circuit board. All of the RF circuitry is located very close to this connection point with all ground returns held to the minimum practicable.

I used a quad JFET opamp for U2 functions, but I recommend use of a suitable dual JFET opamp (LF-353 or TL082). I have also shown some minor changes in the additional circuitry

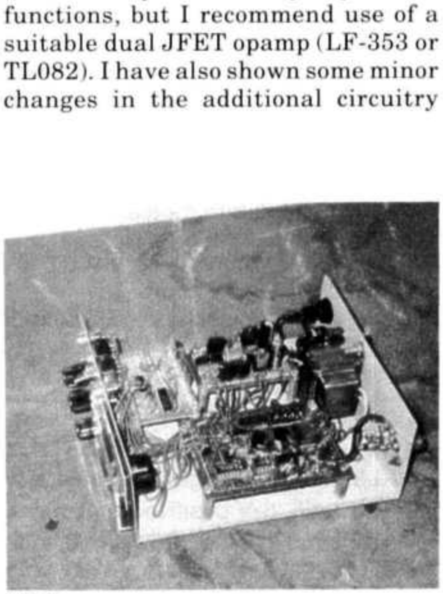


Fig 5

Parts List for Frequency Deviation Meter

BR1—1.2-A, 100 PIV (Radio Shack 270-1120 or equivalent)
C1—22µf, 35-V radial electrolytic
C2, C5, C9, C11, C14, C20, C22, C23, C24, C27, C29—0.1-µf, 50-V disc
C3, C15—1000-pf, 50-V disc
C6, C8—560-pf, 5% 50-V polypropylene
C7—1800-pf, 5% 50-V polypropylene
C10, C13—10-µf, 16-V radial electrolytic
C4, C12—100-pf, 50-V disc
C16—2700-pf, 2% 50-V polypropylene
C17, C19—2200-pf, 2% 50-V polypropylene
C18—0.018-µf, 2% 50-V polypropylene
C21, C25—2.2-µf, 35-V tantalum
C26, C28—1000-µf, 16-V radial electrolytic
D1—1N34A
D2, D3, D4, D5—1N914 or equivalent
F1—250-V, 1-A slo-blo
J1—BNC female chassis mounting
J2, J3—pin jack
LED-1—yellow LED
LED-2—red LED
L1, L2—2.7-mH TOKO type 8RB fixed inductor (Digi-Key part no.TK4317-ND)
M1—0 to 1-mA meter (Radio Shack 270-1754 or equivalent)
P1—CEE-22 compatible connector (Mouser stock no. 161-3516)
Q1—2N3904
R1—four 220-Ω, 1/2-W resistors in parallel
R2—680 Ω, 1/2 W
R3—51 Ω
R4, R9—4.7 kΩ
R5, R10, R42, R44—47 kΩ
R6, R15—470 kΩ
R7, R8—1.362 kΩ, 1%
R11, R12, R32, R35—100 Ω
R13—1.0 MΩ
R14—2.2 kΩ
R16—15 kΩ
R17—8.06 kΩ, 1%
R18—5.23 kΩ, 1%
R19—3.92 kΩ, 1%
R20—1.65 kΩ, 1%
R21, R22, R26—33 kΩ
R23, R28—10 kΩ
R24, R27—10-kΩ pot, panel mounting
R25—1-kΩ pot, panel mounting
R29—330 Ω
R30, R33—30 kΩ, 1% (see text)
R31, R34—150 kΩ, 1% (see text)
R36, R37, R38, R39—33 kΩ matched (see text)
R40—220 Ω
R41—1-kΩ pot, PCB mounting
R43—10-kΩ pot, PCB mounting
R45—1 kΩ
S1—DPST miniature toggle
S2—SPST miniature toggle
T1—120/60-Hz primary; 12.6-V CT @ 450 mA. (Radio Shack 270-1365 or equivalent)
U1—48-MHz TTL oscillator, ECS model OECS-480-1-A101 (Digi-Key part no. X135-ND)
U2, U7—LF353 (or TL082)
U3—LM311
U4—74LS123
U5, U6—LF347 (or TL084)
U8—LM7905, 5-V negative regulator
U9—LM7805, 5-V positive regulator

As required: cabinet, PC boards, mounting hardware, fuse holder, control knobs and IC sockets. All resistors 1/4 W unless otherwise noted. Tolerances for resistors and capacitors are not critical unless otherwise noted.

associated with the full-bore configuration. In the experimental design, I used a quad JFET opamp to implement the precision rectifiers and the DMM amplifier that has been omitted in the recommended design. I used another quad JFET opamp on a second 276-150 circuit board to implement the peak-holding capacitor circuits and the bridge functions for the meter. The peak-holding circuits can instead use the two free sections of U6 and can be assembled on the same card with the precision rectifier components if desired. The comparator, U3, can present problems. After all, the comparator is effectively a high-speed opamp operating with wide-open gain. The circuit measures incorporated in the design help stabilize operation, but it is also very important to observe care in lead dress. Take care to keep the output leads well removed from the inputs since even a small amount of cross-coupling can cause instability.

Since I do a fair amount of experimental design, my parts cabinet and junk box are well stocked. I really didn't buy many new components, but I estimate that all-new parts for the basic unit (Fig 2) are in the $50 range. Most parts are available at Radio Shack. The precision capacitors precision resistors, crystal-oscillator module and the inductors for the IF filter can be purchased from Digi-Key Corp. The CEE-22 compatible power connector, P1, is available from Mouser Electronics. The third and ninth harmonics of the 48-MHz LO provide operation on the amateur 2-meter and 70-cm bands. If operation is desired for use with 1¼-meter FM gear, a different LO frequency must be used. I would suggest the use of a 32-MHz TTL oscillator module (ECS part OECS 320-1-A1O1A or equivalent). Its seventh harmonic when heterodyned against a 224.1-MHz input signal would produce the required 100-kHz IF. I can't vouch for this procedure, since I haven't personally tried it. However, it logically appears to be an extension of the basic design concept.

A frequency deviation meter is not the kind of instrument that's in constant demand or use, but when needed it can be very useful. Constructing such a device seems better suited to a club project than as an individual undertaking. Again, I want to emphasize that my implementation of this design is experimental and that the construction of the deviation meter described in this article should be undertaken only by experienced builders.

A Dual Mixer for 5760 MHz with Filter and Amplifier

This mixer for both transmit and receive sides of a transverter provides good performance.

By Paul Wade, N1BWT

At Microwave Update '92 in Rochester, I presented a description of my modular building-block approach for assembling a transverter for 5760 MHz.[1] I've used this transverter for 3 years, and recently NJ2L described his transverter that uses the same mixer, so perhaps it's time to describe what we've learned and add some improvements.[2]

Mixer

The heart of any transverter system is the mixer, and there are few choices available for 5760 MHz. A recent article by N2SB described a transverter assembled from surplus components.[3] Many components used in the

[1]Notes appear on page 13.

161 Center Road
Shirley, MA 01464
email: n1bwt@iii.net

5.9 to 6.4-GHz microwave relay band work well and are readily available at flea markets, but surplus mixers for this band are scarce, so homebrewing is necessary. One option, the KK7B no-tune transverter, has a simple bilateral mixer for this band, used for both receive and transmit, so switching is needed to use separate power amplifier and receive preamp.[4] Having separate mixers for transmit and receive is preferable so that each path may be optimized.

The KK7B transverter has a 1296-MHz IF, probably because of the difficulty of reproducibly making a sharp filter—or any other high-Q circuit—on a printed-circuit board at this frequency. The dimensions are too small and critical for normal printed-circuit tolerances.

Another transverter, with separate transmit and receive mixers, was described (in German) by DJ6EP and DCØDA and subsequently reprinted in *Feedpoint* and *73*.[5,6] They also described a modification to use a surplus phase-locked microwave source as the local oscillator and made PC boards available, making it even more attractive. I assembled and tested a unit, but the results were abysmal. No apparent mixing was taking place and the only output was strong LO leakage. Closer examination of the mixer circuit suggested that it might be a harmonic mixer, operating with a half-frequency LO. This suspicion was confirmed when we located someone who could fake enough German to translate the article. At 5.7 GHz, the LO input impedance is effectively a short circuit and measures exactly that, preventing it from working as a normal mixer.

It was obviously time for a new design. Some time ago, I designed and built a series of balanced mixers using

90° hybrid couplers from 1296 to 5760 MHz.[7,8,9] Since these worked well as receivers, two mixers were integrated with a third 90° hybrid coupler as a power splitter on a small Teflon PC board. The layout is shown in Fig 1. As expected, it worked well as a receive mixer, with about 7 dB of conversion loss. However, it worked poorly as a transmit mixer, with transmit conversion loss of around 25 dB. This nonreciprocal performance was a mystery until Rick, KK7B, steered me to an article that worked out the math explaining why a 90° hybrid-coupler balanced mixer works as a downconverter but not as an upconverter.[10] I had only worked out the downconverter case and assumed that it would be reciprocal.

One reason for choosing the 90° hybrid coupler is because it is a low-Q structure that uses wide, low-impedance transmission lines, so that dimensions are not extremely critical and performance should be reproducible.

The KK7B mixer used a $^6/_4$-λ rat-race coupler, so the next version, shown in the photograph of Fig 2, used this structure for the transmit mixer (Notes 4 and 9). Line widths are somewhat narrower than the 90° hybrid coupler, but it is still a low-Q structure, so it should still be reproducible. This unit had much better transmit performance, about 8 dB of conversion loss, but its noise figure was not quite as good as the original receive mixer, so the original receive mixer was retained.

The final version integrates "pipe-cap" filters like those in the DJ6EP transverter onto the mixer board (Note 5). These are copper plumbing pipe caps for $^3/_4$-inch copper tubing, with probes $^7/_{32}$-inch long and tuned with an 8-32 screw. Fig 3 is a cross-section sketch of a pipe-cap filter. Dimensions are from the measurements WA5VJB made on individual filters.[11] PC board layout is shown in Fig 4, and the only other components on the board are the mixer diode pairs and a 51-Ω chip resistor termination. IF attenuators like those in some of the no-tune transverters would also fit and are recommended for the transmit side. No through holes are needed for grounding—the radial transmission line stub acts as a broadband RF short. The diodes I used (Hewlett-Packard HSMS-8202) are inexpensive Ku-band mixer diode pairs; they are available from Down-East Microwave, as are the mixer boards.

Mixer Construction

Construct the circuit using minimal lead length on a Teflon PC board, with soldered sheet brass around the perimeter for SMA connector attachment. This is the procedure I use: The copper pipe-cap filter should be installed first, on the ground-plane side of the board. In preparation, I drill tight-fitting holes for the probes and make clearance holes in the ground plane around the probe holes. Then I measure from the holes and scribe a square on the ground plane that the pipe cap just fits inside. Next I prepare each pipe cap by drilling and tapping (use lots of oil) the hole for a tuning screw, then flattening the open end by sanding on a flat surface. Then I apply resin-paste flux lightly to the open end and the area around the screw hole. A brass nut, added to extend the thread length, is held in place by a temporary stainless-steel screw. (Solder won't

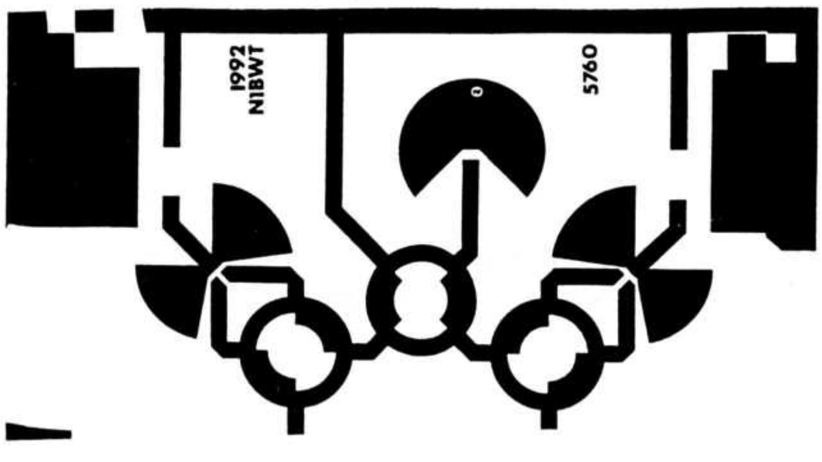


Fig 1—First dual mixer layout

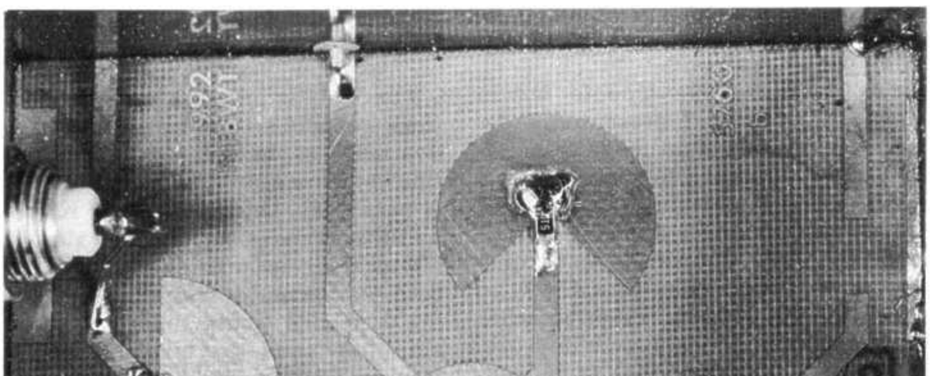


Fig 2

stick to it.) Then I center the open end of the cap in the scribed square on the PC board—the flux holds it in place. Finally, I fit a circle of thin wire solder around the base of each pipe cap and nut, push down gently, and heat each pipe cap for a few seconds with a propane torch until the solder melts and flows into the joints. Don't be shy with the torch—melt the solder quickly and remove the heat.

After everything cools, the temporary stainless-steel screw should be replaced with $^3/_4$-inch long brass tuning screws and locknuts. The remainder of the assembly is performed with a soldering iron, using the photograph of Fig 2 as a guide.

Local Oscillator

Microwave local oscillators normally start with a crystal in the 100-MHz range, followed by a string of multipliers. For 5760 MHz, a multiplication factor of 50 to 60 is necessary—not an easy task. Fortunately, there are many surplus phase-locked microwave sources (often called PLO bricks) available, made by companies such as Frequency West and California Microwave. These units were used in the 5.9 to 6.4-GHz communication band and provide more than enough LO power for the mixer (a 6-dB attenuator was needed with mine). Some units have an internal crystal oven; after a few minutes warm-up, stability is comparable to that of a VHF transceiver. Operation and tune-up of these units has been described by KØKE, WD4MBK and AA5C.[12,13,14] The sources can be used unmodified to provide high-side LO injection, above 5760 MHz, or modified to operate below 5760 for normal low-side injection.[15] Unless you are obsessive about direct digital readout, high-side injection using LSB and reverse tuning is perfectly acceptable. For CW operation, there is no difference.

Most of the available sources operate on –20 V. This is only a problem for portable operation. WB6IGP has described a +12 to –24-V converter, and surplus potted converters are occasionally found.[16] A three-terminal regulator IC provides the –20 V. In order to prevent switching noise generated by the converter from reaching the LO, the converter is contained in a metal box with RFI filtering on both input and output.

Waveguide Band-Pass Filter

A good filter is essential for a serious microwave station, particularly for mountaintop operation. Most accessible high places are crowded with RF and microwave sources, so the RF environment is severe. On other bands, I've seen no-tune transverters with only printed filters fold up and quit in mountaintop environments.

If high-gain amplifiers are used, a good filter is necessary. When I added a surplus power amplifier like the one used by N2SB, the additional 40 dB of gain was enough to amplify the LO leakage through the pipe-cap filter to about $^1/_4$-W (Note 3). Not only is this wasted power, it is also outside the ham band.

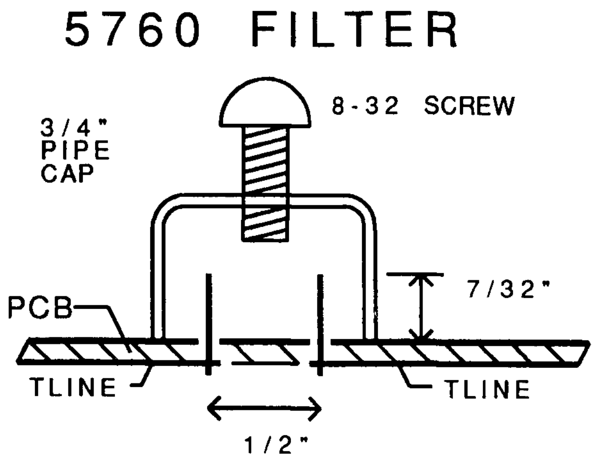


Fig 3—Pipe-cap filter cross-section

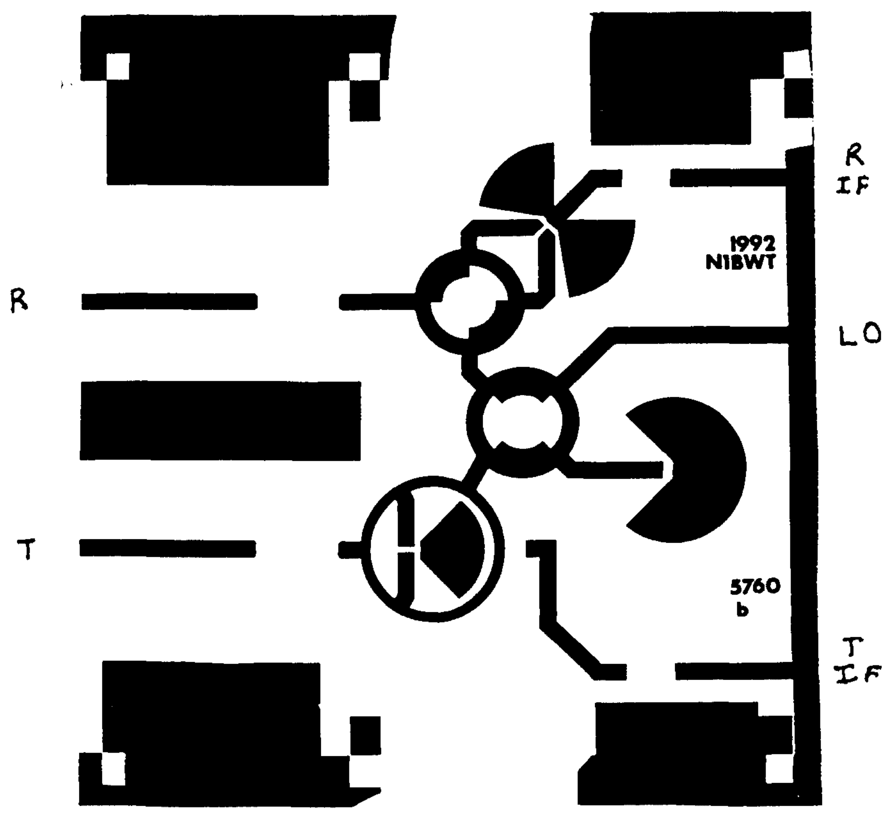


Fig 4—Layout of dual mixer with filters

The best filter I've tried is a waveguide post filter, such as the 10-GHz ones described by N6GN.[17] It is easily built using only a drill, tuning is smooth and non-critical and the performance is excellent. Glenn was kind enough to calculate dimensions for 5760 MHz using standard wave-guide and hobby brass tubing, as shown in Fig 5. Dimensions are for WR-137 waveguide for 5800 MHz; reducing the spacings a hair will give a little more tuning range. I built two units—the second, with careful fit and flux cleaning, had 0.4 dB of loss, while the first, with sloppy fit and soldering, had 0.5 dB of loss. Both units measured as shown in Fig 6, with steep skirts (135-MHz wide at 30 dB down) and no spurious responses detectable (>70 dB down). Tuning was smooth and easy; with high-side LO injection, the LO and image frequencies are outside the tuning range, so 5760 is the only output that can be found while tuning.

Construction hint: Make sure the holes are carefully measured and centered in the waveguide. Centerpunch lightly. Using a drill press, start the holes using a center drill, then drill them out a few drill sizes undersize. Then enlarge them one drill size at a

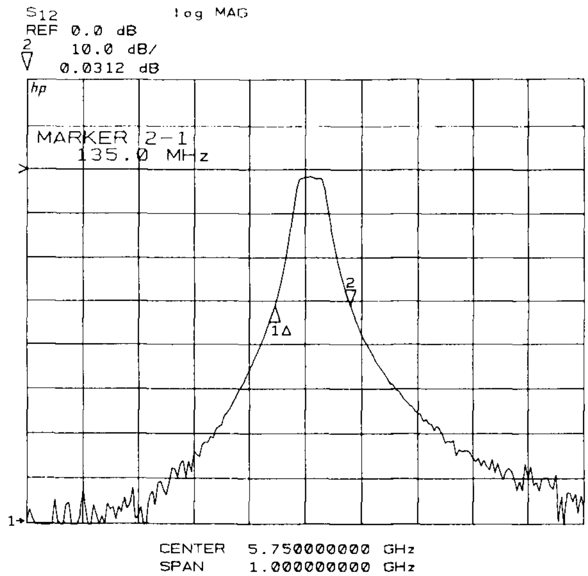


Fig 6—Measured performance of waveguide filter

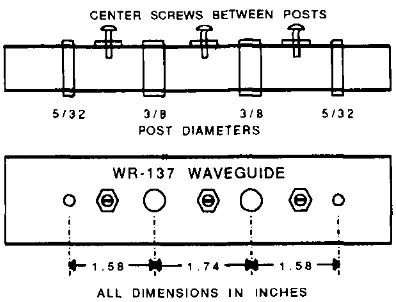


Fig 5—Waveguide filter for 5760 MHz

Table 1—GaAs MMIC Amplifier Performance

Freq	*Gain*	*Return loss*		*|S12|*
MHz		*input*	*output*	
1296	19.2 dB	–3.9 dB	–9.7 dB	–48 dB
2304	20.6	–5.8	–9.7	–37
3456	21.0	–6.4	–10.8	–32
5760	15.4	–7.5	–14.5	–28
10368	7.1	–7.4	–13.9	–20

Noise figure was about 5 dB at 10368 with a 3-dB second stage.

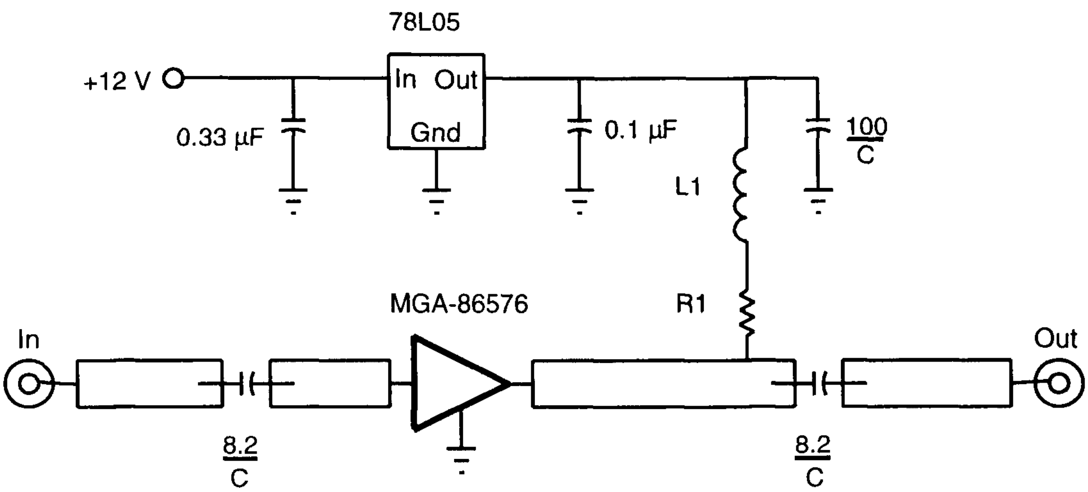


Fig 7

time until the tubing is a snug fit. Solder on a hot plate using paste-rosin flux.

GaAs MMIC Amplifier

Until recently, affordable MMIC devices only worked well at frequencies below 4 GHz. Some devices still had a little gain at 5760 MHz, so it was possible to make an amplifier using multiple low-gain stages.[1,18] A new GaAs MMIC, the MGA-86576 (also available from Down-East Microwave) offers excellent performance to 8 GHz at a very reasonable price. I built an amplifier using one on a scrap of Teflon PC board with a 50-Ω transmission line printed on it. It is important to keep the ground leads very short, so I cut a tight-fitting hole through the board and mounted the MMIC with the ground leads on the ground side of the board and the input and output leads bent up through the hole in the board to the input and output transmission lines. A schematic diagram of this simple amplifier is shown in Fig 7.

The amplifier has about 15 dB of gain and is capable of a few milliwatts of output power, so it would be suitable for a low-power rover station or as a driver for a power amplifier. It also works well as a receiving preamplifier, with a noise figure around 2 dB. The amplifier is quite broadband, with similar noise figure down to 1296 MHz (gain falls off below 1 GHz); measured performance is shown in Table 1.

Conclusion

The dual mixer and two of the GaAs MMIC amplifiers described above could be the foundation of a decent rover station for 5760 MHz, and the addition of a waveguide filter is the next step toward a high-performance station. An obvious next step would be to integrate the MMIC amplifiers onto the dual mixer board; I haven't gotten around to that yet.

Notes

[1]Wade, P., N1BWT, "Mixers, etc., for 5760 MHz," *Proceedings of Microwave Update '92*, ARRL, 1992, pp 71-79.

[2]Healy, R., NJ2L, "A Modular, High-Performance 5.76-GHz Transverter," *QEX*, March 1995, pp 12-15.

[3]Cook, R., N2SB, "5760 MHz from the Junkbox," *QEX*, May, 1994, pp 20-24.

[4]Campbell, R., KK7B, "A Single-Board Bilateral 5760-MHz Transverter," *QST*, October 1990, pp 27-31.

[5]Wesolowski, R., DJ6EP, and Dahms, J., DCØDA, "Ein 6-cm-Transvertersystem moderner Konzeption," *cq-DL*, January 1988, pp 16-18.

[6]Houghton, C. L., WB6IGP, "Above and Beyond," *73*, December 1990, pp 61-62.

[7]Wade, P., WA2ZZF, "A High-Performance Balanced Mixer for 1296 MHz," *QST*, September 1973, pp 15-17.

[8]Wade, P., WA2ZZF, "High-Performance Balanced Mixer for 2304 MHz," *ham radio*, October 1975, pp 58-62.

[9]Keen, H. S., W2CTK, "Microwave Hybrids and couplers for Amateur Use," *ham radio*, July 1970, pp 57-61.

[10]Chang, K. W., Chen, T. H., Wang, H. and Maas, S. A., "Frequency Upconversion Behavior of Singly Balanced Diode Mixers," *IEEE Antennas and Propagation Society Symposium 1991 Digest, Vol 1*, pp 222-225, IEEE, 1991.

[11]Britain, K., WA5VJB, "Cheap Microwave Filters," *Proceedings of Microwave Update '88*, ARRL, 1988, pp 159-163.

[12]Ericson, K. R., KØKE, "Phase Lock Source Update," *Proceedings of Microwave Update '87*, ARRL, 1987, pp 93-95.

[13]Osborne, C., WD4MBK, "Surplus Microwave Local Oscillators, Evaluating and Modifying Them," *Proceedings of Microwave Update '88*, ARRL, 1988, pp 33-41.

[14]McIntire, G., AA5C, "Phase-Locked Microwave Sources," *Proceedings of Microwave Update '91*, ARRL, 1991, pp 113-136.

[15]Houghton, C. L., WB6IGP, "Above and Beyond," *73*, November 1991, pp 66-68.

[16]Houghton, C. L., WB6IGP, "Above and Beyond," *73*, July 1990, pp 68-69.

[17]Elmore, G., N6GN, "A Simple and Effective Filter for the 10-GHz Band," *QEX*, July 1987, pp 3-5.

[18]Hilliard, D., WØPW, "2 GHz to 6 Hz Power Amplifiers," *Proceedings of Microwave Update '87*, ARRL, 1987, pp 78-92.

Upcoming Technical Conferences

1995 ARRL Digital Communications Conference

September 8-10, 1995, La Quinta Conference Center, Arlington, TX—just minutes from Dallas/Fort Worth Airport. Co-hosted by Tucson Amateur Packet Radio (TAPR) and the Texas Packet Radio Society.

For more information contact the TAPR office at 8987-309 E. Tanque Verde Road #337, Tucson, AZ 85749-9399, tel: (817) 383-0000; fax: (817) 566-2544; Internet: tapr@tapr.org

The 1995 AMSAT Annual Meeting and Space Symposium

October 6-8, 1995, in Orlando Florida. For more information contact: Bob Walker, 6601 SW 16th Street, Plantation, FL 33317, (305) 792-7015, email: n4cu@amsat.org.

Microwave Update 95

October 26-28, La Quinta Inn, Arlington, TX.

For more information contact: Al Ward, WB5LUA, 2306 Forest Grove Estates Road, Allen, TX 75002 or Kent Britain, WA5VJB, 1626 Vineyard, Grand Prairie, TX 75052-1405.

Feedback

Whoops! Figures 8 and 10 are interchanged in R. Dean Straw's, N6BV, July 1995 QEX article "The Effect of Local Terrain on HF Launch Angles." The captions are correct.

Using the Motorola DSP56002EVM for Amateur Radio DSP Projects

Motorola's low-cost DSP evaluation module covers all the bases for amateur use and development.

By Johan Forrer, KC7WW

This article is about the Motorola DSP56002EVM, a low-cost, powerful DSP evaluation module (EVM) intended for those experimenting with digital signal processing (DSP). It is ideally suited for filtering CW and EME signals, audio processing applications such as denoising and autonotching and DSP modem work for HF digital and VHF/UHF modems for use in terrestrial and satellite applications. I'll cover hardware and resources for application development and present a collection of advanced DSP applications ready for amateur experimentation.

Background

One way to learn about new computer hardware, especially processor chips, is to actually try your hand at programming them. Unfortunately, for an amateur experimenter with limited resources, mastering a complex processor like the Motorola 56002 DSP is a formidable task. This is not something for the faint of heart.

The recent introduction of low-cost EVMs by DSP chip manufacturers, however, has helped make DSP technology available to experimenters. These EVMs include sufficient educational materials to get you started: a small printed circuit card that contains the DSP chip, software to develop and evaluate educational programs and essential technical reference material. In an amateur-radio environment where cost and availability are major factors, these EVMs have been received with great enthusiasm. The DSP56002EVM is Motorola's entry into this market.

The DSP56002EVM may be used either as a DSP development platform or for a free-standing application when using its onboard E(E)PROM memory. The DSP chip used on the EVM has an impressive track record in communication engineering. Its DSP chip is from the same Motorola DSP family as the chips used in the HAL PCI-4000[1], AEA DSP 1-232, DSP 2-232[2] and the DSP-12[3]—and even in commercial and military products. If judged by these success stories, the 56002 presents splendid opportunities as it offers high clock speed and the advantages of its on-chip hardware emulator.

A Peek at the 56002 DSP

The computing engine of the DSP56002EVM is a 40-MHz 56002 DSP. This is a 24-bit processor, meaning that each instruction is 24 bits

26553 Priceview Drive
Monroe, OR 97456
e-mail: forrerJ@ucs.orst.edu

[1]Notes appear on page 20.

wide. Data and instruction words are kept in separate memory spaces. Instructions are stored in the "P"-space, short for program space. Two separate, independent data spaces, "X" and "Y" are available. This separation of data and instructions is called a *Harvard* architecture. This is quite different from the *Von Neumann* architecture used, for example, in the 80x86 processors. The Harvard architecture is typically used in DSPs and allows for extensive instruction parallelism.

The value of this parallelism becomes evident when one considers that the strength of DSP processors lies in their ability to multiply and accumulate (MAC) very efficiently. The 56002 scores high in this regard. It can multiply two operands, accumulate (sum) the result, fetch two new operands from its two independent data spaces, do address adjustment and fetch the next instruction—all in one clock tick. For a 40-MHz 56002, this happens 20 million times per second. Other DSP chips may have similar features, but having two data spaces and a 24-bit code/data architecture is a distinguishing feature of the 56000 family.

The 24-bit architecture of the 56002 allows a richer instruction set than that of comparable 16-bit DSPs. It also allows an impressive selection of extended addressing modes. These addressing modes are available in one-word instructions and also in a two-word instruction format. Which instruction format to use becomes a trade-off between ease of programming and programming for efficiency. Obviously, the additional dynamic range of 24-bit data (144.5 dB) over 16-bit data (96.3 dB) may prove valuable in certain applications. Higher computational accuracy, or perhaps the ability to detect weaker signals buried in noise, is possible.

The 56002 allows for a considerable degree of flexibility in interfacing configurations. Access by the outside world to the DSP is through several I/O pins, arranged as general-purpose I/O ports named, "PA," "PB," and "PC." Depending on the application, several of these I/O pins double up for use as signals for onboard peripherals such as a synchronous serial interface (SSI) or serial communications interface (SCI). When using these on-chip peripherals in applications, some of these I/O-port pins become dedicated and are no longer available for general-purpose I/O. Port PB in the DSP56002EVM is available at header J7 and is available for the user's interface. Initialization and use of these peripheral devices are documented in the 56002 user's guide.[4] Other DSPs have similar peripherals, particularly with respect to their capability to directly interface high-speed CODECs. But the 56002 SCI and OnCE peripherals are unique for the class of fixed-point DSPs.

The 56002 user-programmable phase-locked loop (PLL) determines the DSP clock rate. A 4-MHz crystal clock is available on the EVM to drive the PLL. This clock, incidentally, also serves as the clock for the on-board 6805 OnCE-interface microcontroller. Such a programmable system clock may serve several purposes. It may allow the designer to meet a variety of internal timing constraints. Or for portable equipment, a variable clock rate may be used for power management. The DSP consumes more power at higher clock rates than lower rates. Power consumption may thus be regulated by selecting the lowest internal clock rate that meets the requirements of the application.

One extremely useful feature of the 56002 is the inclusion of an on-chip sine-lookup ROM that comes in handy for use in communication applications, especially numerically controlled oscillators (NCO) for demodulators.

An Overview of the DSP56002EVM

Fig 1 shows a much simplified overview of the basic hardware components of the DSP56002EVM.

CODEC Interface (SSI)

A Crystal Semiconductor CS4215 coder-decoder (CODEC) handles analog signal conversion. This chip is designed for high-quality multimedia audio purposes and is capable of converting two analog signals to 16-bit digital values (A/D). It also contains two 16-bit digital-to-analog (D/A) converters. These A/D and D/A converters form the analog input and output

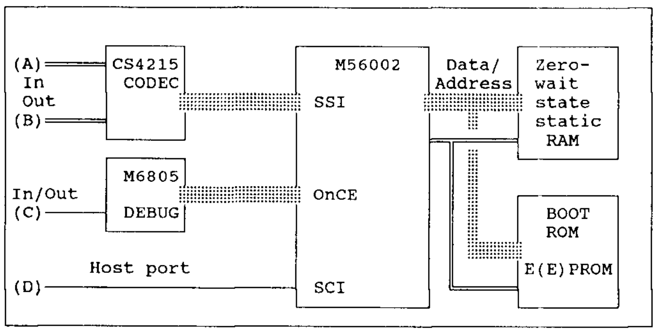


Fig 1—Basic hardware components of the DSP56002EVM. (A)/(B) Stereo input/output ports for analog/digital signal conversion. (C) RS-232 signals for interfacing the DSP56002EVM to a PC computer through a serial interface/used mainly for debugging. (D) RS-232 signals for serial communications.

Table 1. DSP56002EVM assignment of M56002 peripheral port "C" control lines.

M56002 port C	Function
0	SCI Receive (host port)
1	SCI Transmit (host port)
2	CODEC Data/Control (D/C)
3	Not used (used for DCD in Leonid code)
4	CODEC reset
5	CODEC Frame synchronization
6	CODEC Serial clock
7	CODEC Receive (SSI Rx)
8	CODEC Transmit (SSI Tx)

capabilities of the EVM. The CS4215 is a flexible device, capable of a variety of user-programmable data conversion rates, programmable input gain and programmable signal output attenuation. The CODEC interfaces with the DSP via the DSP's SSI using a minimum of interface signals. DSP software must initialize the serial clock rate and framing method of the SSI for the particular CODEC. Additional control lines for reset, and data/control select, are required to fully command and control the CODEC. In the EVM's case, these control signals are assigned from peripheral port "C," as shown in Table 1.

Note that the DSP56002EVM only implements a subset of the CS4215's capabilities. Of particular importance is the following: There is only provision for "mike" input, as the line input is disabled. Only one crystal is used, 24.576 MHz, which is connected to the "XTAL2" position of the CS4215—this normally is reserved for a 16.9344-MHz crystal. XTAL2 must be selected when initializing data sampling rates. Typically, 8, 9.6, 16, 27.42857, 32 and 48-kSPS rates are available. These are commonly used rates for communications applications.

On-Chip Emulation (OnCE) Interface

A 6805-family microcontroller greatly simplifies the OnCE debugging interface. This single-chip solution not only saves board space, but also helps to keep costs down. Its purpose is to translate commands received via the serial interface from the PC-hosted debugging software into parallel control commands for controlling OnCE operation. The low-level OnCE interface logic is very intricate, consisting of several handshaking strobes and intricately-timed pulse sequences—not an easy task to program. The 6805 microcontroller takes care of all this low-level complexity and hides a great number of details from the user without restricting the capabilities of the OnCE, or perhaps only to a small extent: speed. The microcontroller, in its present form, is only capable of operating at a maximum speed of 19200 baud. At this speed, large DSP programs may take a while to download.

Although it is not crucial for a developer to know about the low-level debugger interface protocol to use the DSP5600EVM, it may be valuable when embedding OnCE capability into your own programs, or even to make it possible to use the EVM on other, non-DOS platforms. A summary of the OnCE interface protocol follows.[5]

Communication with the 6805-family microcontroller is eight-bit serial at 19200 baud. The host issues a command code that may consist of a single byte or may be multiple bytes, depending on the command. The response from the microcontroller is also coded,

Table 2. OnCE command structure.

HOST (Hex)	Purpose	EVM Response (Hex)
A0 Rx	Read OnCE register Rx	D7 RaRbRc (4 bytes)
A1 Wa	1-byte write	D1 - success, D2 - failed
A2 WaWbWcWd	4-byte write	D1 - success, D2 - failed
A3	Reset 6805 and DSP	no response
A4	Reset DSP	no response
A5	Force OnCE into debug	D3 - DSP dead D5 - success
A6	Release OnCE	D4 - success
A7	Read DSP status	D7 Sa (2 bytes)
A8 ..not used		
A9	Reset the 6805	no response
AA-AB .. not used		
AC	Check if 6805 is alive	D6 - 6805 responding
AD	Obtain what code level	La - code level

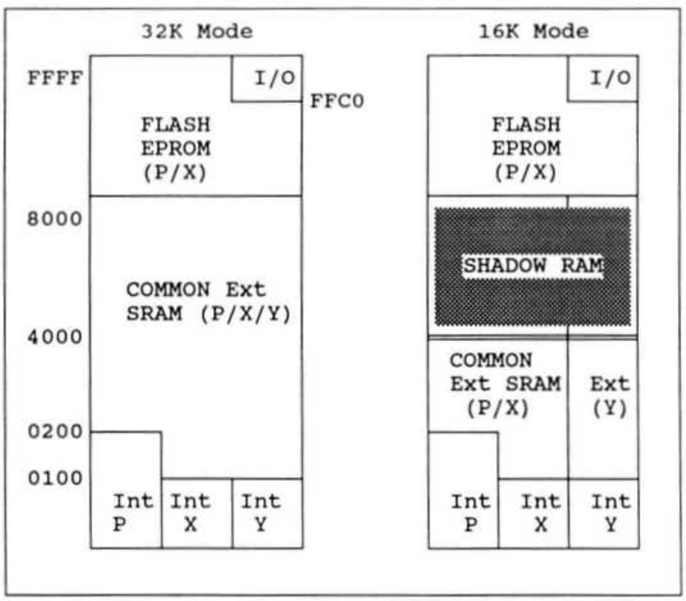


Fig 2—Static RAM layout for the DSP56002EVM. Jumper J12 allows selection of memory configuration.

consisting of a single byte or multiple bytes. In certain situations, such as reset, there is no response, in which case the host should confirm that the EVM is in a usable state by a status query. Table 2 summarizes the command structure.

Host Port Interface (SCI)

The OnCE port is mainly used for debugging purposes. A more traditional serial interface is provided by the SCI. The DSP56002EVM provides a host connector (an optional feature) for this purpose. When the SCI interface is used, PC0 is programmed for SCI data receive and PC1 for SCI data transmit. (These assignments are also shown in Table 1.) Applications that use this interface for communicating with a host may use either ASCII or KISS format.[6] In stand-alone mode, the SCI port is the only means of communication with the EVM. Note that there is no provision for hardware flow control on the DSP560902EVM in its present form—applications must provide software flow control when needed.

Static RAM Memory

The 56002 DSP has 1024 24-bit words of on-chip static RAM (SRAM) that is divided as 512 words of P-memory, 256 words of X-memory, and 256 words of Y-memory. This on-chip memory runs at processor speed and usually is reserved for code that needs to run at the fastest possible speed, such as critical interrupt handlers or variables that need to be accessed very often.

An additional 32K words of SRAM is provided by three (32K × 8) 6206A SRAM chips. Dealing with address decoding logic for these fast DSP chips is very tricky due to the short access times and propagation delays in decoding logic. The EVM designers opted to use a simple, effective method of partitioning the address space. Address lines 14 and 15 and enables X/Y effectively allow for two addressing schemes. These are selected by jumper J12 (16K/32K). The resulting memory layouts are shown in Fig 2.

Both schemes lead to compromises—for example the 32K scheme has P, X and Y all residing in the same physical space. The programmer then has to guard against aliasing side-effects. For example, address P(220) is the same physical RAM location as X(220) or Y(220). In the 16K scheme, on the other hand, P and X share the same space with Y being independent. In this case, the programmer should watch out for aliasing P and X RAM locations. In addition, memory locations above 4000 are "shadow" RAM, that is, address decoding does not distinguish between locations in the range (0000 to 3FFF) and corresponding locations in the range (4000 to 7FFF).

These peculiarities have led programmers to take advantage of the hardware in clever ways. One programming trick that is often used to speed up DSP algorithms on the 56002 is to use a single location pointer for addressing two equal-sized I/O buffers. For example, let one buffer be used as a data source and another for a data destination. If X and Y are then set such that the corresponding elements of the input and output buffers reside at the same numeric address, for example:

```
Input(10)     is at address X(080),
Output(10)    is at address Y(080),
```

then a common single pointer register (R2 in this case) does both fetching and storing of data at the appropriate locations.

```
move X:(R2),x0  ;Input comes from X
move y0,Y:(R2)  ;output goes to Y
```

An example of how this is used can be found in the Leonid kernel's CODEC interrupt handler, where code efficiency is crucial. Some of these peculiarities become important considerations when porting 56002 code across different platforms. The amateur-radio application programs in this article all use the 16K memory mapping.

One fact that is often overlooked pertains to external RAM: No matter which scheme is used, any data accessed in off-chip RAM involves multiplexing/demultiplexing of the address and data busses—the ultimate penalty is speed of execution. It is thus imperative that time-critical code reside in the on-chip RAM if at all possible.

Memory E(E)PROM

The DSP56002EVM includes an option to use an E(E)PROM for booting. This is essential for stand-alone applications. An EPROM such as a 27C256 would be suitable, but this assumes that the user has access to all the necessary tools to convert 56002 object code to a form suitable for booting the DSP. A much more elegant and useful alternative is to use an EEPROM, such as the Atmel At29C256PC FLASH PROM. It is relatively easy to program this device in-place using the software tools supplied with the DSP56002EVM.

Other Miscellaneous Glue Logic

Besides the DSP, CODEC and 6805 OnCE interface, very little additional logic is required to make a functional system. Shifting RS-232 levels to logic levels is handled by a MAX232—both transmit and receive pairs for the OnCE and host interface port are provided for. The 56002 reset logic also requires attention, as the DSP can be forced into one of several operating modes at reset time by presenting the hardware with a special reset signature. For the DSP56002EVM, a 74LS157 chip is used to present a binary "100" pattern to IRQA~, IRQB~, and NMI~ that places the DSP in "EPROM BOOT" mode.[4] Upon recognizing this state, and if a valid E(E)PROM device exists, the DSP's internal boot ROM then performs a bootstrap from this ROM, regardless of whether the OnCE port debugger is used or not.

Development Software and Tools for the DSP56002EVM

The DSP56002EVM ships with the Motorola 56002 version 5.3.2 assembler, Domain Technologies version 1.02 debugger and some (minimal) coding examples.[7] The assembler, as far as I can tell, is very similar to Motorola's professional edition. It generates COFF-format object code. No loader (a program that combines several preassembled objects) is used. This implies that all assembly code is assembled in one step—something that may become awkward when dealing with large projects. The generous use of *include* statements to read in different source code modules is one solution that helps. However, some public-domain software that includes a C language compiler, assembler, linker and 56000 simulator is available that may be of interest.[8]

Domain Technologies, Inc, developed Debug-EVM for the DSP56002EVM. This debugger bears similarities to professional in-circuit emulators in both appearance and capabilities. Such sophistication is made possible by the 56000 OnCE interface that effectively provides in-circuit emulation capability on silicon. Debug-EVM provides access and control over the internal workings of the DSP, memory and peripherals during the debugging process.

Debug-EVM is a DOS-based program that interfaces to the DSP56002EVM through a serial port at 19200 baud. Well-organized multiple-window displays allow for viewing data, program code, DSP registers and processor status in various formats. During the debugging stage of code development, DSP code and data is downloaded from the host PC to the DSP's memory. The developer can then manipulate registers and data, and place breakpoints as required. It also is possible to interact with a running DSP in a non-intrusive manner. This means that the DSP may be debugged at full speed. The debugger offers the following features:

- fully symbolic (data and code labels)
- source level debugging
- on-screen editing
- mouse support
- built-in assembler/disassembler
- character graphics (for plotting memory arrays)
- multiple memory display modes
- trace with count
- single step with count
- subroutine jump
- true real-time hardware breakpoint
- up to 128 software breakpoints
- up to four memory windows
- requires no on-chip monitor code

Amateur-Radio Applications for the DSP56002EVM

The software examples that come with the DSP56002EVM contain only the bare minimum to get you started. One of the included routines is a CODEC support routine, CODEC.ASM. This routine contains a collection of functions that initialize the CODEC and get it up and running. It is a complicated piece of code that requires that the programmer be knowledgeable of both how the CODEC chip works and also all the intricacies of the 56002 SSI interface. Fortunately, the use of *define* statements makes setting up the CODEC a fairly routine procedure. Transferring data to and from the CODEC is fully interrupt driven using agreed-upon buffer locations.

To make the DSP56002EVM more useful to other experimenters, I've put together a program package of useful additional programs. A listing of the programs included in the package is shown in Table 3.

The native DSP56002EVM programs work with the code that came with the EVM—the Leonid kernel is not used. As the simplest code example, the INOUT.ASM program is particularly useful to a beginner to verify the operation of the EVM and carry out some experiments with the CODEC. RTTY.ASM is a bit more advanced and requires further background in FIR filter design. Note that it uses a modified version of the CODEC support routines. The program contains a thorough description of how the DSP modem works. A schematic for a radio interface is included for use with this program.

One code module that may be of general interest is the author's port of the Leonid kernel. This module runs some exciting amateur-radio programs developed by the Finnish Alef Null group.[9] The Alef Null group has been active in the development of DSP projects since 1992—the DSP CARD 4 is their fourth generation DSP card. The DSP CARD 4 is very similar to the DSP56002EVM in many respects, except that it uses a 56001. However, the DSP CARD 4 is a forerunner of the DSP56002EVM. The striking similarity in hardware and software allowed the author to port the Alef Null software to the DSP56002EVM.

Porting Alef Null programs requires

Table 3. Contents of the KC7WW experimenter's program package for the DSP56002EVM. This program package is available for downloading.[11]

Native DSP56002EVM applications (KC7WW)		
INOUT	ASM	-- CODEC talk-through test
RTTY	ASM	-- KC7WW's HF RTTY/AMTOR/Pactor (use with PCTOR)
RCOEFFS	ASM	-- filter coefficients for above
TTYCD	ASM	-- CODEC support for RTTY modem
EVM56K	PLT	-- KC7WW's EVM radio interface schematic (HPGL)
EVM56K	PS	-- KC7WW's EVM radio interface schematic Postscript
Ported applications based on the Leonid kernel Original work by the Alef Null group: Jarkko Vuori, OH2LNS Kaj Wiik, OH6EH and associates		
TALK	ASM	-- INOUT equivalent for Leonid BIOS
BIOS	ASM	-- The source for the ported Leonid kernel
BIOS	CLD	-- loadable module
INTEQULC	ASM	-- essential include files
IOEQULC	ASM	
LEONID	ASM	-- KC7WW's ported Leonid include file
FSK	ASM	-- Jarkko's 1200 baud Bell 202 modem for the EVM
BANDPASS	ASM	-- Jarkko's narrow CW filter
COEFF	ASM	-- filter coefficients for above
QRMQRN	ASM	-- a slightly modified version of Jarkko's LMS denoiser
Contributed programs by Pawel Jalocha, SP9VRC		
CORELNW	ASM	-- Correlation-based CW filter
FFT-CUT	ASM	-- Denoiser based on the spectral subtraction
PARLL1	ASM	-- Early experimental 12-tone OFDM HF modem

```
(1) Initialize CODEC:
        ctrlcd 1,r2,buflen,MIC,0.0,0.0,LINE|HEADP,0.0,0.0

(2) Start CODEC:
        opencd 16, NOHPF
        Jump via P(0028)

(3) Collect data from the CODEC:
        waitblk r2, buflen, batch
```

Fig 3—Command interaction between user program, Leonid interface and BIOS.

only minor changes: pay careful attention to the CODEC input source, P, X, Y and L memory allocation. Then, preload BIOS.CLD before loading the Alef Null application. Finally, start the application using "GO 0." This allows the last-loaded application to gain access to the low-level kernel services. Review the included code examples in the experimenter's package for further details.

The experimenter's package would not have been complete without the contributions of Pawel Jalocha, SP9VRC. These include several excellent examples for advanced DSP experimenters. Pawel has been particularly helpful in including assembly directives that makes his programs run on both platforms.

A Brief Overview of the Leonid Kernel

A DSP kernel is loosely defined as a sort of "command module": a collection of DSP-resident code functions that control, manage and interact with a user's DSP application. An in-depth overview of a complex piece of DSP code the size of the Leonid kernel is beyond the scope of this article. I encourage you to obtain the Alef Null documentation and read it together with BIOS.ASM. The code is reasonably well commented.

The Leonid kernel consists of two parts: the low-level BIOS and a higher-level abstraction that contains several macro calls. The low-level BIOS is contained in BIOS.ASM while the higher-level macro calls are contained in LEONID.ASM. Be sure not to mix the ported versions with the original Alef Null code—these are not quite compatible.

The Leonid kernel contains several critical components necessary for amateur-radio networking applications. A low-level routine handles the HDLC protocol for assembly and disassembly of AX.25 data packets. In addition, the KISS protocol is used together with the HDLC routines. It is thus possible to use the DSP56002EVM with a TCP/IP (NOS) program without any additional hardware. All that is required is the radio, EVM and a PC running NOS. Besides these support routines, additional low-level support is also included for time-out detection and carrier detect—necessary requirements to successfully operate in a network environment. Jarkko's Bell 202-compatible 1200-baud FSK modem (FSK.ASM) is a good example that gives further insight into how this all is put together.

A few hints may help you follow the interrelation of the different program modules. Fig 3 illustrates how a user program interacts with the kernel. Examine this in conjunction with TALK.ASM, a simple example that illustrates how to use the Leonid code and BIOS. The TALK program starts by initializing the CODEC using the "ctrlcd" call (step 1 in Fig 3). This is a macro call (defined in LEONID.ASM) that primes the CODEC's buffers and prepares various pointers and registers that work with the CODEC. The parameters used in the example

Table 4(a): Simplified block diagram for low-level BIOS functionality: Interrupt vector table.

Interrupt vectors: P(0000-001F)	
Reset	vector
Stack Error	vector
Trace	vector
Software interrupt	vector
IRQA	vector
IRQB	vector
SSI Tx	------- Fast interrupt
SSI Tx error	vector
SSI Rx	------- Fast interrupt
SSI Rx error	vector
SCI Rx	vector
SCI Rx error	vector
SCI Tx	vector
SCI Tx error	vector

Table 4(b): Simplified block diagram for low-level BIOS functionality: Jump table.

Jump Table

P-Location	Function	Description
P(0020)	opensci	Opens serial interface. (a)=KISS command routine address else (a)=0 for regular 8-bit serial I/O (b)=Tx on/off routine address.
P(0021)	putc	Places a character in output queue (x0)=argument returns Z is buffer full, NZ otherwise.
P(0022)	getc	Retrieves a character from the queue returns with argument in (x0) returns C if no data available, else NC
P(0023)	tstc	Checks state of queue returns C if no data available, else NC
P(0024)	endc	Terminates a KISS frame.
P(0025)	rejc	Rejects the current KISS frame.
P(0026)	putbit	Places bit in C into host tx queue.
P(0027)	getbit	Gets next bit to be sent returns with bit in C Z if end of transmission.
P(0028)	opencd	Opens CODEC at a specified sample rate Sample rate code is in (x0).
P(0029)	closecd	Shut CODEC down.
P(002A)	stimer	Request a specified time delay (x0) contains number of (1/baud) ticks.
P(002B)	putio	Writes to port PB LSB of (x0) contains argument.
P(002C)	caron	Set transmission "in progress" state. Subject to persistence and DCD logic.
P(002D)	caroff	Set "end of transmission" state. Subject to persistence logic.

P(002E-003D) - not used.
P(003E) Illegal interrupt

shown are as follows:

1—initialize data buffers.

r2—use R2 as a data pointer for reading/writing to the CODEC's buffers.

buflen—defines the internal size of the CODEC's circular buffers.

MIC—a *define* that activates the microphone as input source.

0.0, 0.0—sets the left/right input gain (see CS4215 data sheet).[10]

LINE | HEADP—defines that activate line/headphone output.

0.0, 0.0—sets the left/right output gains (see CS4215 data sheet).

The CODEC is started in step 2 using the "opencd" call. This macro requires two parameters: the first is the sampling rate (ie, 16 means 16 kSPS), the second is whether the on-chip high-pass filter is required or not (see the CS4215 data sheet for its implications). Note that this macro includes a jump instruction to P- address 0028. This location is part of a jump table that routes the command to the appropriate section within the BIOS. This jump table is shown in Table 4(b) as part of the low-order P-address space. The jump to P(0028) will be routed to the CODEC startup code that in turn will bring the CODEC to life. It is imperative that once the CODEC comes alive, its interrupt vector and all associated pointers and data buffers are ready. Otherwise, the code will crash and burn. Jarkko implemented a very efficient CODEC interrupt handler in the BIOS routine by using the so called "fast interrupt" feature of the 56002. The way this works is, instead of a proper interrupt handler routine, two instructions are placed in the interrupt vector address. When a CODEC interrupt occurs, only these instructions are executed, then program execution resumes. These fast interrupts are shown in Table 4(a). The actual instructions in the interrupt vector locations are indirect read/write via register R7, so it is imperative that R7 be reserved for this purpose. Similarly, R3 is used as a SCI buffer pointer (see the SSI interrupt vector in BIOS.ASM for further details).

Actual data collection occurs in step (3). The "waitblk" macro requires three arguments: "R2" indicates that register R2 is to be used as an input/output buffer pointer, "buflen" is the name of the input/output buffer to be used and "batch" contains the block size. The block size feature provides a flexible programming environment. This is illustrated by the following example: If you need to process one sample at a time, the block size will be one. But keep in mind that the data collection process occurs as a background task—often it may take longer to process a block of code than the period between CODEC interrupts. By working with chunks of data at a time, data collection can continue in parallel with processing. This remains valid as long as the time it takes to process a block of data is shorter than the time to acquire a new block of data, so the I/O buffers won't overflow. A further example to illustrate this method of data acquisition is doing real-time FFT processing. Assume that 256-point blocks are acquired and processed and 256 points are written back out. Now suppose it takes longer to process a 256-point FFT than the time between CODEC data interrupts. Clearly, it is not possible to output a continuous stream of processed FFT data. However, collecting a chunk of data first, the FFT calculation can now proceed to completion before the next 256-point chunk arrives. Thus, a continuous stream of data can be collected, processed and written back out, without gaps.There would, however, be a time delay between input and output of duration equal to the time it takes to fill a block.

A number of additional function calls are available that deals with the SCI, timer, KISS and HDLC interface routines. These are listed in Table 4(b).

Conclusion

This article shows that the DSP56002EVM is in a class of its own—outstanding in flexibility and capability that should be hard to beat for some time to come. It is capable of serving the needs of beginners and advanced experimenters in DSP. Amateur-radio experimenters in DSP can also take advantage of an established base of applications.

Acknowledgements

Jarkko Vuori and the Alef Null group's permission to use and publish their code is gratefully acknowledged. This contribution is an outstanding achievement in software engineering.

It was a pleasure to work and discover many new aspects about the 56002 with Pawel Jalocha. His advice, assistance and permission to use and publish his code is acknowledged.

I am also indebted to George Hawkins, KI5X, and the folks at Motorola Digital Signal Processor Systems for their keen support.

Notes

[1]PCI-4000. HAL Communications Corp, PO Box 365, Urbana, IL 61801, tel: (217) 367-7373.

[2]DSP-1232 and DSP-2232. AEA Advanced Electronic Applications, Inc (AEA), PO Box C2160, Lynwood, WA 98036, tel: (206) 774 5554.

[3]DSP-12 Multimode Communications Controller. L.L. Grace Communications Products, Inc, 41 Acadia Drive, Voorhees, NJ 08043, tel: (609)751-1018.

[4]The following literature may be ordered through Motorola's Literature Distribution Center, PO Box 20912, Phoenix, Arizona 85036 (Phone: 1-800-4412447). Please note that some of these materials are free, others not:

DSP56000 Family Manual (DSP56K FAMUM/AD).

DSP56002 Digital Processor User's Manual (DSP56002UM/AD).

DSP56000/DSP56001 Digital Signal Processor User's Manual (DSP56000UM/AD).

[5]*EVM56002 EVM OnCE Interface, Rev 1.1.* Craig Heller et al, Motorola Digital Signal Processing Division, Semiconductor Products Sector, 6501 William Cannon Drive West, Austin TX 78735-8598. This software is supplied with some versions of the kit.

[6]Chepponis, M. and Karn, P. "The KISS TNC: A simple Host-to-TNC Communications Protocol," *Proceedings of the Sixth ARRL Computer Networking Conference*, ARRL 1988.

[7]Domain Technologies, Inc, 1700 Alma Drive, Suite 245, Plano, TX 75075. Developers of the Debug-EVM. Version 1.02 was shipped with the author's system. An updated version, 1.0402, is available for downloading from Domain's BBS (214) 587-1090. Domain also offers a professional version of the debugger, the LINK-56002 development system. In addition, a 56002-based, ISA-bus development card is also offered. Domain Technologies may be contacted via the Internet at: domain@metronet.com.

[8]Public-domain software for the 56000-family is available from several Internet locations. One such location is nic.funet.fi in file /pub/ham/dsp/dsp56k-tools/gcc5616.tar.Z. This program is the work of Andrew Sterian and was intended for the 56116/56156 DSP but, with some reservations, is useful for the 56002. There also is a C language compiler for the 56002, by the same author. This one is named gcc56k.tar.Z. The Alef Null software package mentioned in Note 9, includes a (somewhat dated) assembler, linker, and simulator. These were placed in the public domain because they are not the latest official Motorola versions. However, they all work well. Another useful source for software is Motorola's Dr.Bub BBS (512) 891-3771 or (512) 891-3773.

[9]Kaj Wiik, OH6EH and Jarkko Vuori, OH2LNS. *Alef Null DSP CARD 4, User's Manual.* Manual and software package available for downloading from jeeves.hut.fi. Look in the alefnul directory for program libraries and documentation.

[10]CS4215 16-bit Multimedia Audio Codec. *Crystal Semiconductor Audio Databook.* Crystal Semiconductor Corporation, PO Box 17847, Austin TX 78760.

[11]KC7WW's experimenter's code package for the DSP56002EVM is available for downloading from:ftp.tapr.org/tapr/SIG/hfsig/upload/EVM56K1.ZIP.

Digital Communications

By Harold E. Price, NK6K

A Hardware Interface for Software People

Much like *The Hitchhikers Guide to the Galaxy* has the words "Don't Panic" inscribed in large friendly letters on its cover, the Basic Stamp from Parallax has "BASIC Stamp modules and chips are available at very reasonable prices" etched on its PC boards.[1] References to the Basic Stamp show up from time to time on the odder pages of the 'Net. (One that comes to mind allows you to remotely wave a mechanical hand at the sysop's cat.) Since I don't think we're having enough fun on the amateur digital bands, I decided to look into the Basic Stamp.

This article is based on the BS1-IC Basic Stamp 1 ($34) placed in a #17110 Carrier Board ($15). More information is available at **http://www.parallaxinc.com/** or see the files in the public directory at **ftp.parallaxinc.com**.

The Basic Stamp (hereinafter referred to as "the Stamp") gets its name from two attributes: it is directly programmable in a sort of BASIC language, and the module is about the size (and weight) of a postage stamp. Fig 1 shows the actual stamp module. It is packaged as a 14-pin SIP module, measures 0.5 × 1.5 inches and weighs 0.01 pounds.

Fig 2 shows the module plugged into its carrier board. The carrier board is very useful for testing and has a reset button, 9-V battery connector and prototyping area.

What Does It Do?

The Stamp has 256 bytes of EEPROM-based memory for programs and program data. The user's BASIC program is tokenized by the compiler/uploader (STAMP.EXE) and stored in EEPROM by an onboard loader. The tokens are then interpreted by the Stamp's 18-bit CPU. The BASIC includes standard elements like LET, FOR/NEXT, IF/THEN and GOSUB. It also includes high-level subroutines that make the 256-byte program space stretch further than you might think.

The Stamp has 14 bytes of user accessible RAM, 12 if you use GOSUBs. While this again seems small, many control applications can be written using only one or two variables. The stamp also has eight general-purpose I/O pins.

"Ho hum," you might say. But even though the Stamp has only digital I/O pins, many are the ways in which these I/O pins can be used. With the addition of embedded high-level commands, Parallax has made the stamp into a Really Useful Engine.[2] Any of the pins can be used for any of the functions described below, in any combination.

Pulses

The PULSOUT command inverts the state of an output pin for a given number of 10-µs ticks. As this gives just a single pulse, it can't be used to generate a clock signal unless you take loop execution time into account. It is more useful as a triggerable one-shot.

The PULSIN command returns the length of an input pulse (transition from and back to a given state) in 10-µs ticks.

Button

This command assumes the I/O pin is connected to a momentary switch. It performs debounce and also provides a "repeat on hold down" service. Once the button changes to the desired state, the BUTTON command will return a toggle indication at a given rate after a given delay if the button is held down, similar to the "typematic" action of a computer keyboard.

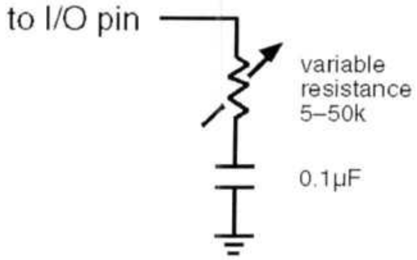
Analog In

The stamp can read variable resistance, sort of, with the POT command. This command determines resistance by measuring the time it takes to dis-

[1]Notes appear on page 24.

5949 Pudding Stone Lane
Bethel Park, PA 15102
email: nk6k@amsat.org

charge a capacitor through the resistor, using this circuit:

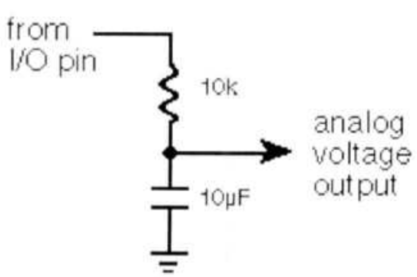


With a capacitor and a thermistor, the Stamp is a simple digital thermometer. I tried this application in the preparation of this article, and found that a standard Radio Shack 0.1-µF capacitor is as good at "reading" temperature as the thermistor is! You must have a temperature-stable capacitor for this application.

Many control applications are built around the ability to read a resistance or capacitance.

Analog Out

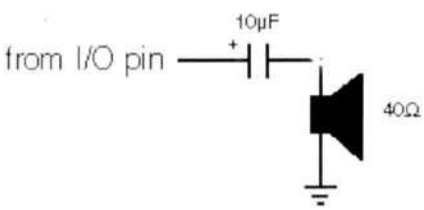
The PWM (Pulse Width Modulation) command will emit a burst of 1's and 0's with a given overall duty cycle. This can be used with the circuit shown here to generate a given analog voltage.



As the charge in the capacitor gradually decays, the PWM command must be used to periodically refresh the voltage.

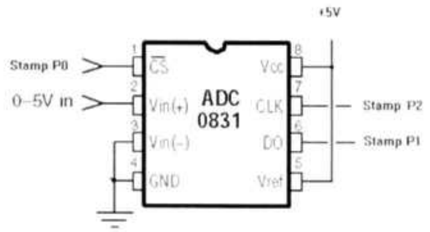
Sound

The SOUND command sends a train of square waves at a given frequency for a given duration. This circuit could be used to output the audio:



Digital I/O

The digital I/O lines can be used to provide an interface to serial interface chips, to provide true analog-to-digital and digital-to-analog conversion or for many other functions. An example of a serially interfaced ADC is the National Semiconductor ADC0831. With serially interfaced devices, the chip sends its data on a serial bus rather than to a parallel processor bus. As shown here, the data pins on the Stamp can be used to trigger an ADC sample and read the resulting serial data.



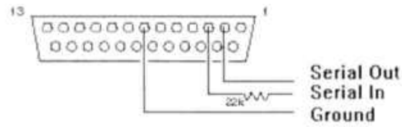
Stamp pin 0 starts a sample, pin 2 provides the serial clock and pin 1 reads the serial data.

According to the Stamp application notes, the ADC0831 Stamp program shown in Listing 1 can sample at a 20-ms rate.[3] The sample program sends the result of the A-to-D conversion to an asynchronous serial port.

Asynchronous Serial I/O

Any pin can be configured as a serial input or output pin, providing asynchronous serial I/O at 300, 600, 1200, or 2400 baud with eight data bits, one start and one stop bit, and no parity. This is suitable for use with a TNC. While the Stamp's serial data is limited to TTL, TTL will work directly with most PCs and TNCs. It worked with my MFJ1270B and a Micron Millennia PC.

The Stamp has internal static protection diodes on all of its I/O pins that limit the plus and minus 12-V RS-232 input signals to +5 V and 0 V. Use a 22-kΩ resistor on the serial input to the Stamp to limit the current and prevent damage to the diodes. Below is a typical configuration to connect the Stamp to a TNC; your pin-out may vary. Labels are from the Stamp's point of view.



Aha!

Now the point of this article should become clear. The Stamp can be used to interface a variety of real-world devices to a TNC for remote access.

A Stamp-equipped TNC, with suitable hardware interface, can control or read back weather instruments, camera slewing equipment, door and light status, hot tub and coke can temperatures...to name just a few common Stamp uses. Because the

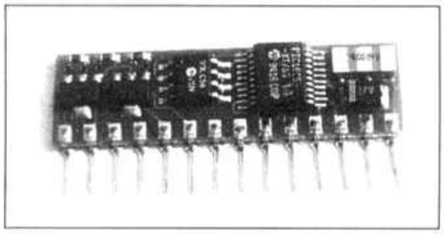


Fig 1—The Parallax Basic Stamp

Listing 1—A Simple Basic Stamp ADC0821 Program.

```
' PROGRAM: ad_conv.bas
' BASIC Stamp program that uses the National ADC0831 to acquire analog data and output
' it via RS-232.
         Symbol CS = 0
         Symbol AD = pin1
         Symbol CLK = 2
         Symbol S_out = 3
         Symbol data = b0 Symbol i = b2
setup:   let pins = 255                     ' Pins high (deselect ADC).
         let dirs = %11111101               ' S_out, CLK, CS outputs; AD input.
loop:    gosub conv                         ' Get the data.
         serout S_out,N2400,(#b0,13,10) ' Send data followed by a return and linefeed.
         pause 2000                         ' Wait 2 seconds
         goto loop                          ' Do it forever.
conv:    low CLK                            ' Put clock line in starting state.
         low CS                             ' Select ADC.
         pulsout CLK, 1                     ' 10 us clock pulse.
         let data = 0                       ' Clear data.
         for i = 1 to 8                     ' Eight data bits.
         let data = data * 2                ' Perform shift left.
         pulsout CLK, 1                     ' 10 us clock pulse.
         let data = data + AD               ' Put bit in LSB of data.
         next                               ' Do it again.
         high CS                            ' Deselect ADC when done.
         return
```

Stamp has a rich set of control primitives and is programmable, Stamp-based solutions should be more programmer friendly than a pile of individual chips.

Stamp Software Development

Part of the appeal of the Basic Stamp is that Parallax knows how to market to core-heads. Their catalog, programming manual and application notes are available on the net in Adobe's PDF format. With the shareware *Acrobat* display program, you can see a reasonable representation of the manual—pictures and all. The manuals and the Adobe reader are available from Parallax's WWW page. Follow the software link to pub/acrobat.

Using the STAMP.EXE program downloaded from the Parallax Web or FTP site (pub/parallax/stamp.zip), I was able to immediately write and test Stamp programs. I modified one of the programs in the Stamp's application notes to gather temperature data and forward it in TNC UI frames, as shown in Listing 2. Rather than just giving the current temperature as in some weather boxes, this program gives data for the previous *n* samples each time it sends. This can be used to show the temperatures every half hour over the last 24 hours.

The program uses GOSUBs for readability, but since each subroutine is only called from one place the size of the program could be reduced by a few bytes by using in-line code. Complex Stamp programs will exercise programming muscles you haven't used for years as you try to save a byte here and there.

The program in Listing 2 uses 110 bytes for program storage and 48 bytes for the stored temps, leaving room for 98 more bytes of program.

The Stamp is programmed through your PC's printer port, using the STAMP.EXE program and the cable described in Table 1.

With the .PDF documentation files and a do-it-yourself programming cable, you don't even need to buy the $99 programming kit.

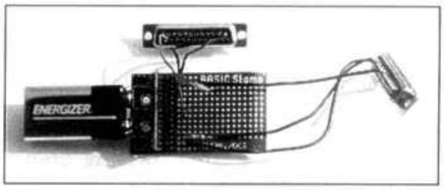


Fig 2—All you need: the Basic Stamp, its carrier, a battery and a connection to the PC parallel port. Also shown is a serial I/O connector for use with the program in Listing 2.

TANSTAAFL

There are, of course, many limitations with a simple device like the Stamp. The Stamp uses 16-bit positive integer math only. That means floating point, negative numbers and numbers bigger than 65535 can't be used. This limitation can be worked around for most control applications. For example, to multiply a variable by π, you must use:

w0=b4*22/7

with 22/7 as an approximation of π.

Table 1—Basic Stamp Programming Cable

Signal	*Stamp Pin*	*PC Printer-Port Pin*
Ground	2 (VSS)	25
Output	3 (PCO)	11
Input	4 (PCI)	2

Listing 2—Basic Stamp program for 24-hour Temperature Readings via a TNC.

```
' Program TNCTEMP.BAS
' This program reads a thermistor with the BASIC pot command,
' computes the temperature using a power-series polynomial equation,
' and sends the result to a TNC via a serial pin.
' The EEPROM is used as a ring buffer to store data.  Data is printed
' newest to oldest. (fifo)

' Based on therm.bas app note #7.
' Harold Price, NK6K

' Symbol constants represent factored portions of the coefficients
' C0, C1, and C2. "Top" and "btm" refer to the values' positions in
' the fractions; on top as a multiplier or on the bottom as a divisor.
' See Stamp app notes #7 for the derivation of this equation.

Symbol  co0     =  162
Symbol  co1top  =  255
Symbol  co1btm  =  2125
Symbol  co2bt1  =  25
Symbol  co2top  =  3
Symbol  co2btm  =  50

Symbol  maxel   =  48  'store 48 samples

'
' register assignments
'
Symbol  sample  =  w0  'pot value (b0 and b1)
Symbol  wscr    =  w1  '16 bit temporary variable (b1/b2)
Symbol  loopcnt =  b1
Symbol  bscr    =  b2  ' 8 bit temporary
Symbol  rp      =  b3  'removal pointer
Symbol  ip      =  b4  'insertion pointer to array
Symbol  count   =  b5  'number of stored elements
Symbol  async   =   6  'async output pin

' program start.  Set number of elements to zero,
' ring insertion point to 0.
  read 255,b0                    'get program bottom location
  debug b0                       'show free space to pc

  let ip = 1                     'start ring buffer at 1
  let count = 0                  'empty buffer

loop:
  gosub  Check_temp              'Get a temp value into sample
  write  ip,b0                   'write lsb of temp into array
  if count >= maxel then skip1   'wrap at maxel elements
  let count = count+1
skip1:

  let rp = ip                    'Set removal point

  gosub wrap_ip                  'increment and wrap insertion
                                 'pointer as required

  for loopcnt = 1 to count       'send the stored values
  read rp,bscr
  serout async,N1200,(#bscr," ")
  gosub dec_rp
  next

  serout async,N1200,(13,10)     'send a CR '
  sleep 1800                     'sleep 1/2 hour
  goto loop                      'again

dec_rp:
  let rp=rp-1                    'decrement RP, wrap around bottom
  if rp > 0 then rp_exit
  let rp = maxel
rp_exit:
  return

wrap_ip:
  let ip=ip+1                    'increment IP, wrap around top
  if ip <= maxel then ip_exit
  let ip = 1
ip_exit:
  return


Check_temp:
  pot 0,46,sample                ' 46 is the scale factor.
  let wscr = sample*sample/co2bt1*co2top/co2btm
  let sample = sample*co1top/co1btm+sample
  let sample = co0+wscr-sample
  return

end
```

Listing 3—PB log data for KO-23 passes at 40.21 N, 80.04 W.

```
950725 055602 - 055801 : Max=612 Bps  Gross=149 Bps  Total Bytes=17836
950725 055931 - 060220 : Max=867 Bps  Gross=375 Bps  Total Bytes=63439
950725 060248 - 061125 : Max=918 Bps  Gross=145 Bps  Total Bytes=75457
950725 075246 - 080329 : Max=867 Bps  Gross=191 Bps  Total Bytes=123180
950725 183017 - 184025 : Max=255 Bps  Gross=6 Bps  Total Bytes=3668
950725 202346 - 203826 : Max=867 Bps  Gross=242 Bps  Total Bytes=213123
950725 221640 - 223351 : Max=816 Bps  Gross=151 Bps  Total Bytes=156626
950726 042100 - 043050 : Max=867 Bps  Gross=176 Bps  Total Bytes=104007
950726 061232 - 062738 : Max=867 Bps  Gross=229 Bps  Total Bytes=208195
950726 185335 - 185812 : Max=437 Bps  Gross=18 Bps  Total Bytes=4991
950726 204054 - 205458 : Max=867 Bps  Gross=209 Bps  Total Bytes=176905
950726 223440 - 225000 : Max=612 Bps  Gross=32 Bps  Total Bytes=29909
```

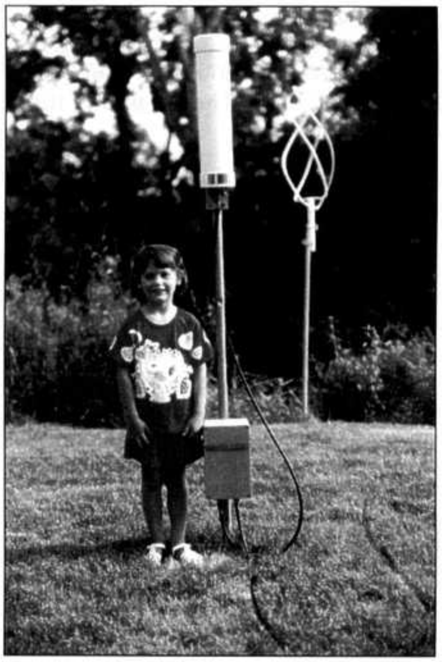


Fig 3—Omnidirectional antennas for amateur digital satellite use.

You must also take care to avoid temporary results that are negative or larger than 65535.

There is no operator precedence and no parentheses; equations are interpreted strictly left-to-right.

The BS1 version of the Stamp does not allow you to "poll" for serial input; you must hang, looking for a character. The BS2 version of the stamp ($20 more, with carrier) does allow polling I/O, as well as other new functions. You can watch the BS1 serial input "manually" of course, waiting for the input to change state and then calling the SERIN function. Send an extra character (or better yet, a break) in front of data input for this to work well. There are many other restrictions, harking back to tiny BASICs of yore, such as: the target of the THEN clause of an IF statement must be a label, it can't be another statement.

Even with these limitations, you can have a lot of fun with the Stamp, gathering data and controlling hardware through your TNC. If you use a Stamp, drop me a line.

Amateur Equipment Prices via the WWW

While browsing the Web looking for Stamp applications, and at the same time wondering what a new TNC costs these days, I came across **http://www.hamradio.com/**, the home page of Ham Radio Outlet stores.

Ham Radio Outlet has most, if not all, of the products they carry in a searchable data base. As you know if you read *QST*, almost all the big store ads now carry a cheery "call" or "lowest" or "new" where the prices used to be. This is a combination of yen values, long advertising lead times and competition, so you can't fault the advertisers, much as I'd like to send a cheery little "no sale" message some times. HRO has come up with an excellent solution to this problem by placing their latest prices in their on-line data base. I entered TNC for product type and got a list of all the TNC makes they carry, along with the prices. Truly useful.

Omnidirectional Antennas

I'm asked from time to time about the use of omnidirectional antennas with the digital ham sats. The question is, of course, "do they work?" The answer is, "sort of." If you are expecting to receive every byte, as is typical with a 10-dB gain antenna and computer tracking, you will be disappointed. If you would be happy getting some data, some telemetry and uploading short messages, then an omni will do the job.

I've been working with a commercial ground station called the "Integrated Ground Station" (IGS) from Surrey Satellite Technology for my real job lately. This unit has receiver, transmitter, file upload, download and pass prediction all rolled into one box, so it's easy to let it run off in a corner by itself. This is what makes omni antennas look good, by the way. If you aren't sitting in front of the screen noticing all the data you're missing—watching the pot not boil, as it were—performance is psychologically much better. I have the IGS hooked up to quadrifilar antennas from Satellite Data Systems. Quadrifilars are of interest because they give circular polarization, as recommended for use with the UoSAT and Microsat type spacecraft. The IGS is not intended for amateur use, but I've monitored KO-23 and other amateur satellites with it. I'm using a preamp mounted near the base of a 430-MHz downlink antenna and about 80 watts of two-meter uplink. Fig 3 shows an omni antenna farm, with four-year-old Rebecca for scale.

Geometry seems to make a big difference in reception; some passes that look good based on az/el and local blockage do not result in many received bytes, while others work very well. Listing 3 is the PB.LOG file from two days of KO-23 passes. I have the antennas mounted on four-foot-high poles on the ground; the antenna is blocked by houses to the north. I can see from 0 to 70% of a KO-23 pass, depending on the geometry of the pass.

I'm located at 40.31N 80.04W, in case you want to reproduce the pass az/el that matches this data. As you can see, two days resulted in 1.1 Mbytes of data—less than 20% of what was available, but still a large number.

Next Time...?

That's it for this month. I don't have a clue what to talk about next time. If there is anything you'd like to see discussed, write me.

Notes

[1]Parallax, Inc. 3805 Atherton Road #102, Rocklin, CA 95765.

[2]An especially obscure reference to "Thomas the Tank Engine." You have to have a four-year-old.

[3]The *ad_conv* sample program and some of the diagrams are from the Parallax documentation and are used with permission.

Proceedings

Proceedings of the 29th Conference of the Central States VHF Society. ISBN: 0-87259-519-6. Cost: $12 from ARRL. Order number: 5196.

SETI Range: How Near Do We Hear?, Dr. H. Paul Shuch, N6TX

Conservative Design for 10 Watt V/UHF Transverters, Bill Wageman, K5MAT

A Matched Filter for EME, C. R. MacCluer, W8MQW

SMIRK: History, Accomplishments and Future, Pat Rose, W5OZI

Amplifiers and Tubes, Russ Miller, N7ART

A 23-cm Diagonal Waveguide Feed for Most Dishes, Russ Miller, N7ART

Cavity Design Tips, Russ Miller, N7ART

Easy Local Oscillators, Chuck Clark, AF8Z

144-MHz Preamp, Thomas Henderson, WD5AGO

The Rolling of Wheels, Dave Clingerman, W6OAL

The Di-Conical Monopole Antenna, Dave Clingerman, W6OAL

ATV "L"-Band Filter, Dave Clingerman, W6OAL

The Beachball Antenna, Dave Clingerman, W6OAL

The ATV Ballooning Ground Station, Dave Clingerman, W6OAL

The Skeleton Slot, Dave Clingerman, W6OAL

Amateur Television (ATV) Ballooning Video Signal Polarization,

David A. Clingerman, W6OAL

Don't Discount The Discone, Dave Clingerman, W6OAL

How Useful is the Groundplane Antenna?, Dave Clingerman, W6OAL

A VHF SSB-CW Transceiver with VXO, Rick Campbell, KK7B

50 MHz TVI, Ray Rector, WA4NJP

Amateur Radio at the Edge of Space, Jack Crabtree, AAØP

Low-Loss Connector Grease, Kent Britain, WA5VJB

High Efficiency Power Amplifier Design, Jerry Kleker, W9QXP

NØYGM: An Amateur Radio Station Anthology, Ken Knight, NØYGM

Graphing Receiver Sensitivity, Ray Uberecken, AAØL

The Central States VHF Society Members: A Brief Profile, Emil Pocock, W3EP

Portaple and Rover Operating, Randy Simons, NØLRJ

Proceedings of the Twenty-First Eastern VHF / UHF Conference of the Eastern VHF / UHF Society. ISBN: 0-87259-525-0. Cost: $12 from ARRL. Order number: 5250.

The Eastern VHF/UHF Conference Experience, Stan Hilinski, KA1ZE, Chairman

The Eastern VHF/UHF Conference Trivia Quiz, Stan Hilinski, KA1ZE, Chairman

Operating Related Topics

50 MHz Beacons List, Geoff Brown, GJ4ICD

VHF/UHF Beacons List, Ron Klimas, WZ1V

Solar Equivalent Scale for K and A Indices, Ron Klimas, WZ1V

European 48/49 MHZ TV Transmitters List, Geoff Brown, GJ4ICD

Transatlantic VHF TV, Emil Pocock, W3EP

A Review of the Amazing Sporadic-E Phenomenon, Ken Neubeck, WB2AMU

Narrow-Band 10 GHz and Some Observations From New England, Tom Williams, WA1MBA

144 MHz EME Basics, Tim Marek, NC7K

Some Notes on 2 Meter Operation, Roger Amidon, K2SMN

"Atlantis This is N2XOU", SAREX Footnote to a Historic Mission, Fred Winter N2XOU, Bruce Wood N2LIV, Niel Heft KC2KY

Antennas

How Does One Determine Proper Stacking Distances for Yagis?, Ian White, G3SEK

Antennas—How Long is Enough ?, Hank Kasper, K2GAL

Improved T-Match for the CC 3219, 4218XL, Etc., Hank Kasper, K2GAL

Equipment Design & Modification

Suggestions for Modification of Qualcomm LNA Boards for 10 GHz, Ken Schofield, W1RIL

Modification Update of Qualcomm Omnitrack PA Board for 10 GHz, Ken Schofield, W1RIL

An AC/DC Power Supply for Microwave Brick Oscillators, Ken Schofield, W1RIL

10 GHz Waveguide to Coax Transition, Ken Schofield, W1RIL

Linearization of Solid State "Brick" Amplifiers, Fred Stefanik, N1DPM

Improving the IC202S 144 MHz Receiver, Dave Robinson, WG3I, G4FRE

The WG3I PIC Keyer, Dave Robinson, WG3I, G4FRE

RF Field Indicators for 432 MHz and 1296 MHz, Ian White, G3SEK and Roger Blackwell, G4PMK

FYI: Fixing Dead Bird 43 Slug, David Feldman, WBØGAZ

Using and Understanding Decibels, Paul H. Bock Jr., K4MSG

Revitalization of the Yaesu FT-736, Bruce Wood N2LIV and Ron Katz WB2DVK .

A Digital Sequencer for Tower Mounted Preamps—Update, Ron Klimas, WZ1V

Improvements to Using the Heath SB-200 Linear on Six Meters, Ron Klimas, WZ1V

Economical New Power Tubes for Amateur Use, Ron Klimas, WZ1V

Testing

Antenna Gain Measurements, Twentieth Eastern VHF/UHF Conference

Noise Figure Measurements, Twentieth Eastern VHF/UHF Conference

The Truth about UHF Connectors, Alan Bloom, N1AL

Selected Reprints from QEX of Interest to Microwave Experimenters