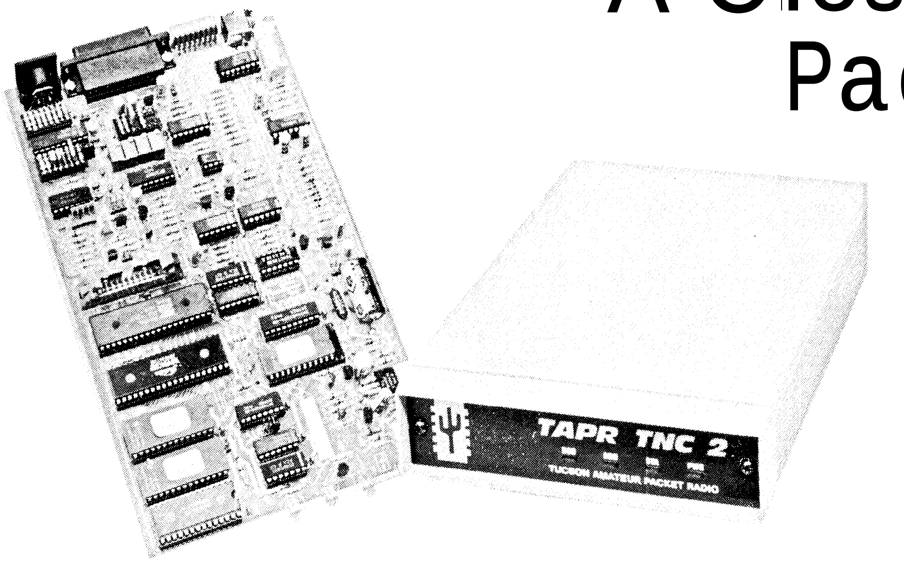
A Closer Look at Packet Radio



Last month we introduced you to packet radio; this month we'll look at what makes packet work-the terminal-node controller.

By Harold Price, NK6K
1211 Ford Ave., Redondo Beach, CA 90278

If you've read last month's article on packet radio, you have probably already joined the ranks of packet users and are ready to learn more about how packet works. If you haven't read last month's article, please read it now. We'll wait.

While we're waiting for them to get back, let me mention a few things that happened in the shack today, all of them fairly common occurrences on packet. You'll recall, of course, that packet is a mode of communications that allows information in digital form to be passed easily between amateur stations, that it is high speed, and that it guarantees perfect transmission of data. Packets are also easily manipulated by computer, allowing computer-assisted relaying of messages and data files.

I received a message that originated at a station in Newington, Connecticut. It had been relayed by computer on VHF to a station in Boston, where it was relayed by computer on 20 meters to a station in California. Then, I picked it up on VHF at my home in Redondo Beach in Southern California. I added some comments to the message and transmitted it on VHF through two real-time relay stations to a computer 200 miles north of me, where it will be picked up tomorrow by another station 200 miles farther to the north. I "talked" by keyboard to a station 400 miles north, in San Francisco, using four relay stations. I transmitted a copy of a packet-radio newsletter to a local ham at the speed of 120 characters per second. All of this was done using a VHF radio attached to a packet-radio controller called a *terminal-node controller* (TNC). One of the messages, by the way, discussed plans to begin transmitting data at 960 characters per second.

Is everybody back now? Good. This month, we're going to take a look at some of the technical parts of packet radio, specifically the TNC-that combination of hardware and software that does much of the hard work involved in supplying all of the services described previously.

Since it is sometimes useful to point to a concrete example of a concept under discussion, we'll use a TNC called TNC 2 (Fig. 1), designed by several hams who are part of an organization called TAPR, the Tucson Amateur Packet Radio Corporation. TAPR is a nonprofit research and development group that does work in packet radio in much the same way that AMRAD and AMSAT do work in their fields. The concepts that will be discussed here hold true for most TNCs, but are particularly applicable to the implementations by AEA, Heath and Kantronics, since their TNCs employ elements of the hardware and software previously developed by TAPR.

What A TNC Does

In professional circles, a TNC is called a packet assembler/disassembler (PAD). From this name, it is easy to figure out that a TNC's primary task is to convert data into packets, and packets into data. The TNC gets data from the user, forms it into packets and sends it out. The TNC also listens for packets, changes them back into data and passes the data to the user. There are several subsections to a TNC that allow it to do this. In the following discussion, refer to the block diagram in Fig. 2.

The User Connection

The TNC is usually attached to a local data device. This can be a terminal, a computer, a modem, a digital voice encoder or any other data-generating/using device. The TAPR TNC 2 communicates with this "user" through a serial communications port using standard RS-232-C voltage levels and signals. This means that if you have a terminal or computer that can be connected to an external modem, you can use the TNC 2. The flow of information on the user port is independent of the flow of information through the radio; the speed and data format used on the user connection don't have to match what's going on elsewhere in the TNC. Your terminal or computer doesn't even have to "know" that it is connected to a TNC. Most TNCs permit data rates between 110 bit/s and 19.2 kbit/s on the user port. While some TNCs change data rate with a software command, the TNC 2 uses switches.

The Radio Connection

The TNC must monitor the incoming signals and convert the tones it hears into ones and zeros that the rest of the TNC can deal with. It must also convert ones and zeros that the TNC wants to send into a form the radio can transmit. These jobs are performed by a demodulator and a modulator, respectively. The combination of modulator and demodulator is called a *modem.* The TNC 2 is equipped with an on-board *AFSK* modem that can be used to send data at various speeds, using various mark and space tones. On VHF packet radio, the 1200-bit/s standard is based on the Bell 202 modem standard. It uses mark and space tones of 1200 and 2200 Hz. For HF, 300 bit/s and the Bell 103 modem standard is used, using 1070- and 1270-Hz tones. As it turns out, which tone is used for mark and which for

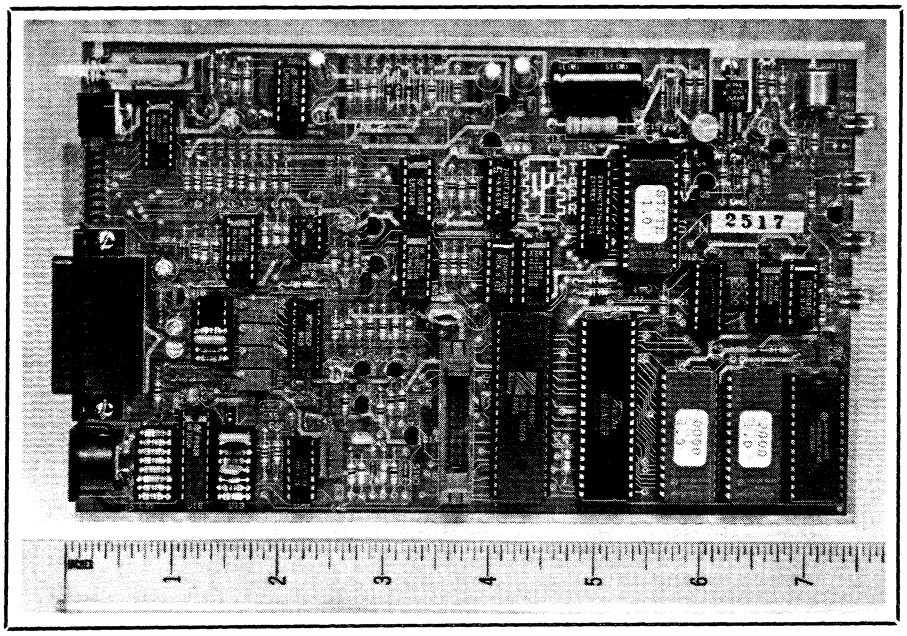


Fig. 1-The PC board for the TAPR TNC 2 shows the size and parts-count reductions made possible by advances in IC technology and experience gained through design of the TAPR TNC 1.

space is of little importance on packet, since the data is sent using the special Non-Return-To-Zero-Inverted (NRZI) encoding. With NRZI, a change from one tone to the other is used to signal a zero, and no change of tone signals a 1.[1]

Although the modem in the TNC 2 is limited to 1200 bit/s or slower, TNC 2 and several other TNCs provide a way to bypass the on-board modem and use an external modem. A *modem disconnect* jack is available for this purpose, and, with the correct external modem, the TNC 2 will support data rates up to 56 kbit/s.

Before they are sent to the demodulator, received signals are conditioned by a switched-capacitor input filter. This is done because the frequency response of most VHF FM radios is somewhat less than optimal for easy decoding of a 1000-Hz shift. An entire article could be devoted to the intricacies of the modem, and, in fact, one has.[2]

In addition to modulating and demodulating, the TNC must control the push-to-talk (PTT) circuit of the radio. Since TNCs are sometimes used as unattended automatic repeaters, the FCC and common sense dictate that there be some protection against a TNC failure causing a long key-down period. The TNC 2 provides a timer on the PTT line that will turn off the transmitter if it is on for more than 15 seconds. Fifteen seconds is longer than it will take to transmit the longest possible group of contiguous packets.

In the TNC 2, as Fig. 2 shows, both the radio I/O functions and the user I/O functions are performed by a single Zilog serial I/O (SIO) chip. The SIO provides two independent serial I/O channels in one IC. This reduces parts count, power consumption and price over previous TAPR designs, while retaining the high speed made possible by having these functions performed in hardware.

Data Processing

Packet radio is easy for the operator, but it is no simple task for the TNC. The TNC must listen to both the user's data port and to the radio. It must watch the stream of packets, looking for packets addressed to it. It must acknowledge packets received correctly, complain about those received out of order and ignore those received with errors. The TNC must send out its own packets, keeping track of those that have been acknowledged and those that haven't. It must time several events: how long to wait for an acknowledgment, how long to wait for the transmitter to turn on, and others. The TNC must know who it is talking to (so it can tell other TNCs it is busy), keep track of the number of times a packet has been retransmitted and be able to relay ***(digipeat)*** other user's packets when

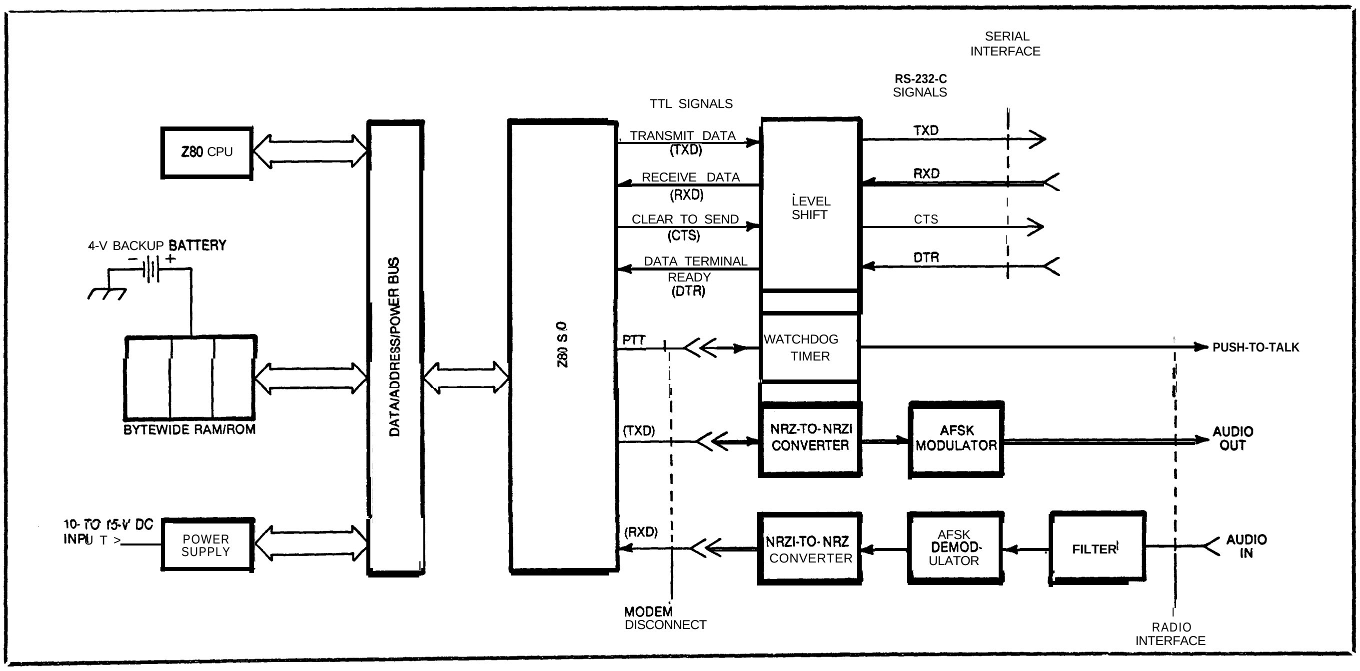


Fig. 2-A block diagram of the TAPR TNC 2. While TNC implementations vary, the services provided by the subsections in this TNC are provided in all other TNCs.

---

[1] Notes appear on page 20.

Glossary

acknowledgment (ACK)-a packet sent'& a station to confirm correct: reception of a packet or group of packets.

&dress--in amateur packet radio, a station's call sign followed by a number from 0 to 15.

AFSK—audio-frequency shift keying.

ASCII—the American National Standard Code for Information Interchange, a common binary code that specifies 7-bit representations of letters, digits, punctuation and special characters..

AX.25—the most common packet-radio protocol. AX.25 is specified in the ARRL publication *AX.25 Amateur Packet-Radio Link-Layer Protocol.* This protocol defines communication between two stations with only rudimentary relaying by intermediate stations. . .

bit-& single signal element, which can be either a binary 0 or 1. Groups of bits are used to convey information. Signaling speeds are usually stated in bits per second (bit/s).

bulletin board-a computer system used to store messages, files and bulletins.

connection—the imaginary "data pipeline"' that exists between twa stations that are communicating using a packet-radio protocol. During a connection, all data sent by one station will be delivered to the other station without errors.

data—bits that convey information.

digipeat—to retransmit a packet after it has been received, This is a specific function of the AX.25 protocol and is performed by *digipeaters.* With suitable radio equipment, a TNC can act as' a digipeater.

flow control—a means of controlling the rate at which data is transferred between two devices. Examples are the XON/XOFF character-based flow-control system used between the TNC and the host computer or user; hardware flow control, which uses the RS-232-C control lines; and the "sliding window" flow-control system used between TNCs.

frame check sequence (FCS)—a 16-bit number included in every packet to aid error detection. The number is the result of a calculation called a cyclic redundancy check (CRC).

gateway—A device that provides a path for data flowing between two networks,. In amateur packet radio, it may be used to connect stations on two different bands or frequencies- .

host system-a computer system that can be accessed via packet radio, As well as providing the services of a bulletin board, a host system can run programs for remote users.

mail box-see *bulletin board.*

modem—the device that converts logic voltages to audio tones fur transmission, and tones to voltage levels for reception, A contraction of modulator/demodulator. ,

monitor mode—a mode of TNC operation during which the TNC displays all packets that it receives, not just those addressed to it,

network—a group of stations that can relay data among themselves.

network node—a station that uses a special *network protocol* (as yet,, none of these exist' in Amateur Radio).

NRZI—Non-return-to-zero-inverted. A form of bit coding in which a 0 bit causes a change of state, or level,, and a 1 bit causes no change In state,

packet-a group of bits that contains, along with any data being communicated, the addressing, control and error-detection information necessary for error-free communication.

protocol—a set of rules agreed upon by two stations in order to communicate.

RS-232-C—the specification of the voltage levels and signals of a serial interface.

teleport—a station that provides a link between a terrestrial network and a satellite.

terminal-node controller (TNC)—a dedicated, microprocessor-based device that communicates with other TNCs using a packet protocol and with the user's terminal or computer using serial ASCII (RS-232-C).

user interface—the set of commands that a user can enter into the TNC and the messages or responses returned by the TNC to the user.

called upon to do so. It must listen to the radio and not transmit if another signal is on the air.

It takes a computer to keep track of all this. The last item in the preceding paragraph has proven especially difficult for human operators to do; just listen to 20 meters on any contest weekend. The processing power for the TNC 2 is provided by a Zilog Z80® CPU, running at 2.45 MHz. The TNC 2 has three memory sockets. These sockets usually hold 16 kbytes of read-only memory (EPROM) for program storage and 8 kbytes of read/write memory (RAM) for data storage. Using the largest available ICs, ıt is possible to get a total of 56 kbytes of memory into the three sockets, which leaves some room for expansion of the TNC software.

A lithium battery supplies backup power to the TNC 2 RAM when main power is removed from the TNC. This is exactly the same scheme that is used in most mobile or hand-held VHF/UHF radios; a small internal battery keeps memory active, so the rig can "remember" repeater frequencies and offsets. TNC 2 uses its battery-backed-up RAM to remember your call sign and the settings of the more than 70 variable parameters used to configure the TNC to your needs.

Software

Writing the software used to control a TNC is one of the more difficult programming tasks required in Amateur Radio. It is rivaled only by the software in OSCARs 10 and 11, and the software on some of the more complex hilltop repeater and remote-base systems.

A TNC requires two different types of software. First, the TNC is a computer speaking to other computers. It does this using an agreed-upon language called a *protocol.* A good protocol definition is very precise, leaving no room for interpretation or surprises.. The packet-radio protocol in widest use today is called *AX.25*$^{\mathrm{SM\ pend.}}$. The specification of AX.25, 40 pages of protocolese,, is perhaps the most comprehensive set of rules that any large segment of the Amateur Radio population has ever agreed to live by (apart from Part 97 itself).[3] AX.25, however, is only the first floor of a multiple-story house that is being built by the packet-radio community. The task of specifying (and agreeing upon!) protocols has really just begun. Although the AX.25 protocol is sometimes hard for humans to understand, it is just the type of well-defined task at which computers excel.

The second type of software in a TNC is the *user interface* program. Here things are a little less certain; humans have an amazing propensity for attempting things never before tested, tried, planned or even imagined. What humans lack in speed, they make up for with the talent for unerringly choosing the wrong thing to do at the wrong time. There are no reliable methods for guessing what people will do next:. Computers and computer programmers do not like this kind of behavior. Thus, the amount of software written to talk to the user usually exceeds the amount written to talk to the other TNCs.

Writing TNC code is not for the faint of heart, but it can be a rewarding experience. TNC 2, like most other TNCs, comes with all the necessary protocol and user software stored in EPROM. This means that if software updates are necessary, they can be accomplished by merely changing a memory IC or two.

Power Supply

TNC 2 requires an external 10- to 15-V dc supply. An on-board switching-mode power supply converts that input to regulated +5 V, and -5 V. The supply also provides -7 V for the RS-232-C outputs. (The two RS-232-C output levels

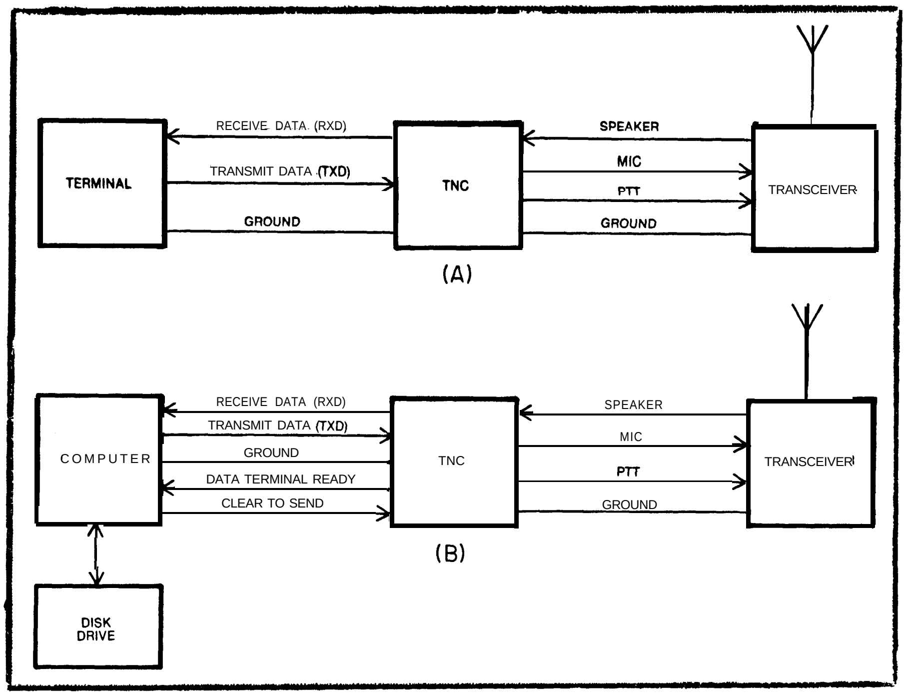


fig. 3-A shows the simplest TNC, radio and terminal connections. In B, the additional hardware flow control lines between the computer and the TNC make it possible to transmit large data or message files.

are -7 V and the positive input voltage.) Populated with NMOS ICs, the TNC 2 draws 260 mA at 12 V. For lower-power consumption, the TNC 2 was designed so that it could be built with CMOS parts. With CMOS ICs, the TNC draws less than **120 mA** at 12 V.

Status Indicators

You never quite feel you've gotten your money's worth unless you've got flashing lights. TNC 2 has four of them: POWER ON (the function should be obvious); CONNECTED, lights when a connection to another TNC has been established; DATA CARRIER DETECT glows when a mark or space tone is heard; and STATUS is lit when the TNC has sent a packet but has not yet received an acknowledgment for it.

Hooking It Up

The TAPR TNCs grew from a single overriding desire: to make packet radio as easy as possible to use! The Bell 202 modem standard and 1200-bit/s data rate are used because this is as fast a signal as can be easily forced into the microphone jack and taken from the speaker terminals on most VHF/UHF radios. To go faster requires a direct connection to the modulator and discriminator. This is not particularly difficult, but would limit packet operation to those with older, larger rigs or to surgeons and jewelers with the steadiness of hand, keenness of eye and tiny tools necessary to make modifications to shirt-pocket radios.

Since the TNC was designed for easy installation, it should not be surprising that hooking up a TNC to your station is very simple. As shown in Fig. 3A, you can get by with three wires to your computer or terminal and four to your radio. The hard part will be finding a proper mic connector for your radio! Just remember, when connecting a TNC to your radio, think of the TNC as a microphone and a speaker; when connecting a TNC to a computer, think of the TNC as a modem.

If you use your TNC only for RTTY-like typing contacts, you need to connect only three wires between the TNC and your computer: one for data from the computer to the TNC (TXD), one for data from the TNC to the computer (RXD) and one ground (Fig. 3A). If you want to use your computer to send large files or messages to your TNC, then you must provide a way for the TNC to control the stream of data coming from your computer. This is called *flow* control. Flow control is required because your computer can send data to the TNC faster than the TNC can send data to the receiving station. If you send a stream of data at 1200 bit/s to your TNC, retries-caused by collisions, dropped packets or other mishaps-will cause the limited RAM in your TNC to fill up with characters waiting to be transmitted. Your computer must be prepared to wait when the TNC memory gets full. Two flow-control methods are available on the TNC 2: hardware flow control, using the CTS and DTR lines on the serial port (Fig. 3B), or software flow control, using the ASCII XON and XOFF characters.

As mentioned earlier, on TNCs that have a modem disconnect, you are not limited to use of the on-board modem. The TNC 2 is capable of running at 56 kbit/s with an appropriate external modem. A 9600-bit/s modem that connects to the modem disconnect has been designed, and other, even faster, modems are under discussion.[4] High speeds will be used primarily for communication between gateways and network nodes, but that's a topic for another day.

Wrapping It Up

We've seen what a TNC is and what it does. We've used the TAPR TNC 2 as a specific example. I'd like to mention the chief architects of that project. Paul Newland, AD7I, did the hardware design, and Howard Goldstein, N2WX, did the software. Steve Goode, K9NG, provided input on the modem, and other design input and review came from Pete Eaton, WB9FLW, and Lyle Johnson, WA7GXD.

What happens next? I'd like to suggest that you stop reading about packet radio and do something about it. Packet radio is a young enough part of our hobby that you can get in on the ground floor and have a very real effect on the future growth and direction of computer networking in the Amateur Radio Service. You might just also affect the future of Amateur Radio itself.

If you'd like to see more *QST* articles about packet, including things on high-speed modems, gateways, n-port digipeaters and network access ports, send a letter to the editors. If they hear that there is interest in packet radio, they are more likely to publish packet-related articles. To stay current in the meantime, you can join any of the several packet radio clubs that print newsletters. Also, the biweekly ARRL packet radio newsletter, Gateway, provides short reviews and summaries of packet development activity.

See you on packet!

Notes

[1] ***The 1985 ARRL Handbook for the Radio Amateur,*** p. 19-25.

[2] M. Morrison and D. Morrison, "Designing the TAPR TNC Audio Input Filter," ***Proceedings of the 2nd ARRL Amateur Radio Computer Networking Conference.*** Available from ARRL for $9.

[3] *AX.25* ***Amateur Packet-Radio Link-Layer Protocol, Version 2.0, Oct.*** 7984. Available from ARRL for $8.

[4] S. Goode, "Modifying the Hamtronics FM-5 for 9600-Baud Operation," ***Proceedings of the 4th ARRL Amateur Radio Computer Networking Conference,*** available from ARRL for $10.

I would like to get in touch with. . .

☐ anyone using an EAGLE IIE computer to work RTTY, AMTOR and packet. Jack Clark, W9HJM, 93 Downing Dr., Chatham, IL 62629.

☐ anyone with a four-section electrolytic can capacitor rated at 20 µF/ 20 µF/ 20 µF/ 30 µF at 650 V. Charles Schramm, Jr., KA2JLC, 28-28 35 St., Long Island City, NY 11103.