QEX 14 March 1983

The ARRL Experimenters' Exchange

W4RI and QEX Are Moving to Newington

ARRL General Manager David Sumner, K1ZZ has announced my appointment to the position of Manager of the ARRL Technical Department, succeeding Doug DeMaw, W1FB. The move to Hq in Newington, CT will take place in early May, 1983 when Doug retires from full-time duty with the League.

This and the April issue of QEX will come from McLean, VA. There will be no May issue in order to make time to fix up the house, pack and move everything to Newington. QEX publication will resume with the June issue. Please don't worry about losing an issue from your subscription because May won't be counted as an issue.

Will this change QEX? The first issues from Newington will probably have very little change. The goal is to continually improve the editorial content and, if feasible, increase the technical information in each issue. When the new word-processing and photocomposition equipment at Hq get tied together, it will be possible to improve the print quality of QEX. There will also be more opportunity for members of the Hq technical staff to become involved in QEX as they now are in other League technical publications.

Starting when you read this issue, please send all mail pertaining to QEX to:

American Radio Relay League
225 Main Street
Newington, CT USA 06111

Envelopes containing manuscripts and correspondence for publication in QEX should be marked:

Editor, QEX

QEX subscription orders or change-of-address notices may be marked:

QEX Circulation

Mission of the Technical Department

Historically, the Technical Department has had four basic functions:

1. Editing of QST technical articles.

2. Editing of technical books such as "The Radio Amateur's Handbook," "The ARRL Antenna Book," and other specialized technical books.

3. Answering members' technical information questions.

4. Checking products to be advertised in QST.

Some other functions of the department include participation in Institute of Electrical and Electronics Engineers (IEEE) conferences and operating the Technical Advisor (TA) program. An immediate addition to the list is to work with the new Technical Coordinator (TC) who reports to the new Section Manager (SM) in each ARRL Section. So it's a full platter. - W4RI.

Housing Information for 2nd ARRL Packet Conference

The following hotel is recommended for out-of-town packeteers attending the ARRL Packet Radio Conference:

Holiday Inn/Civic Center
50 Eighth Street
San Francisco, CA
415-626-6103

An alternate selection which is nearby:

Carriage Inn (Best Western)
140 Seventh Street
San Francisco, CA
415-552-8600

Both of these hotels are within walking distance of the Civic Center, and prices for rooms are in the 60 dollar range. The Computer Faire has reserved a large block of rooms at these hotels, and the simplest way to confirm your reservation is to send a letter (or the form in the back of the Silicon Gulch Gazette) to:

Computer Faire Housing Bureau
San Francisco Convention & Visitors Bureau
P.O. Box 5612
San Francisco, CA 94101
415-974-6900

They will be able to book rooms until 5 days before the Faire (the February 17th deadline in the Gazette is incorrect) at which time the rates expire and the space reverts to the hotels' control. Reservations are valid until 6 P.M. unless you have called the hotel with your credit card number. Write your reservation letter today, because these rooms are going fast. There were 30 rooms available as of February 17th. You can call the hotel direct, but the rate may be 10-15% higher. Call the hotel direct for cancellations or any changes.

After arriving at SFO Airport, take an Airporter Bus marked "Knob Hill - Financial District Express" and ask to be let off at the Americania. Fare will be $4.00. Regular taxi fare is about $24.00 from the airport.

The San Francisco Radio Club has agreed to co-host the meeting and is offering the services of its repeater W6PW (144.55/145.15) to all packeteers. If you need help or information, place a call on that machine.

The schedule for the meeting includes these special events:

3/18	9 A.M.-6 P.M.	Faire meetings and exhibits
3/18	7 P.M.- ?	Dinner for speakers and ARRL Digital Committee members
3/19	9 A.M.-6 P.M.	Packet Conference
3/19	Noon-1 P.M.	Informal Luncheon at Salmagundi's Restaurant
3/19	7 P.M. +	No formal plans - on your own
3/20	9 A.M.	Breakfast at Zim's Restaurant
3/20	10 A.M.-Noon	ARRL Digital Committee Meeting

(continued on page 5)

Correspondence

QEX Constraints

My compliments to you on QEX! I have enjoyed my subscription immensely. Special kudos to G.M. Palmer, K8LG for the "PROM Programmer/Reader & Software" feature in February 1983. That was a perfect example, in my opinion, of why ARRL needs QEX. The article was perhaps too technical for QST, but will be a great help to those of us indulging in hardware development. A fine article.

I note with concern, however, some statements made on the first page of that same issue. We are informed that budget constraints preclude the inclusion of more than one article of feature length in each issue of QEX. I wonder if such a tight restriction on the flow of information is consistent with the purpose of QEX.

It seems to me that such a stringent limit on the number of articles per month greatly reduces the utility of the magazine. After all, the magazine is called the "ARRL Experimenters' Exchange." With a limit of one feature article per month there can be very little exchanging going on!

We might consider the plight of some of our experimentally inclined bretheren, waiting breathlessly for their favorite technical subject to be treated at length in QEX. Assuming that there are perhaps 24 (an absurdly low figure) subjects which cry out for treatment in QEX, they can look forward to seeing ONE article on their pet subject sometime in the next two years! Now, how many people might we reasonably expect to subscribe to our magazine, given the likelyhood of their favorite subject being covered only once during each two year period? Not all that many I would suppose.

I would like to encourage you to investigate alternatives to the imposition of such a limitation. I for one would be receptive to the imposition of higher subscription fees if the content of the magazine could be enlarged. With so many varied and exciting developments taking place in Amateur Radio today, I feel that we ought to be fostering as much communication as possible between the developers of circuits and software and those who are either working on similar problems or who are anxious to put the latest developments to work. Expanding the number of feature articles in QEX would help foster this communication and help to draw more subscribers to an already interesting medium exchange.

Please do keep up the good work. - David S. Day, WD4FWP, 509 Coach Lane, Temple Terrace, FL 33617.

Digital Voice Modulation

I was pleased to see an article on digital voice ("Digital Voice Modulation," by G. W. Horn, I4MK) in the January 83 issue of QEX. Digital voice will certainly be part of the future of amateur radio. However, there were some errors and misleading statements in the article which should be cleared up, particularly with regard to LPC (Linear Predictive Coding), and other vocoders.

The name "vocoder" is short for voice coder and means a coder which employs knowledge about the human voice and its characteristics to gain an advantage in the coding. LPC analysis/synthesis systems are definitely vocoders, because they use filtering techniques to model the human voice production mechanism. There are many other types of vocoders, such as channel vocoders, phase vocoders, phoneme vocoders, etc. A vocoder may be either analog or digital.

A typical data rate for LPC and most vocoders is 2.4 kbits/s, a rate which is common because 2400 bauds is a standard data rate for modems and (of greater interest to hams) because the military uses sophisticated modems to transmit 2.4 kbits/s over hf radio links. Such equipment is fairly new, even to the military, so don't expect to find any at your local surplus house any time soon.

The quality of any vocoder increases with increasing data rate and vice versa. Phoneme vocoders, currently found only in laboratories, can encode speech at rates of a few hundred bits per second or so, but the resulting quality is poor.

It is definitely not true that with vocoders there is no way of recognizing the speaker. Any good vocoder will allow speaker recognition at 2.4 kbits/s. Since most vocoders detect and transmit the pitch of the voice as part of their human voice model, it is possible to produce speech with a constant, monotone pitch using a vocoder, and this is sometimes done for special effects in movies and TV shows. Perhaps this is what I4MK has heard.

I certainly would not call CVSD at 8 or 9 kbits/s "commercial telephone quality." Generally 32-kbit/s CVSD is considered to be "toll quality." Some newer 16-kbit/s systems may be reasonable, but I would rate all of the 8-kbit/s CVSD systems I have heard or worked on as "poor to fair."

The VLSI revolution may bring the cost of low data rate speech within the reach of the amateur radio operator in the next decade or so. Recently a signal processing oriented microprocessor (the TI TMS 320 series) was announced which is fast enough to implement a complete LPC vocoder. TI and several other manufacturers already make single-chip LPC synthesizers.

Further information on digital voice in general and packet voice in particular can be found in my paper entitled "Packet Voice: When It Makes Sense," published in Speech Technology magazine, vol. 1, No. 3, September/October 1982. - Randy Cole, KN6W, Information Sciences Institute, University of Southern California, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90291-6695.

Experiences Building a UVEPROM Eraser

Some of the articles in popular magazines have contained misleading information on the proper bulb. The common "black light" tube, designated BLB, emits insufficient UV (ultraviolet) at the required wavelength -- 2537 angstroms. Thus, the PROM must be set close to the tube for so long a time that it is almost too hot to handle before complete erasure is accomplished.

The correct bulb to use carries a "G" nomenclature (for germicidal) and can be purchased in several wattages. The one I found satisfactory is marked G8T5. It is an 8-watt, tube about 11 in. long and fits a T5 fluorescent fixture. Prices in the Washington, DC area for the tube varied from $16 to nearly $25. A lower wattage would also be satisfactory and probably cheaper, but I happened to have a suitable 8-watt fixture on hand. However, if the budget does not allow purchase of a fixture, it is possible to buy the proper socket, ballast and starter and make your own.

I secure PROMs to be erased about 1 in. from the tube. My device will erase up to 10 chips at one time, and 30 minutes exposure has always been adequate. Be sure to enclose the tube in some way so that one's eyes are not exposed to the emitted light as it can damage the eyes. - Jim Simpson, K4NWE, 7721 Weber Ct, Annandale, VA 22003.

Packet Radio and Radio Communication Requirements

By Karl Meinzer,* DJ4ZC

Introduction

The recent adoption of [the amended level 2 draft of] the AX.25 [1] standard for radio-amateur packet radio has been a significant milestone towards a more widespread use of digital communication. So far efforts concentrated on the digital and logical requirements of this new form of communication. Adopting essentially the existing [CCITT] X.25 standard has some potential problems, though, when facing the realities of Amateur Radio. The standard had been designed for the "good" communication links normally used in the commercial environment. With amateur signals often fading into the noise and with QRM being quite frequent, the standard may be less than ideal.

Sync-Vector Requirements

When the author developed the AMSAT Phase III digital communication techniques, a blocked format was found to present the best overall properties. This means that each valid transmission consists of a block of constant length (512 bytes with Phase III), and in between these blocks some idle characters are being sent. The satellite uses synchronous transmission; this means that no start and stop bits are used and that the data stream is continuous. There are a number of ways to recover the bit clock; in the case of Phase III biphase encoding of the data was adopted to simplify the process on the ground.

Synchronous communication has many advantages over the older nonsynchronous forms. The data rate is effectively 20% higher. More important, though, is the possibility to use more modern modulation techniques having considerably better weak-signal performance.

Because of the continuous nature of the data stream it becomes necessary to identify the beginning of the individual bytes and the entire block of valid data. When the signal/noise ratio is good and data errors are an infrequent occurrence, the recognition of one particular byte in the data stream usually is sufficient to get this information. Most of the ICs commercially available for synchronous communication thus provide the capability for recognizing 1 or 2 bytes as the so-called "sync vector"; and the X.25 standard also uses the "flag" byte for this purpose.

If the link performance may get very poor temporarily, e.g., in space communication or with Amateur Radio, one or two bytes for synchronization are not adequate. In this environment it makes sense to require that sync (the beginning of a block) is declared only if the block start is positively identified. If the receiver is just getting noise, the modem will output random data, and the probability of this random data declaring "sync" should be reasonably small to prevent this false sync masking the real sync vector possibly coming a little later. (This assumes that no squelch can be utilized in the receiver -- the are no practical ways to design an effective squelch with the better communication modes requiring low signal levels.)

The time between declaration of false sync on random data is $t = 2^n$/data rate, n being the length of the sync vector in bits. With n = 8 every 256 bits (about once every 1/4 second at 1000 bit/s) a false sync is declared. Even with a sync vector of 16 bits long (2 bytes) about once per minute a false sync will be "recognized." Because of this problem AMSAT adopted a 4-bytesync vector for the Phase III satellites. NASA also uses 4-byte vectors in their telemetry transmission. With the 4-byte vector the probability of false sync is so low that only every 50 days or so a false sync will occur. Since at the end of the block a CRC normally will be used to identify errors, this performance is quite satisfactory.

Modulation Formats

Most high-performance data transmission systems use some form of phase-shift keying (psk) because these modulation systems offer excellent Eb/No performance for vhf and uhf data transmission. The receiver for these formats must reconstruct carrier and bit-rate phase and frequency for proper operation. This reference information then is used by the circuitry to demodulate the received data.

It should be noted that the quality (absence of phase jitter) of the reconstructed carrier determines if the theoretical predictions as to the Eb/No performance can be obtained. Without going into the mathematics it can be stated here that for 2-phase modulation formats the effective two-sided bandwidth of the carrier-recovery PLL should not exceed the data rate/4. With 4-phase techniques (e.g., qpsk or msk) this PLL bandwidth must be reducted to the data rate/100. (With 2-phase systems the signal degradation is proportional to the cosine of the phase error; with 4-phase systems the degradation results from quadrature-channel crosstalk proportional to the sine of the phase error.)

The loop bandwidth determines the time required for the receiver PLL to become locked after the transmitter is keyed on. A very careful PLL design is necessary in order to make this time to lock short enough so that the transmitter on time is not governed by receiver acquisition requirements rather than the data content of the packets. In the design of the Phase III modem (the so-called AFDEM) a 3-level search strategy was necessary to accomplish a time to lock of 5 seconds (with a 5-kHz search range and two-sided loop bandwidth of 100 Hz).

With coded transmission the acquisition time becomes even more constraining. This situation clearly favors 2-phase psk (which also is simpler to implement) because the total amount of data transmitted in a given time may be larger even if the data rate is cut in half because the receiver lock-up time is so much shorter.

Formats for Coded Transmission

Although not known very much to amateurs, there are simple ways to considerably improve the weak-signal performance over the capabilities offered by straight transmission of data. These methods make use of the prediction of the communication theory of Shannon that the performance may be improved by redundancy, i.e., transmitting more information than just the desired data and using this information to remove errors.

Typically the data stream transmitted is doubled (or also sometimes tripled) by producing two symbols from each data bit using a simple shift register and exclusive-OR gate arrangement (a so-called convolutional coder). This operation also can be performed entirely by software.

Recovering the data from the symbols is the

(continued on page 5)

*AMSAT-DL e.V., Hoehenweg 38, D-355 Marbach, Hessen, West Germany

Bibliography on Minimum-Shift Keying

By Den Connors,* KD2S

The Amateur Radio Service will be undergoing a transformation as personal computers bring the need for high-speed digital communications into the ham shack. The modems now heard on the air will be woefully inadequate to provide bandwidth-efficient data transmission. This has led a number of amateurs to look into synchronous modulation techniques, and in particular, minimum-shift keying (msk).

Msk was first patented as a "Minimum-Shift Data Communication System" in March of 1961 by Melvin Doolz and Earl Heald of the Collins Radio Company. Known alternately as fast-frequency-shift keying (ffsk) in the early literature, the technique provided the minimization of bandwidth the researchers were looking for, using simple frequency-shift keying. The price they paid was in the complexity of the synchronous demodulators.

Msk can be descibed alternately as frequency-shift keying with a fixed modulation index of 0.5, or as phase-shift keying of a digital signal with Gaussian pre-shaping of the signal input to the modulator. This very cryptic non-explanation may be clarified by digging into the available literature.

General Articles

There are several articles which can give a general overview of the technique. Pasupathy presented msk in an easy-to-read format for the IEEE in [PAS79]. A more-complete technical presentation was done by Rudi De Buda [DEB72], [DEB74]. Morais and Feher [MOR79] introduce investigations of the effects of noise on msk, and Gronemeyer and McBride fold in similarities of msk and psk in their 1976 article [GRO76].

[DEB72] R. DeBuda, "Coherent Demodulation of Frequency-Shift Keying with Low Deviation Ratio," IEEE Trans. on Communications, Vol. COM-20, No. 6, pp. 429-35, June 1972

[DEB74] R. DeBuda, "About Optimal Properties of Fast Frequency-Shift Keying," IEEE Trans. on Communications, Vol. COM-22, No. 10, pp. 1726-7, October 1974

[MOR79] D. Morais & K. Feher, "Bandwidth Efficiency and Probability of Error Performance of MSK and Offset QPSK Systems," IEEE Trans. on Communications, Vol. COM-27, No.12, pp. 1794-1801, December 1979

[GRO76] S. Gronemeyer & A. McBride, "MSK and Offset QPSK Modulation," IEEE Trans. on Communications, Vol. COM-24, No. 8, August 1976

[PAS79] S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," IEEE Communications Magazine, Vol. 17, No. 4, pp. 14-22, July 1979

Comparisons with Other Techniques

There are several articles which compare msk and other modulation techniques, with special emphasis on usable signal-to-noise ratios and required bandwidth. The canonical paper is by John Oetting [OET79], who compares all known modes, and provides a bonus of a extensive bibliography on all of the techniques. Feher and his group have looked at the problem of use on satellites in [LEN82]. A few other pertinent comparisons are included here.

*4708 West Wild Horse Drive, Tucson, AZ 85741.

[HIR81] W. Hirt & S. Pasupathy, "Suboptimal Reception of Binary CPSK Signals," IEE Proc., Vol. 128, Pt. F, No. 3, pp. 125-34, June 1981

[LEN82] T. Le-Ngoc, K. Ferer & H. Pham Van, "New Modulation Techniques for Low-Cost Power and Bandwidth Efficient Satellite Earth Stations," IEEE Trans. on Communications, Vol. COM-30, No. 1, January 1982

[MIL81] L. Milstein, D. Shilling & R. Pickholtz, "Comparison of Performance of 16-ary QASK and MSK Over a Frequency Selective Rician Fading Channel," IEEE Trans. on Communications, Vol. COM-29, No. 11, November 1981

[OET79] J. Oetting, "A Comparison of Modulation Techniques for Digital Radio," IEEE Trans. on Communications, Vol. COM-27, No. 12, pp. 1752-62, December 1979

[TEY79] W. Tey & T. Tjhung, "Characteristics of Manchester-Coded FSK," IEEE Trans. on Communications, Vol. COM-27, No. 1, pp. 209-16, January 1979

Filtering

The utility of msk has been shown to improve with adequate pre-filtering of the input signal. Morais and Feher [MOR80] give an excellent overview of the effects of filtering and limiting. [BAY79] and [RAB78] discuss bandwidth shaping, and [MUR81] and [SUZ81] explore the use of Gaussian filtering.

[BAY79] J. Bayless & R. Pedersen, "Efficient Pulse Shaping Using MSK or PSK Modulation," IEEE Trans. on Communications, Vol. COM-27, No. 6, June 1979

[MOR80] D. Morais & K. Ferer, "The Effects of Filtering and Limiting on the Performance of QPSK, Offset QPSK, and MSK Systems," IEEE Trans. on Communications, Vol. COM-28, No. 12, December 1980

[MUR81] K. Murota & K. Hirade, "GMSK Modulation for Digital Mobile Radio Telephony," IEEE Trans. on Communications, Vol. COM-29, No. 7, pp. 1044-50, July 1981

[RAB78] M. Rabzel & S. Pasupathy, "Spectral Shaping in Minimum Shift Keying (MSK)-Type Signals," IEEE Trans. on Communication, Vol. COM-26, No. 1, pp. 189-95, January 1978

[SUZ81] H. Suzuki, "Optimum Gaussian Filter for Differential Detection of MSK," IEEE Trans. on Communication, Vol. COM-29, No. 6, June 1981

Implementation

Although the best implementation discussions are found in proprietary technical manuals of systems built by several corporations, a few articles do describe some details of construction. [AMO78], [CRU78] and [MAT78] present some practical hardware notes. Masamura et al. [MAS79] show a different technique using differential detection, which needs a higher signal-to-noise ratio to work as well as straight msk synchronous detection. This differential technique has the benefit of providing error correction, however, and could be used in a number of weak-signal or fading situations.

[AMO77] F. Amoroso & J. Kivett, "Simplified MSK Signalling Technique," IEEE Trans. on Communications, Vol. COM-25, No. 4, pp. 433-41, April 1977

[CRU78] J. Cruz & R. Simpson, "Minimum-Shift-Keying Signal Detection with Noisy Reference Signals," IEEE Trans. on Communications, Vol. COM-26, No. 6, pp. 896-902, June 1978

[MAS79] T. Masamura, S. Samejima, Y. Morihiro & H. Fuketa, "Differential Detection of MSK with Nonredundant Error Correction," IEEE Trans. on Communications, Vol. COM-27, No. 6, pp. 912-18, June 1979

[MAT78] R. Matyas, "Effect of Noisy Phase References on Coherent Detection of FFSK Signals," IEEE Trans. on Communications, Vol. COM-26, No. 6, pp. 807-15, June 1978

[MORI79] Y. Morihiro, S. Nakajima & N. Furuya, "A 100 MBit/s Prototype MSK Modem for Satellite Communications," IEEE Trans. on Communications, Vol. COM-27, No. 10, pp. 1512-8, October 1979

This bibliography is by no means complete, and all of the articles mentioned above will lead the reader to more specialized papers.

Packet Radio and Radio Communication Requirements
(continued from page 3)

more complex part of the operation. There are two different methods in widespread use. The so-called Viterbi decoding requires special and fairly complex hardware and is used mostly with continuous data. For amateur use this technique is probably too complex. For blocked data there is another technique called sequential or "Fano" decoding. This decoder can be implemented by software and requires little or no change in the receiving hardware.

AMSAT-DL implemented Fano decoders on the Radio Shack TRS-80 (tm) Model I and the Color Computer as a backup for the Phase III satellite -- should something make the links poorer than calculated. These decoders are fast enough to decode a few-hundred bits/s in real time.

The performance of these decoders is such that they can recover a block of data if the total number of symbol errors does not exceed 3 to 3.5%. This means that an Eb/No of 2 to 3 dB at the symbol rate is sufficient, while normally 10 to 11 dB Eb/No is required for acceptable performance. Because the data rate is only 1/2, there is a net gain of about 5 dB - and all that is required is a short program in the computer which is needed anyway.

In view of this situation it would be a good idea not to preclude the use of coded transmission for packet radio. These considerations should be taken into account:

- An even longer sync vector is needed to allow a probabilistic recognition of this vector. AMSAT-DL uses an 8-byte vector - if 60 of the 64 bits are correct, symbol sync is declared.

- Differential coding (which normally is used with psk to eliminate the sign ambiguity of the carrier) cannot be used. For this reason the sync-recognition program also looks for the inverse sync vector and sets a sign flag accordingly.

- A CRC is not needed at the end of the block. Either the decoder can decode the received block - there is a negligible probability of error - or the decoder will declare the block as undecodable in the time allocated for decoding.

Conculsions

Some of the properties of the advanced synchronous communication techniques giving the best-known weak-signal performace were discussed. In order not to preclude the use of these techniques in the emerging amateur-packet-radio systems it is recommended that standards with the following properties be adopted:

- Sufficiently long sync vectors (4 bytes for uncoded, 8 bytes for coded transmission) should be adopted. The AMSAT vectors allow both simple hardware and software recognition.

- A fixed block length should be adopted to make the block end uniquely identifiable even in the case of bad error rates. This also allows making the blocks entirely transparent, i.e., arbitrary bytes (like binary data) may be included in the data of the block.

- Synchronous techniques should be recommended using efficient modulation techniques. Four-phase techniques (like qpsk or msk) should be used for data rates in excess of 10 kbit/s only because otherwise the carrier-recovery loop bandwidth becomes too narrow and the time to lock becomes excessive.

The author hopes that these hints based on the experience gathered ruing the development of the digital communication system of the AMSAT Phase III satellites will help to implement state-of-the-art communication technology with amateur packet radio. The price for this is only the understanding of the issues - realizing these ideas for the most part requires only appropriate software and is well with the reach of amateurs due to the widespread availability of low-cost microcomputers.

Reference

[1] Magnuski, "Agreement on Packet Radio Standards," QEX 12 October 1982 and ASR.

[Ed. Note: This article gives excellent insight into Karl's design philosphy that he used in developing the AMSAT Oscar Phase III telemetry and telecommand data system. QEX readers should be aware that other designers involved in packet-radio development are working with variable-length frames, different modulation formats, various sync techniques and other binary baseband encoding schemes. The proposed AX.25 standard is only in draft form and is subject to change in the reviewing process. Articles or correspondence on packet-radio techniques and proposed standards are invited for publication in QEX.]

Packet Conference
(continued from page 1)

In addition to the ARRL Conference, Hank Magnuski and Paul Rinaldo will be speaking on Packet Radio during regular Faire sessions. As of this date the exact times of these presentations have not been confirmed.

Preregistration special rates expire March 7th. I should have your check for $19 (Faire registration plus conference proceedings) in hand no later than March 1st, or you may not be able to get the special rate.

If you want more general information about the Faire, call or write Faire HQ and ask to be sent a copy of the Silicon Gulch Gazette:

Computer Faire
345 Swett Road
Woodside, CA 94062
415-851-7077

Hope to see you in March! - Hank Magnuski, KA6M.

ZX81 RTTY Receive Program

By Brian Davis,* W9HLQ

The ZX81 is an excellent single-board microcomputer. The Z80A CPU has a powerful instruction set. The Sinclair running at 3.5-MHz clock speed can develop into an excellent and fast controller for the ham shack for many purposes limited only by our imagination. And -- the price is right! My purpose in publishing this program is to encourage owners of the Timex/Sinclair computer to investigate the capabilities of their unit. Then they can use this information as a starter to expand upon.

This program is simple to use and requires a very simple interface connection to the computer. The RTTY Receive program will run only in the "fast" mode. It will collect a block of incoming RTTY characters in 60-wpm Baudot. When the BREAK, space bar, is pressed the computer will stop and display the received text. While viewing the text some characters will be missed. Entering RUN will direct the computer to go back and collect more characters. By alternately BREAKing and RUNning, most of the incoming text can be read. Continuous display of received text would be very desirable, but it will require additional hardware to assist the computer. I will leave that project for someone else to implement. How about it?

Notes on Using This Program

1) Enter the BASIC program as shown.

2) The BASIC routine at 2000 will be used temporarily to load the machine language (ML) into the 1 REM. The 2 REM is a translation table that is 64 positions long. The first 32 positions are for LETTERS shift, and the second 32 are for FIGURES shift. You can EDIT this table to make corrections or to add any special characters that I haven't provided for here. Use Table 2 provided to enter the 2 REM translation table.

3) Enter the ML program into 1 REM by typing RUN 2000. You will be prompted to enter a number from Table 1; one at a time. The BASIC program will then POKE this number into 1 REM at consecutive slots. After the ML is in, delete statements 2010 and 2020 and type RUN 2000 again to list all the ML you put in. When you verify that it is correct, delete the 2000 series as it is no longer needed. Note: If you need to make a correction use the "immediate mode" to POKE in any corrections. For example: POKE 16514,21 (without providing a sequence number) will allow you to make singular changes without having to enter all of the ML codes again.

4) SAVE several copies of the program as you go along and before you run the program the first time. An error in the ML program may cause a system "crash," and the only way out is system reset (power off, power on) and a reload of the program. The BREAK won't work here.

5) The input interface is simple but does require opening up the computer. By grounding pin 20 of IC 1, we can then "read" the incoming pulses by having the ML issue the Z80 "IN A,n" instruction. See Diagram 1 for input wiring diagram. IC 1 is the special Sinclair IC located adjacent to the rf modulator box near the rear of the board. I talk-soldered one wire to pin 20 and the other to a convenient ground. Be careful here. Programming errors can't damage a computer, but wiring errors can!

*17038, Oconto Ave, Tinley Park, IL 60477, 312-532-5833.

6) I used a small relay driven by the TTY loop to isolate and key the computer. It was satisfactory for 60 and 67 wpm. An optoisolator is highly recommended here to drive the computer, especially for 100-wpm Baudot and higher ASCII bit rates. See Diagram 2.

7) Fine tuning of the ML delay loops may be necessary to optimize operation. This will compensate for varying relay ballistics and also allow the switch to other speeds. Here are the values that work for my system as driven by a small relay:

Speed (wpm)	Delay 1	Delay 2
60	POKE 16708,25	POKE 16724,18
67	POKE 16708,20	POKE 16724,15

Other speeds probably could be determined by decreasing the delay values in proportion to the increase in speed. I couldn't verify this since my relay couldn't follow higher speeds reliably.

8) Note: The logic used for this system dictates that pin 20 be grounded during mark (loop current flow). You cannot LOAD while pin 20 is grounded. To LOAD you must open the loop or otherwise unground pin 20. Note also that if you stop receiving RTTY you will be in a mark condition. The ML program will be looping and looking for a space. During this time you cannot BREAK into the program. By simultaneously pressing the BREAK (space) and having incoming RTTY pulses you can BREAK any time. Otherwise, you may want to install and activate Switch 1 to allow a BREAK into the program.

Program Description

Machine Language - See the assembly listing for comments on the ML program. Note the location in the program for the delay values that you may want to adjust. The are identified as D1 and D2 respectively. The ML program loops indefinitely at LOOP1 looking for the start bit. When the mark level drops, we fall through to start collecting subsequent bits of data. Delay 1 delays across the start pulse into the middle of the first data bit. The data bits are available because we have issued another "IN A,n" instruction. If present, the bit is shifted out of the A register into CARRY and then into the C register where they are accumulated. When all 5 Baudot bits have been accumulated, a return (RET) to BASIC is done. The bits accumulated in the C register are available now in the BASIC variable "Z." This value is now used as an index (offset) into the lookup table for translation into displayable characters.

BASIC

50	Set shift for LTRS
100	Call to ML to get a character value
110	If Z=0 IDLE character received: throw away.
120	If Z=4 SPACE: set J=0 unshift on space.
130	If Z=27 FIGS shift: Go set J=32 to use back part of table.
140	If Z=32 LTRS shift: Set J=0 to use front part of table.
150	Print the character found in the table as pointed to by sum of Z and J.
160	Go back for next character.
170-200	SHIFT set routines.

→

```
   1 REM 0123456789012345678901234567890123   (34 POSITIONS)
   2 REM ---        (SEE TABLE 2 FOR VALUES)
  50 LET J=0
 100 LET Z=USR 16514                          CALL ML
 110 IF Z=0 THEN GOTO 100                     IDLE-THROW AWAY
 120 IF Z=4 THEN LET J=0                      SPACE-USOS
 130 IF Z=27THEN GOTO 170                     GO SET FIGS SHIFT
 140 IF Z=31 THEN GOTO 190                    GO SET LTRS SHIFT
 150 PRINT CHR$(PEEK(16553+Z+J));             LOOK UP CHAR & PRINT IT
 160 GOTO 100                                 GO BACK TO GET NEXT CHAR.
 170 LET J=32
 180 GOTO 100
 190 LET J=0
 200 GOTO 100
2000 FOR I=16514 TO 16514+33                  ML LOAD/VERIFY ROUTINE
2010 INPUT N
2020 POKE I,N                                 (DELETE 2XXX SERIES WHEN
2030 PRINT I;"=";PEEK I,                       ALL IS OK)
2040 NEXT I
9000 SAVE "RTTYRX"
9010 GOTO 9000
```

TABLE 1

* ENTER THE DECIMAL VALUES INTO THE REM NUMBER 1 BY ENTERING "GOTO 2000".

ADDRESS	DECIMAL*	HEX	
16514	14	0E	
15	0	00	
16	6	06	
17	5	05	
18	219	DB	
19	0	00	
16520	230	E6	
21	128	80	
22	32	20	
23	250	FA	
24	22	16	
25	25	19	(D1 - DELAY 1)
26	30	1E	
27	255	FF	
28	29	1D	
29	32	20	
16530	253	FD	
31	21	15	
32	32	20	
33	248	F8	
34	219	DB	
35	0	00	
36	203	CB	
37	39	27	
38	203	CB	
39	17	11	
16540	22	16	
41	18	12	(D2 - DELAY 2)
42	5	05	
43	32	20	
44	237	ED	
45	201	C9	
46	0	00	
47	0	00	
16548	0	00	

==

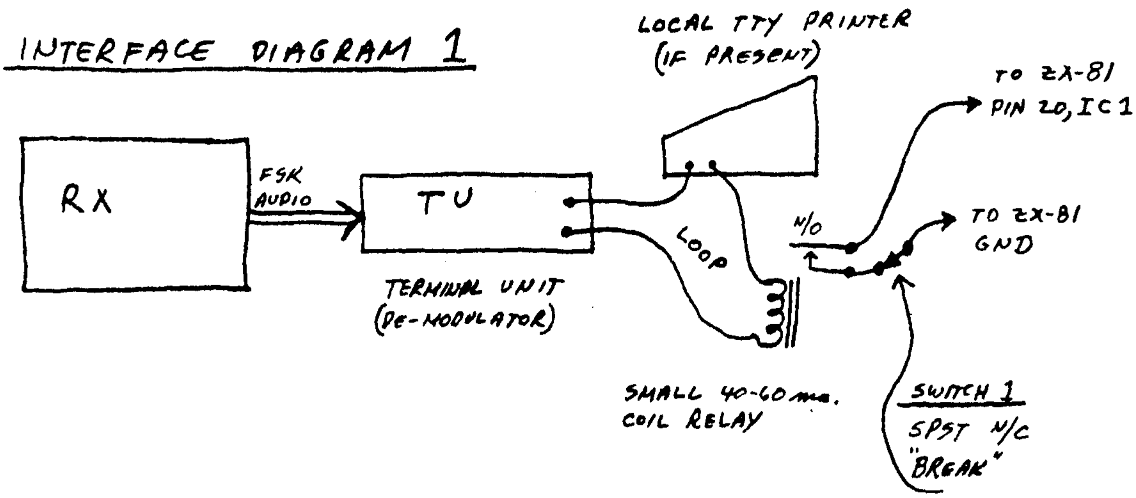
TRANSLATE TABLE TABLE 2 (FOR 2 REM)

ENTER/EDIT THE FOLLOWING CHARACTERS INTO THE "2 REM". FOR EASE OF READING THE TABLE IS BROKEN INTO GROUPS OF 10. DON'T ENTER THE SPACES BETWEEN THE GROUPS. THERE ARE 64 CHARACTERS IN THE TABLE. AN ASTERISK REPRESENTS A SPACE (BLANK). SEVERAL BAUDOT CHARACTERS ARE MISSING FROM THIS TABLE AND YOU CAN EASILY EDIT THEM IN AT YOUR CONVENIENCE.

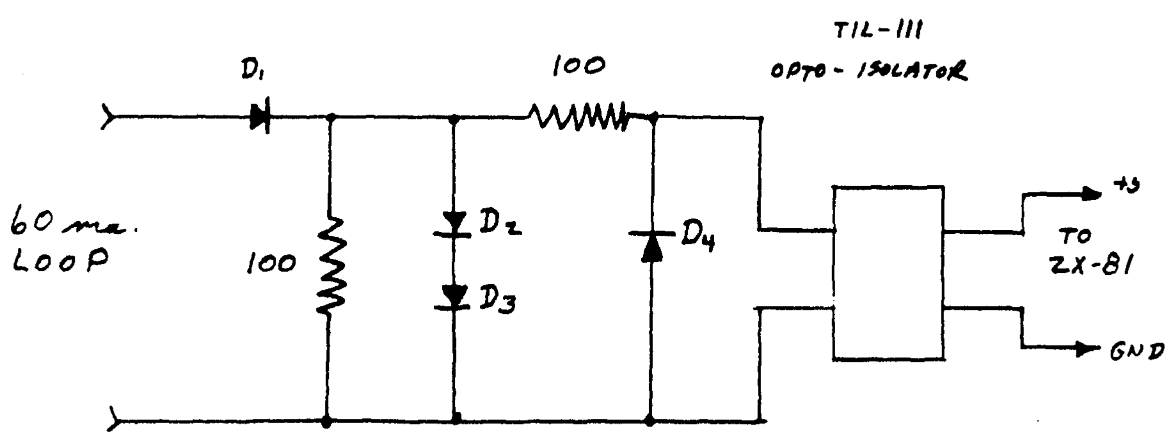
```
*T*O*HNM*L  RGIPCVEZDB
SYFXAWJ*UQ  K**5*9**,.
*)4*80**3"  ***6*/-2**
71(*
```

==

INTERFACE DIAGRAM 1



SUGGESTED OPTO-ISOLATOR CIRCUIT - DIAGRAM 2



D_1 - 250 PIV SILICON DIODE

D_2-D_4 SMALL SIGNAL SILICON DIODES

(DESIGN BY KEN HEITNER WB4AKK)

```
15514 0E00    OUTPUT  LD    C,0     CLEAR REGISTER C
16516 0605            LD    B,5     SET BIT COUNT
16518 DB00     LOOP1  IN    A,0     LOOK FOR START BIT
16520 E680            AND   A,"80"  TEST FOR BIT
16522 20FA            JR    NZ,-6   LOOP TO LOOP1 IF NO START BIT
16524 1619            LD    D,25    SET DELAY 1  (D1)
16526 1EFF     LOOP3  LD    E,255   INNER LOOP COUNT
16528 1D       LOOP2  DEC   E       REDUCE E BY ONE
16529 20FD            JR    NZ,-3   LOOP TO LOOP2
16531 15              DEC   D       REDUCE OUTER LOOP BY ONE
16532 20F8            JR    NZ,-8   LOOP TO LOOP3 IF NOT ZERO
15534 DB00            IN    A,0     LOOK FOR DATA BIT
16536 CB27            SLA   A       SHIFT BIT TO CARRY
16538 CB11            RL    C       SHIFT BIT INTO C REGISTER
16540 1614            LD    D,20    SET DELAY 2   (D2)
16542 05              DEC   B       REDUCE BIT COUNT BY ONE
16543 20ED            JR    NZ,-19  LOOP TO LOOP3 UNTIL ALL 5 BITS ARE IN
16545 C9              RET           RETURN TO BASIC
16546 00              NOP
16547 00              NOP           PROVIDE ROOM FOR PATCHING
16548 00              NOP
```

RTTY RECEIVE

W9HLQ

ZX 81
8K ROM

ENTER FROM BASIC

WAIT FOR START. LOOP BACK ON MARK; FALL THRU ON SPACE

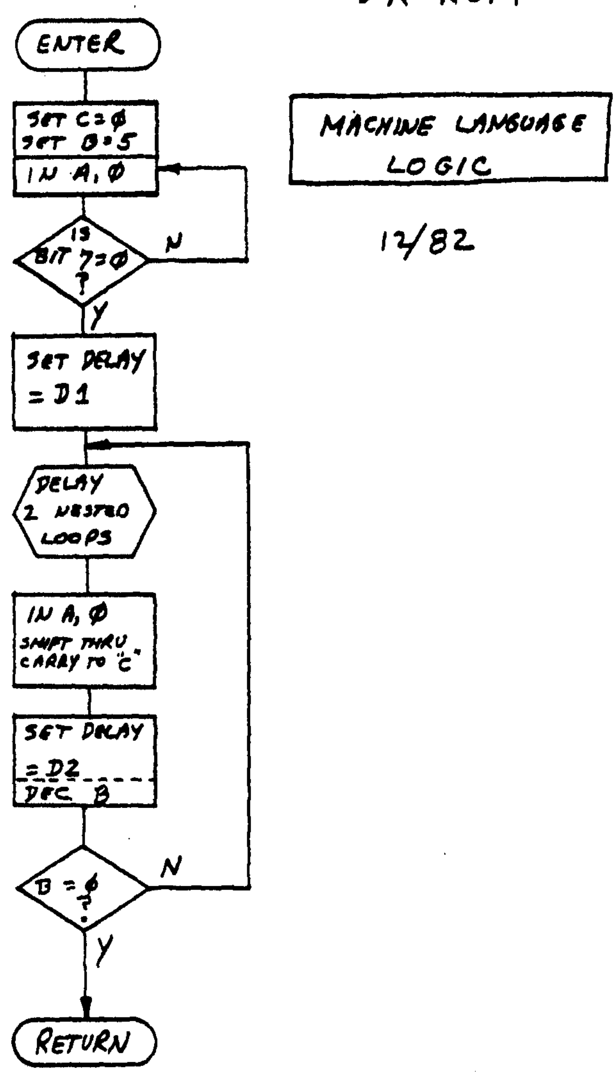
DELAY OVER START BIT AND INTO MIDDLE OF 1ST DATA BIT.

READ INPUT

USE NEW DELAY FOR REST OF DATA BITS

ARE ALL 5 BITS ASSEMBLED YET?

GO BACK TO BASIC. PASS ASSEMBLED CHAR. BACK IN "BC" REGISTER



Data Communications

Conducted by
David W. Borden,* K8MMO

Digital Phase Lock Loops

There is a study being undertaken by Paul Rinaldo, W4RI to determine what to do about sync in an hf environment. Traditionally, this is done with bit reversals, transistions from 1 to 0. The transmitter sends them and the receiver clocks on them, when both clocks agree, synchrony is achieved. Paul's problem, as a designer of hf modems, is hf is noisy. Since our current packet radio technology is based on synchronous operation, it behooves us to study the clock recovery scheme we are using and how to optimize it.

First, let us review the data encoding scheme. In our packet work we use NRZI (Non-Return-Zero-Inverting) encoding in which a 1 is represented by no change in level and a 0 is represented by a change in level. Thus to achieve reversals using NRZI encoding our initial starting data must be 00000000 to produce 10101010. The data is restored at the receiver by differentiating the data which extracts impulses which are used to repeatedly trigger a timing circuit. The timing circuit outputs a zero bit for one bit time when triggered, thus the data is restored to orginal form. The purpose of this encoding scheme is to eliminate long strings of zeros which would be detrimential in our synchronous operation of asynchronous modems. Long strings of ones are prevented in HDLC by bit stuffing a zero every five ones. Now that we have these nice transistions going for us, we can recover clock by them as below.

Three chips are in use currently in the amateur community for packet radio. The Intel 8273 chip is used on the Vancouver Digital Communication Group TNC board. The Zilog 8530 chip is used on the AMRAD PAD board (under development). The WD-1933 chip is use by the Tucson Amateur Packet Radio group. We should examine how these chips recover clock by DPLL.

The 8273 protocol controller chip, introduced the digital radio amateur to modem clock recovery. A digital phase lock loop (DPLL) was provided on board the chip to allow clock recovery based on two inputs, the receive data (RxD) and a 32X (thirty two times the bit rate) clock. This device provided the chip with a receive data clock. Thus the modem did not have to supply it (which is good since we are again using asynchronous modems in synchronous service and have none to supply). According to the Zilog 8530 technical manual, the DPLL counts the 32X clock to create nominal bit times. While it counts, the DPLL searches the data stream. Whenever an edge is detected, the DPLL adjusts its count (during the next counting cycle) so that the terminal count is closer to the center of the bit cell." I call this pussyfooting around to find the center of the bit window. Adam Osborne really explained it well in his explanation of the 8273 protocol controller chip. He states that there are two problems associated with generating DPLL. First, initially the 8273 cannot know if the DPLL is correctly synchronized with RxD. Second, the 32X clock may not be exactly 32 times the serial data baud rate. He says this problem is solved in the 8273 by using every active transistion of RxD as a synchronization reference point for DPLL. There is our problem: on hf we have lots of transistions from the hefty noise component on our noisy channel. The 8273 is clever in pussyfooting. Each serial bit time is divided into four quarters, each 8 clock pulses wide. When DPLL is correctly synchronized with incoming RxD, transistions will occur between the second and third quarters of 32X clock. If the transistions occur between the first and second quarters of 32X clock, the DPLL must be advanced. Conversely, if the tranistions occur between the third and fourth quarter of 32X clock, the DPLL must be retarded. But, just like the correction of hot water in the shower, we can not go too far too fast. Correction should be made in the correct direction, advancing or retarding as indicated, but less than the indicated amount or sharp instability of the DPLL is encountered. This instability is noted if we do not damp our frequency variations in small steps, but rather attempt to do it all at once. Thus, a one-clock-pulse correction is made for an error of between one and eight clock pulses (second or third quarters) and a two-clock-pulse correction is made for an error of between 8 and 16 clock pulses (first or fourth quarter). Thus our pussyfooting is in small careful tiptoes and not big jumps. The worst case that Adam Osborne cites is the DPLL takes twelve active transistions to center. In this worst case, the device will detect transistions every 6th serial bit (the zero insertion!).

Thus, for the DPLL, we can make quite a case for having at least 12 transisitons at the start of each frame for our noisy circuit. The 8273 allows this in what they call preframe sync, 16 active transistions to allow the DPLL to pussyfoot around on and sync up.

The 8530 still requires further study, but the DPLL on board that device divides each nominal bit time into three regions, vice four for the 8273. The 8530 also makes small damped adjustments, but requires more time to lock up to the data stream. The DPLL could require up to 16 transistions on the line to count the sampling edge. To get around this problem, the 8530 has a special startup mode which has the DPLL sit at count 16 and wait for an edge. The first edge detected is assumed valid data (probably very true on 2 meters afsk fm). If it was valid data then the DPLL will sample correctly in the middle of the bit cell. If it was noise, the DPLL will take longer to lock on. Start-up mode is cancelled when the first edge is detected. At first blush, this method seems bad for hf use. It would suggest the 8273 is a better chip than the 8530 on hf. More study is required, a software workaround may be possible.

The final chip under study, the 1933, is more difficult to speak about since the available data sheet was not as clear on how DPLL was handled. Some of my friends will clear this up, but it appears from a WD-193X DPLL Timing Diagram I studied, they divide the bit cell into two areas of 32X clock. If data transistion occurs in the first half of 32X clock, the DPLL counter is decremented by one and if in the last half, the counter is incremented by one. No discussion of the need for damping was conducted on the data sheet I studied. Thus, more examination of this chip is required, but on the surface, it seems the least choice for hf work.

A disclaimer seems appropriate here. My assumption, in this study, is that it is best to pussyfoot in small steps where a bit correction is required so that glitches in the RxD line do not cause sharp instability in the DPLL. This assumption may be incorrect, and I await argument from anybody.

In software, we may be able to command any protocol chip to send transistions (as many as we want -- lots on hf) before the first flag byte of a frame. We also may be able to send lots of flags. You'll be hearing more about this subject as we gather more information and test results.

*Rte 2, Box 233B, Sterling, VA 22170.

Components

Conducted by Mark Forbes,* KC9C

I have just received the latest edition of National Semiconductor's New Product Update (Number 7). National publishes these about every quarter or so. I will pass along some data from this update in this and subsequent editions of "Components."

National COP420R/COP444LR Piggyback EPROM Microcontroller

For those of you who have been interested in the features of the National COP series of microcomputers but were unable to use them because they are mask programmable, the COP420R/COP444LR may be use the answer. The COP (for Control-Oriented Processor) family is a wide range of microcomputers intended for control applications. In general, they feature an 8-bit instruction word and operate on 4-bit data, as is common to many masked programmable "4-bit" computers.

The 420R and 444LR are different in that they may be programmed with a 2716 or 27C16 EPROM. The microcomputer itself is in a 40-pin DIP with a 28-pin DIP residing on the tope in piggyback fashion. Thus all the features of the COPs can be used at a fraction of the cost necessary to go to mask. Both units feature 23 I/O lines, have an internal timer/counter and operate from a single supply. The 420 has an instruction time of 4 us while the 444 requires 16 us per instruction.

Information on this and all other National products mentioned this month may be obtained from: National Semiconductor Corp., 2900 Semiconductor Drive, Santa Clara, CA 95051.

National DS8621 Vhf/Uhf Prescaler

The DS8621 is a low-power, high-speed prescaler, mainly intended for TV tuners, but having possible Amateur Radio usage as well. In the vhf mode, the input frequency is scaled by a factor of 64, and in the uhf mode the scale factor is 256. The inputs are fully buffered and have a sensitivity of 50 mV at frequencies over 275 MHz and 1.2 GHz for vhf and uhf respectively. These devices are pin-for-pin compatible with the RCA CA3179 and Motorola MC12071.

National MM74HC942 300-Baud Modem

The MM74HC942 is a 300-baud modem constructed in the new high-speed CMOS process. The '942 is Bell 103 compatible. Switched-capacitor techniques are employed for the filtering. The modulator section produces phase-coherent fsk. The line driver has a 600-ohm impedance and can perform two-to-four-wire conversion. The nine-pole filter in the receiver provides 60 dB of transmitted tone rejection. The device is contained on a single 20-pin IC. Contact National for pricing and availability.

*1000 Shanandoah Dr, Lafayette, IN 47905, 317-447-4272, 2300-0230 UTC weekdays, until 0230 weekends.

National LH0082 Optical Communication Receiver/Amplifier

For those of you interested in experimenting in optical communications, the LH0082 is a welcomed help. The unit can be used for digital communication with the on-board FET amplifier and comparator with hysteresis, or analog information can be sent to a separate analog input. In the digital mode, the LH0082 can be used at data rates up to 5 Mbits/s NRZ (nonreturn to zero encoding).

Telcone DTMF Receiver Family

For use in DTMF control applications, the Teltone Corporation offers a complete family of DTMF decoders. The family encompasses devices which can meet almost any conceivable application. Of special note are two specially priced kits -- the TRK-927 and TRK-947. Both kits contain the IC, a socket and a color-burst crystal.

The TRK-927 features both DTMF and rotary-dial decoding, selectable output formats (binary, 2 of 8, and 1 of 12), dial-tone immunity, and speech immunity. The price of the kit is $75.00.

The TRK-947 is similar to the 927, without some of the bells and whistles. The 947 decodes only DTMF, and the output is fixed at binary. The 947 is housed in a 22-pin package, while the more-complex 927 comes in a 40-pin hybrid package. The price of the TRK-947 is $53.00.

For additional information, write: Teltone Corporation, 10801 120th Ave. NE, Box 657, Kirkland, WA 98033.

Sprague Tantalum Chip Capacitors

Sprague offers a wide variety of chip tantalum capacitors for use where space is at a premium. A wide range of values and sizes are available. Ask for Engineering Bulletin 3539A for complete details and applications from: Sprague Electric Company, 87 Marshall St., North Adams, MA 01247.

QEX: The ARRL Experimenters' Exchange is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111 USA
telephone 203-666-1541

Victor C. Clark, W4KFC
President

David Sumner, K1ZZ
General Manager

Paul L. Rinaldo, W4RI
Editor

Associate Editors:
David W. Borden, K8MMO (Data Communications)
Mark Forbes, KC9C (Components)
Geoffrey H. Krauss, WA2GFP (VHF+ Technology)

The purposes of QEX are to:

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, and

3) support efforts to advance the state of the Amateur Radio art.

Subscriptions are available to ARRL members and non-members at the rates shown on the QEX Subscription Order Card inside this issue.

All correspondence concerning QEX should be addressed to the American Radio Relay League, 225 Main Street, Newington, CT USA 06111. Envelopes containing manuscripts and correspondence for publication in QEX should be marked: Editor, QEX. QEX subscription orders, changes of address, and reports of missing or damaged copies may be marked: QEX Circulation. Members are asked to include their membership control number or a label from their QST wrapper when applying.

Both theoretical and practical technical articles are welcomed. Manuscripts should be typed and double spaced. Please use the standard ARRL abbreviations found in the latest December issue of QST. Authors should supply their own artwork using black ink on white paper. When essential to the article, photographs may be included. 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 it will be when printed in QEX.

Any opinions expressed in QEX are those of the authors, not necessarily those of the editor or the League. While we attempt to ensure that all articles are technically valid, authors are expected to defend their own material. Products mentioned in QEX are included for your information, not advertising, nor is any endorsement implied. The information is believed to be correct, but readers are cautioned to verify availability of the product before sending money to the vendor. Material may be excerpted from QEX without prior permission provided that the original contributor is given credit, and QEX is identified as the source.

QEX: The ARRL
Experimenters' Exchange
1524 Springvale Avenue
McLean, VA USA 22101

Nonprofit Organization
U.S. Postage
PAID
McLean, Virginia 22101
Permit No. 235