W 6 SCR
CHUCK COOK

OPERATION

HEATHKIT® MANUAL

for the

TERMINAL NODE CONTROLLER

Model HD-4040

USER'S COMM PAGE 3-3 595-3428

HEATH COMPANY • BENTON HARBOR, MICHIGAN

HEATH COMPANY PHONE DIRECTORY

The following telephone numbers are direct lines to the departments listed:

Kit orders and delivery information .... (616) 982-3411
Credit .... (616) 982-3561
Replacement Parts .... (616) 982-3571

Technical Assistance Phone Numbers
8:00 A.M. to 12 P.M. and 1:00 P.M. to 4:30 P.M., EST, Weekdays Only

Audio .... (616) 982-3310
Amateur Radio .... (616) 982-3296
Test Equipment, Weather Instruments and
Home Clocks .... (616) 982-3315
Television .... (616) 982-3307
Aircraft, Marine, Security, Scanners, Automotive,
Appliances and General Products .... (616) 982-3496
Computers — Hardware .... (616) 982-3309
Computers — Software:
Operating Systems, Languages, Utilities .... (616) 982-3860
Application Programs .... (616) 982-3884

YOUR HEATHKIT 90-DAY LIMITED WARRANTY

Consumer Protection Plan for Heathkit Consumer Products

Welcome to the Heath family. We believe you will enjoy assembling your kit and will be pleased with its performance. Please read this Consumer Protection Plan carefully. It is a "LIMITED WARRANTY" as defined in the U.S. Consumer Product Warranty and Federal Trade Commission Improvement Act. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Heath's Responsibility

PARTS — Replacements for factory defective parts will be supplied free for 90 days from date of purchase. Replacement parts are warranted for the remaining portion of the original warranty period. You can obtain warranty parts direct from Heath Company by writing or telephoning us at (616) 982-3571. And we will pay shipping charges to get those parts to you . . . anywhere in the world.

SERVICE LABOR — For a period of 90 days from the date of purchase, any malfunction caused by defective parts or error in design will be corrected at no charge to you. You must deliver the unit at your expense to the Heath factory, any Heathkit Electronic Center (units of Veritechnology Electronics Corporation), or any of our authorized overseas distributors.

TECHNICAL CONSULTATION — You will receive free consultation on any problem you might encounter in the assembly or use of your Heathkit product. Just drop us a line or give us a call. Sorry, we cannot accept collect calls.

NOT COVERED — The correction of assembly errors, adjustments, calibration, and damage due to misuse, abuse, or negligence are not covered by the warranty. Use of corrosive solder and/or the unauthorized modification of the product or of any furnished component will void this warranty in its entirety. This warranty does not include reimbursement for inconvenience, loss of use, customer assembly, set-up time, or unauthorized service.

This warranty covers only Heath products and is not extended to other equipment or components that a customer uses in conjunction with our products.

SUCH REPAIR AND REPLACEMENT SHALL BE THE SOLE REMEDY OF THE CUSTOMER AND THERE SHALL BE NO LIABILITY ON THE PART OF HEATH FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING BUT NOT LIMITED TO ANY LOSS OF BUSINESS OR PROFITS, WHETHER OR NOT FORSEEABLE.

Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Owner's Responsibility

EFFECTIVE WARRANTY DATE — Warranty begins on the date of first consumer purchase. You must supply a copy of your proof of purchase when you request warranty service or parts.

ASSEMBLY — Before seeking warranty service, you should complete the assembly by carefully following the manual instructions. Heathkit service agencies cannot complete assembly and adjustments that are customer's responsibility.

ACCESSORY EQUIPMENT — Performance malfunctions involving other non-Heath accessory equipment, (antennas, audio components, computer peripherals and software, etc.) are not covered by this warranty and are the owner's responsibility.

SHIPPING UNITS — Follow the packing instructions published in the assembly manuals. Damage due to inadequate packing cannot be repaired under warranty.

If you are not satisfied with our service (warranty or otherwise) or our products, write directly to our Director of Customer Service, Heath Company, Benton Harbor MI 49022. He will make certain your problems receive immediate, personal attention.

Heathkit® Manual

for the

TERMINAL NODE CONTROLLER

Model HD-4040

USER'S COMM PAGE 3-3 595-3428

HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

WARNING

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of the FCC Rules. Only computers certified to comply with the Class B limits may be attached to this equipment. Operation with non-certified computers is likely to result in interference to radio and TV reception.

This equipment uses radio frequency energy for its operation and if not installed and used properly, that is, in strict accordance with the instruction manual, may cause interference to radio and television reception. It has been type tested and found to comply with the RF emission limits for a Class B computing device which is intended to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio and television reception, which you can determine by turning the equipment off and on, try to correct the interference by one or more of the following measures:

- Move the computing device away from the receiver being interfered with.
- Relocate the computing device with respect to the receiver.
- Reorient the receiving antenna.
- Plug the computing device into a different AC outlet so that the computing device and receiver are on different branch circuits.
- Disconnect and remove any I/O cables that are not being used. (Unterminated I/O cables are a potential source of high RF emission levels.)
- Unplug and remove any serial I/O circuit board cards that are not being used. (Here again, unterminated cards can be a source of potential interference.)
- Be certain that the computing device is plugged into grounded outlet receptacles. (Avoid using AC cheater plugs. Lifting of the power cord ground may increase RF emission levels and may also present a lethal shock to the user.)

If you need additional help, consult your dealer or ask for assistance from the manufacturer. Customer service information is on the inside back cover of this Manual or on an insert sheet supplied with this equipment. You may also find the following booklet helpful: "How to Identify and Resolve Radio-TV Interference Problems." This booklet is available from the US Government Printing Office, Washington, D.C. 20402, Stock No. 004-000-00345-4.

Heathkit®

TABLE OF CONTENTS*

* Portions of this Manual were copied directly from TAPR (Tucson Amateur Packet Radio) documentation.

PACKET RADIO THEORY

> We strongly recommend that you read the following paragraphs so you will become familar with packet radio theory before you begin to operate your Terminal Node Controller.

A number of radio amateurs in the USA and various other countries have, for some time, been experimenting with packet radio, a system of computer-based communications. This new mode combines reliable, high-speed communication with efficient use of the radio spectrum. Its transmission is resistant to interference by signals from other stations and to signal degradation due to adverse band conditions. Besides using packet radio for informal amateur QSOs and traffic handling, it has additional functions such as the exchange of data between hams with computers, "bulletin boards" and message systems, and remote computer programming.

WHAT IS PACKET RADIO?

Packet radio is a communication system that digitally encodes information (text or data). In this respect, it is similar to radio teletype, but with important differences. These differences are the key to insuring error-free reception and, at the same time, allowing maximum use of the radio spectrum through shared frequency use. Unlike regular radio teletype, which transmits data one character at a time, each transmission from one packet radio station to another consists of a bundle (packet) of data. In addition to transmitting the usual data between stations, the packet includes source and destination addresses, packet control information, and error check information.

Packet radio provides data integrity through handshaking and error detection techniques. Each transmission includes a computed value called FCS (frame check sequence). The FCS allows the receiving station to check the received message for errors. The receiving station acknowledges an error-free packet with a special acknowledgment (ACK) message. If the sending station does not receive such a message within a certain time period, it automatically retransmits the packet. This process is repeated either until a valid packet is acknowledged or a preset number of attempts have been made.

A packet also identifies the communicating stations. This permits several QSOs to take place on the same frequency. A packet radio station can automatically ignore any packets which are not addressed to it. Because most packet transmissions are very short, you do not need the "channel" most of the time. The time between each transmission is available to other operators on the frequency. This system is called time-domain multiplexing. On a very busy channel, you will notice an increased delay time before getting replies to transmissions. However, the packet radio equipment takes care of automatic retransmissions and sorts out the replies meant for your station. Note that you never "hear" the interference on the channel.

WHAT IS A PACKET RADIO STATION?

Packet radio requires the use of a microprocessor-based controller at both the transmit and receive stations. It will obviously appeal to you if you have a computer in your shack. However, it does not require that you be a programmer, or even that you have a personal computer. All that is really necessary is a terminal, a Terminal Node Controller (TNC), and an amateur radio transceiver.

The terminal can be a simple display or typewriter terminal that recognizes ASCII characters, a personal computer, or even a commercial mainframe computer. What you need is a terminal with a keyboard to allow you to "talk" and a screen or printer to allow you to read incoming information. You can even use an inexpensive terminal with a TV set display.

Most terminals encode ASCII characters in an "asynchronous" format. This means that there is no predictable time interval between individual characters, because they are encoded as they are typed. A flag, consisting of one or more "mark" (binary 1) values, is used to mark the beginning and end of each character. The device decoding the characters expects a specific "baud rate" (number of transitions from "mark" to "space" per second) during the character, but no particular time interval between the characters themselves. The TNC communicates with your terminal or computer using ASCII characters in this format.

The TNC is the heart of the packet radio system. It has one port that is connected to the terminal or computer. It communicates through this port in an asynchronous ASCII format at the baud rate established by the terminal. The TNC converts the ASCII data stream from the terminal into a packet by attaching a header showing the packet destination and control information for the network, a tail containing the result of the FCS calculation for error detection, and flags to mark the beginning and end of the packet.

The second TNC port connects it to the transceiver microphone and speaker audio lines, and the PTT (Push-To-Talk) line. Ordinarily, the TNC produces AFSK modulation by feeding one of two tones into the microphone input. These tones correspond to a mark or space. In this manner, the complete packet is transmitted at a packet channel baud rate. Note that this rate is unrelated to the terminal baud rate at the other TNC port.

The receiving TNC reverses this procedure, decoding the audio tones from the speaker audio line of the radio, removing and reading the header and tail information, and passing a successfully received packet to the terminal at the terminal baud rate.

The part of the TNC that translates the sequence of tones into digital characters is called a "modem," an acronym for **mod**ulator/**dem**odulator. This device is built into your TNC. However, the TNC has a connector that allows you to use an external modem with different characteristics. Most packet radio modems operate at 1200 baud, which corresponds to about 1200 wpm, although the FCC now authorizes much higher baud rates on some amateur bands. The audio tones used are 1200 Hz and 2200 Hz, which are the frequencies chosen for the Bell 202 modem.

The final component of a packet radio station is an amateur radio transceiver. Most packet radio activity so far has been in the 2-meter band. The only important requirement of the radio is that its audio frequency response at 2200 Hz be adequate. In other words, the 2-meter FM rig you already have is probably just fine.

WHAT DOES THE TNC DO?

The TNC consists of a special-purpose microcomputer. This unit contains all the necessary software and hardware needed to communicate with your terminal, assemble a packet, operate your transmitter and receiver to send and receive a packet, and decode a packet. The special TNC functions would be difficult to implement with an ordinary personal computer because of the use of protocol to communicate with other TNCs and the real-time control requirements.

The encoding and decoding of packets involves a carefully standardized set of procedures called a protocol. The protocol determines the exact form and content of the packet header and tail. The header allows receiving TNCs to automatically determine the purpose of the packet; that is, net check-in, part of a QSO, or acknowledgment of a previous transmission. The tail contains the FCS, which allows the TNC to automatically determine whether the packet was received correctly and, if so, to automatically acknowledge it. Since the protocol is programmed into the TNC, you do not need to know how the destination of your packet is indicated. You communicate with other amateurs by using your call sign, and the TNC translates the call sign into the identification required by the protocol.

Programming of the TNC must be as flexible as possible, since there will inevitably be unforeseen enhancements to and problems with the initial software. For this reason, the Heath TNC is designed to use Erasable Programmable Read-Only Memories (EPROMs). These devices normally function like the ROM in a personal computer, where the vital software is stored in an indestructible form. However, you can reprogram the TNC if the need arises.

WHAT IS A PACKET?

A packet is the basic message unit in packet radio. Ordinarily, it consists of a text message that you type in, sandwiched between the header and tail information required by the protocol. In a typical QSO, a packet is encoded and sent by the TNC when you end a line of typing by pressing the RETURN (or CR or ENTER) key. The length of a packet is limited, usually to 128 characters. This helps to prevent a single operator from "hogging" the channel, as well as making sure that the sending and receiving TNCs are not swamped with information.

A packet need not be made up of ASCII or Baudot character strings. It may contain information in other coding systems, such as BCD or EBCDIC, or even binary data such as a compiled computer program. The TNC, which uses a bit-oriented protocol based on a standard called High-Level Data Link Control (HDLC), can encode any of these equally well. An advantage to this choice of protocol is that the functions it requires are available on a single, large-scale integration (LSI) chip, which simplifies the TNC hardware and software. A second advantage of HDLC protocol is that the beginning and end of the entire message are flagged, making the start and stop bits for each character unnecessary as the packet is transmitted in "synchronous" format.

The HDLC frame is represented below. Each packet field is encoded as a sequence of ones and zeros (bits) which are transmitted as mark and space tones. With the exception of the DATA field, all these fields are generated by the TNC as it assembles the packet for transmission. As an operator, you are concerned only with the contents of the DATA field.

The following paragraphs describe each packet field in detail.

FLAG	ADDRESS	CONTROL	DATA	FCS	FLAG

The FLAG is a unique bit sequence which identifies the beginning of a packet to the HDLC controller. This pattern does not correspond to any sequence which would be encountered in any of the other fields, except possibly in the transmission of binary data. Even in this case, there are provisions for distinguishing data from the flag sequence.

The ADDRESS field contains packet routing information. This information may include the destination station, the originating station, and possibly intermediate routing information if the packet is relayed to the destination. The destination and originating stations may be identified by a network address number or by an amateur call sign, depending on the exact form of the protocol being used.

The CONTROL field describes the purpose of the packet to the recipient. It identifies packets with such functions as initialization or termination of communications, packet acknowledgment, or request for retransmission. It may also contain a sequence number for a multipacket message which must be received in the correct order.

The DATA field contains the message being sent, which is ordinarily the text you typed converted into an ASCII data string. In the case of a packet identified in the control field as performing a control function, the DATA field may be absent.

The FCS allows the receiving station to verify that the packet has been received correctly. The final FLAG marks the end of the packet and, if the FCS calculated by the receiving TNC matches the FCS of the packet, an acknowledgment is sent; otherwise, the packet is ignored.

WHAT IS A PACKET NETWORK?

A local area packet radio network (LAN) consists of a number of individual packet radio stations, which are ordinarily within simplex range. The net may also contain a digital repeater or "digipeater," which may also function as an individual station. The digipeater is a single-frequency relay station which retransmits any correctly received packets.

The protocols currently implemented by amateur packet radio require stations to communicate pairwise through connections established through the exchange of special packets. An operator who wishes to start communicating with another net station subsequently has his transmissions addressed to that station. In order to simulate a conventional amateur net, stations can monitor transmissions from stations other than the ones to which they are connected. Of course, the TNC only acknowledges those transmissions intended for that station.

WHAT IS IN THE FUTURE FOR PACKET RADIO?

As more packet radio LANs become active, we may soon have link stations with access to two different areas. These stations can serve as communications links through which packets originating in one area can be funneled to an addressee in another area.

A more sophisticated idea is the possibility of having a "gateway" station, which will be a specialized station having access to some long-distance mode of communications. The gateway station will reformat packets with another layer of protocol containing internetwork linking information and transmit it to another gateway station in a distant LAN. Three possibilities are being explored for long-distance links.

Ground Relays

TERRACON will be a high-speed, ground-based linking system utilizing UHF and/or microwave relays. It could potentially handle most long-distance packet radio communications in the USA and Canada. It will probably be a few years before TERRACON is implemented as a useful system, and somewhat longer before the North American continent is linked.

Satellite Service

AMICON is a satellite-based network that will utilize one of the special-services channels on the AMSAT Phase III B satellite. It will use a reserved 5 kHz segment of the mode B transponder (with 435 MHz uplink and 145 MHz downlink). AMICON will allow intercontinental linking and contact with isolated areas not accessible to TERRACON. High-data rate experiments are being planned for the mode L translator (with 1296 MHz uplink and 435 MHz downlink).

PACSAT is a new class of satellite designed solely for digital communications. Current designs call for multiple packet radio uplinks on 435 MHz into a low earth-orbiting (LEO) UoSAT-class OSCAR satellite containing a packet radio repeater (digipeater). A common downlink on 145 MHz would provide either real-time repeating service, or a delayed message storage facility, using up to one megabyte of on-board storage. This "flying packet radio mailbox" could also have more traditional digital channels, like RTTY and ASCII.

There are also possible plans for a packet radio digital repeater aboard the AMSAT Phase III C satellite.

Short-Wave Links

SKIPCON is Amateur Radio Research and Development Corporation's (AMRAD's) projected HF network of LAN gateway stations. The nature of HF propagation will require slower data rates (75 to 600 baud) and error correction as well as error detection protocol. A form of error-correcting code for amateur radio known as AMTOR may be the best candidate for handling the unpredictable characteristics of the HF channels. SKIPCON experiments have been conducted since the end of 1981.

OPERATION

Pictorial 1 shows the front panel of your Terminal Node Controller and briefly describes the function of each LED indicator and the rocker switch. Pictorial 2 identifies the rear panel connections and describes their functions. The following pages describe the operation of the unit, and the typical use of the TNC to receive and transmit "packets."

Before attempting to use the TNC, be sure you are thoroughly familiar with your communications equipment and its operation. This equipment should either be crystal-controlled or synthesized to ensure excellent frequency stability, which is important for packet radio operation. For a transmitter, be sure it has the ability to handle "key down" operation (continuous transmission of a CW carrier). Also, be sure you are familiar with your computer terminal or computer (used in the terminal mode) and its operation.

NOTES:

1. After the TNC types a sign-on message on your terminal, you are ready to operate.

2. You can use your computer to emulate a terminal by running a terminal emulator program. NOTE: This program will make the RS-232C port on your computer appear as a terminal input to the TNC.

GENERAL

You may find HF packet radio operation on the same subbands used for RTTY (radio teletype). These transmissions are generally within the frequency ranges shown below:

80 meter band:	3600 — 3680 kHz
40 meter band:	7075 — 7100 kHz
20 meter band:	14075 — 14110 kHz
15 meter band:	21075 — 21100 kHz
10 meter band:	28075 — 28100 kHz

Federal Communications Commission (FCC) regulations presently limit the baud rate for RTTY transmissions as follows:

FREQUENCY RANGE	MAXIMUM BAUD RATE
Below 28 MHz	300
28 to 50 MHz	1200
50 to 220 MHz	19600
Above 220 MHz	56000

A TNC running the Heathkit AX.25 software (built in ROM) has the following operating modes:

Command Mode — In this mode, everything you type is interpreted as instructions for the TNC. Instructions to the TNC are in the form of command lines terminated by a RETURN. These commands allow you to change the TNC operating parameters, perform special functions, or change modes. If your TNC receives packets while it is in the command mode, you will be able to see them (on the screen). To send packets, you must instruct the TNC to enter a data mode.

Data Modes — Two data modes are available; the Converse Mode and the Transparent Mode. In these modes, the information you type to the TNC is assembled into packets and transmitted on the radio.

Debug Mode — In addition to the above mentioned modes, the special Debug Mode is provided to allow you to examine and change TNC memory locations. You may use this mode if you are a software developer or if you wish to track down obscure problems. NOTE: A list of debug mode commands is given in the "Special Functions" section for those who may be interested.

The remainder of this section first describes the terminal you will use. It then explains how to use the commands to configure the TNC to suit you and your station, and how to get started talking to other hams on packet radio. This is not intended to be an exhaustive description of every command, but rather a discussion of how the various commands are related and how you may use them. An alphabetical catalog of commands that describes the format and parameters of each is given in the "Commands and Messages" section of this Manual.

TERMINAL CHARACTERISTICS

Baud Rate Selection — Most terminals use serial communications. In this mode, each bit of a 7- or 8-bit character is sent in sequence over the same wire. Serial data must be transmitted at a predetermined bit-per-second rate called baud rate. There are many standard baud rates. Unless the TNC and the terminal use the same baud rate, they will not be able to communicate. The TNC supports all standard baud rates from 50 to 19,200. You can select the desired baud rate from the table on Page 3-3 of this Manual.

Word Length/Parity/Stop Bits — In addition to the data rate, there are three other characteristics of serial data which should be the same for the TNC and the terminal. These are word length, parity, and number of stop bits. Use the commands AWLEN, ABIT, and PARITY to set these characteristics. Serial data may represent ASCII data, which has seven data bits per character, or binary data, which has eight data bits per byte. Unless you are operating in the transparent mode, the TNC will ignore the extra bit if you use 8-bit characters. This means that the eighth bit will be set to 0 before data is assembled into a packet.

If you change any of these configurations, the new values will not take effect until you perform a reset of the TNC (using switch SW-3☆, switch the power off, or by typing RESET).

If you start the TNC in the default-parameter mode with switch SW-1 on, the serial port will be initialized at 300 baud, space parity, 7-bit characters, and one stop bit. A message will then be typed at 300 baud. If you enter an * (asterisk) before the message is completed, the TNC will sign on at 300 baud. If you did not enter an * before the message was complete, type the * twice. The message will reappear — before it is completed, type another *. The TNC will then sign on at 300 baud.

☆ Refer to "DIP Switches" on Page 9-6 in the Assembly Manual for a detailed description of the function of each of the four sections of circuit board switch SW.

GETTING STARTED

After your TNC and terminal are set up properly, you should see the TNC sign on with a message, followed by a prompt:

```
cmd:
```

When you see this prompt, first set your call sign. To do this, type MY and your call sign (as shown in the example below):

```
cmd: MY W8XYZ <RETURN>
```

Then save this information by typing:

```
cmd: PERM <RETURN>
```

This writes your call and the terminal baud rate into the NOVRAM so they will be set automatically the next time you power up your TNC. Set switch SW-1 to the off position so the TNC will use the NOVRAM information instead of the default values. NOTE: Do a "soft reset" (by pressing the R and RESET keys) (TNC off) after resetting SW-1.

Once you have set your call sign, you are ready to try sending a packet. First, type the command CONVERS and press <RETURN> to enter the converse mode. Then type a message to be transmitted as a packet:

```
HELLO WORLD!
```

End your message by pressing the RETURN key. If you watch the LEDs on the TNC front panel as you do this, you should see the following activity take place:

1. The PTT light will come on, indicating that the transmitter is being keyed.
2. The TXD light will flicker on and off.
3. If this is the first packet you have transmitted, and if the CWID feature is on, you will see the CWID light blink on and off as your call sign is transmitted in FSK.
4. When the transmission stops, the PTT light will be off and, if a Morse code message was sent, the CWID light will be off.

NOTE: Do not be concerned if the TXD light is on at the end of the transmission. The encoding scheme used by the TNC represents data as transitions of the transmit-data line, so the light will be on at the end of a transmission about half the time. Also, the CWID light may be randomly on or off.

The message you just transmitted was sent to the address specified by the UNPROTO command. This address will be set to CQ when the TNC is turned on or reset.

If you want to know which stations use packet radio in your area, you must turn your monitor functions on so you can "read the mail." To do this, first return to the command mode by typing CTRL-C (the character produced when you type C and, at the same time, press the "control" key); then type each of the following commands:

```
cmd: M ON <RETURN>

cmd: MA ON <RETURN>

cmd: MC ON <RETURN>

cmd: MT ALL <RETURN>

cmd: MF ALL <RETURN>
```

NOTE: See "Commands and Messages" on Page 3-1 for a full explanation of each of these commands.

Any packet received by your TNC will now appear on your screen.

In order to make full use of the TNC's capabilities for reliable data communications, you should establish a connection with another station. This means that everything you type while in the converse mode will be automatically addressed to that station, and packets sent between your station and the other station will be automatically acknowledged by the recipient. The sending station will continue retransmitting a message until it has been received correctly. To connect to WB9XYZ, for example, return to the command mode by typing CTRL-C and type:

```
cmd: C WB9XYZ <RETURN>
```

If WB9XYZ is on the air, tuned to your frequency, and within range of your transmission, you should see a message coming back to your TNC. If you have a speaker as well as your TNC hooked up to your radio, you will hear the packets; otherwise, you can see the DCD LED light up, and you may see the two REC. AUDIO lights flicker. When your connect request (the packet your TNC sent) has been acknowledged, the TNC will display the following message:

```
*** CONNECTED TO WB9XYZ
```

The TNC will then move to the converse mode. If you now type a message, it will be formed into a packet and stored in memory until you press the RETURN key. At that moment, it is sent to WB9XYZ.

After you complete the conversation, either you or the operator of the other station may initiate a disconnect. To do this, return to the command mode (by typing CTRL-C) and type the following command:

```
cmd: D <RETURN>
```

After an exchange of packets, you will see the message:

```
* * * DISCONNECTED
```

This message indicates that your disconnect request packet was acknowledged by the station you were connected to. If you have enabled CWID, the TNC will send a final Morse code identification as you disconnect. NOTE: You must send a disconnect request to the station you are communicating with before you can connect to other stations or hear them, if your monitor functions are off.

If you are ready to receive messages from other packet radio stations and you wish to let them know that you are active on this mode, you may use the TNC as an automatic repeater (or beacon) to transmit this fact. First, you must decide how often you wish your station to send the desired message, and then store it in RAM in your TNC. For example, if you wish to send it every 15 minutes (or every 900 seconds), type the following after you see the command prompt:

```
cmd: BE 90 <RETURN>
```

NOTE: You must divide the number of seconds by 10 before you enter the desired time.

The TNC may respond with the following message:

```
was  0
```

if no number was previously entered in RAM, or the number that corresponds to the one you entered earlier.

Then type BT, followed by the message you wish to be sent:

```
cmd: BT W8XYZ PACKET RADIO STATION, BENTON HARBOR,
     MICHIGAN <RETURN>
```

The TNC will respond with the following message:

```
was  AX.25 level 2 protocol software version 3.2
```

If WB9XYZ, for example, "hears" your beacon and wishes to contact you, he will type: C W8XYZ <RETURN>

When your TNC acknowledges his connect request, it will display the following message:

```
*** CONNECTED TO WB9XYZ
```

Your TNC will then move to the converse mode. If you now type a message, it will be formed into a packet and sent to WB9XYZ.

As you now have succeeded in getting your packet radio station on the air, read the following pages which describe the TNC operation in more detail. The remainder of this chapter will help you get the most out of your TNC. Also refer to the alphabetical list of commands in the "Commands and Messages" section of this Manual.

OPERATING MODES

COMMAND MODE

Use the command mode to enter commands that alter the TNC's operating parameters. You must enter all other modes, except the debug mode, from the command mode. When the TNC is in the command mode, "cmd:" is printed as a prompt at the beginning of each input line. This is the TNC's signal that it is waiting for instructions.

The TNC is always in the command mode after a reset or power-up. You can perform a reset by disconnecting power from the TNC for several seconds by toggling reset switch SW-3 on the circuit board, or by issuing the RESET command. After a reset operation, all operating parameters of the TNC are reinitialized by the resident software.

You can store the value of most parameters in a permanent but easily changed form in the NOVRAM memory. Since this space is limited, some of the more complex parameters are not saved and must be initialized from the default values every time. If you change some of the parameters and want to use the new values upon reset, you can store them in NOVRAM by using the command PERM. In order to use these values, make sure switch SW-1 is off when the TNC is reset.

There are several ways you can get from the command mode to one of the data modes. You can type the command CONVERS or TRANS, depending on the desired data mode. This will cause an immediate mode change. If you issue a CONNECT command to initiate a conversation with another station, or if your TNC receives a connect request packet, the TNC will automatically change to a data mode after the connection is established. The data mode used is specified by the CONMODE command as CONVERS or TRANS. However, if you specify the data mode in the CONNECT command, that mode will be used without altering the setting of CONMODE.

DATA MODES

Converse Mode

The data mode you will probably use most often for ordinary contacts is the converse mode. In this mode, the information you type is assembled by the TNC into packets and transmitted over the radio. A packet is terminated whenever you type the send-packet character, which is set by the command SENDPAC and may optionally be included in the packet. In order to allow you to correct typing errors in your messages and to return to the command mode, there are nine characters which have special meanings to the TNC. These characters are not ordinarily included in the packet. These include input editing characters, which are discussed in a later section. NOTE: To get back to the command mode from the converse mode, type CTRL-C.

Transparent Mode

Packet radio is very well suited for the transfer of large amounts of data between computers. For some types of data transfer operations, the converse mode will work very well. However, you may want to send special information such as ready-to-run programs to another amateur radio operator. A .COM file on a CP/M system or even a BASIC program may contain many strange characters which could be confused with the special reserved characters in the converse mode. For this application, you will want to use the transparent mode — a data mode like the converse mode except that there are no special characters — everything you type, or everything your computer sends to the TNC, is sent over the radio exactly as it appeared to the TNC. Packets are sent at regular time intervals or when a full packet of information is ready. You may use the PACTIME command to change the time intervals at which data is put into packet form.

The display characteristics of the TNC are also modified in the transparent mode. Data is sent from the TNC to the terminal exactly as it is received over the radio channel, including all eight bits of each byte received. All features such as LINE-FEED and RETURN insertion, ESCAPE translation, and case conversion are disabled. In addition, echoing of input characters is disabled. None of the parameters which control these features in the converse mode are changed when you enter the transparent mode, and all display features are re-enabled when you return the TNC to the command mode. Most of the informative messages that appear in the converse mode as the TNC moves between the disconnected and connected states are also disabled.

If you wish to escape from the transparent mode to the command mode, you must follow a special procedure. After a time equal to PACTIME has elapsed, the last data you typed will have been put in packet form for transmission (although it may not be transmitted yet). You must then wait an additional time, which is set by the CMDTIME command. Following this wait, you must type CTRL-C three times within an interval CMDTIME of each other. After a final CMDTIME interval in which you did not type any characters, you will see the "cmd:" prompt. If you type any character in this interval, even if they are more COMMAND characters, the escape will be aborted and the three COMMAND characters will be sent as packet data. If you set CMDTIME or PACTIME to 0, you will not be able to escape from the transparent mode except by performing a hard reset (switch or power-down reset).

FLOW CONTROL

Whenever you transfer data to computers, there is a chance that the data will be received faster than the computer can handle it. In order to prevent loss of data, the computer must be able to make whatever is sending data stop sending, and later tell it to resume sending. If you are a home computer user, you are probably already familiar with one type of flow control, which allows you to stop the output from the computer while you read it and restart it when you have finished.

There are two methods of providing flow control that are supported by the TNC. XON/XOFF flow control, sometimes called "software flow control," is accomplished by sending a special character (usually a CTRL-S) to request that the output stop and another special character (usually a CTRL-Q) to restart the output. Hardware flow control may be used if both computers use the RTS (Request To Send) and CTS (Clear To Send) lines of the RS-232C interface.

Many terminal programs and file transfer programs for home computers do not implement flow control in software, and many so-called RS-232C ports do not support hardware flow control. Even if the RTS and CTS lines appear at the connector, software that directly reads the CTS line may be required in order for flow control to be implemented. If you find that the TNCs seem to lose data during file transfers, immediately suspect a flow control problem.

XON/XOFF FLOW CONTROL

If you are using a terminal rather than a computer, or if your computer does not support RTS/CTS flow control, you can use the XON/XOFF flow control. You can enable this method by setting XFLOW ON. The special flow control characters are set to CTRL-S and CTRL-Q by default.

In the command mode, the TNC input buffer may fill up if you try to type too long a command. In the data mode, the buffer may fill up if you are using your computer to transfer data at a rate that is faster than the data rate for radio transmission, if radio data transmission has slowed down because of noise or other users on the channel, or if the operator or computer at the other end has stopped the output from his TNC. The TNC will send the terminal an XOFF character when there is room remaining for about ten characters in the buffer. If you continue sending data until there are only five spaces left, the TNC will send an XOFF character after each character received. When the buffer fills up completely, data will be lost. When the buffer empties out, the TNC will send a single XON character to the terminal.

If you disable XON and XOFF by setting them to 0, the TNC will automatically use the RTS/CTS flow control to stop input from the terminal.

XON/XOFF flow control is normally disabled in the transparent mode. This is done as all characters are treated as data; therefore, the XON and XOFF characters will not be recognized. If you can not use RTS/CTS flow control, you may enable the XON and XOFF characters (the commands from the TNC to the terminal) by setting TXFLOW ON and XFLOW ON. However, START and STOP characters (the commands to the TNC from the terminal) will still be treated as data.

HARDWARE FLOW CONTROL

This method of flow control is preferred, since it usually does not depend on the programming of a particular communications program. The RS-232C handshaking signals are described in the "Circuit Description."

DISPLAY OPTIONS

Several parameters control the way output is formatted for display on your terminal. Most of these are parameters that are determined by the display capabilities of your terminal and would be changed only if you changed terminals.

In the converse mode, it is natural to choose a line-termination character such as a <cr> or <lf> to terminate packets. However, for some applications, you may want to use an "invisible" command character to force the TNC to transmit a packet. In the first case, the send-packet character is interpreted as part of the input as well as a command; in the second case, it is a command only. You can choose either option with command CR. If CR is on, the send-packet character is data, and is echoed to the terminal and included in the packet. You should disable CR if you are using packet timeout (CPACTIME ON) in the converse mode.

A common occurrence when two stations are exchanging packets is for incoming packets to arrive when the operator is in the middle of typing a line. In order to prevent the new line from disrupting the screen display, you can enable FLOW. In this mode, output to the screen is disabled as soon as you begin to type, and is enabled when a packet is completed. If you want to see the incoming packet before you transmit your line, you can type the redisplay-line character, which is set by the command REDISPLA. This will display the incoming packet and then retype your partially completed line. If FLOW is OFF and an incoming packet disrupts your typing, you can also use this character to redisplay your input line.

The parameter SCREENL sets the width of the terminal screen or page. Whenever this number of characters has been sent to the terminal without an intervening <cr>, a <cr> is inserted in the output. A <cr> is also echoed if you type a line that exceeds the width of your screen; however, the extra <cr> is not included in a packet. If your terminal performs automatic line-wrap, you should disable this feature by setting the SCREENL parameter to 0. The TNC does not carefully distinguish between printing and nonprinting characters and it does not correct its line count for horizontal tab characters; however, backspace characters are counted correctly.

For normal display, a <cr><lf> is the new-line sequence. However, you terminate a line of typing with a single character, usually a <cr> (called RETURN or ENTER on some terminals). If only a <cr> is displayed, the next line will be typed over the previous one instead of appearing below it. Some terminals automatically display an <lf> following each <cr>, but most do not. If auto line-feed is enabled by AUTOLF, the TNC will add a <lf> after every <cr> displayed or echoed.

The conventional response to character deletion on display terminals is to feed a backspace/space/backspace sequence to the terminal. This removes the character from the screen and leaves the cursor ready to type a new character in its place. On a hardcopy terminal, however, this results in unreadable text. If backspace display is disabled with BKONDEL OFF, a backslash symbol (\) will be displayed for each character deleted. You can use the redisplay-line character to see the corrected line.

The <escape> character (ASCII code 27 or hexadecimal $1B) is used by many terminals to control cursor movement and special display modes. (NOTE: Throughout this Manual, the dollar sign symbol ($) prefaces all hexadecimal numbers). If you do not want this effect, enable the <escape> translation with ESCAPE OFF. This will cause all <escape> characters to be sent to the terminal as the dollar sign symbol. This does not affect <escape> characters that are transmitted in packets.

Some terminals echo characters typed in locally, before the character is transmitted to the I/O port. Also, some terminal programs on computers may perform local echoing. If the TNC also echoes characters, you will see two of every character. You can disable the echo mode with ECHO OFF.

A few terminals require particularly long times to respond to <cr>s or <lf>. Some hardcopy terminals require time to move the print head to the beginning of the line following a <cr>. Some display terminals require long times to scroll their screens following a <lf> character. If characters are sent to such a terminal before it is ready, the characters will be lost. If your terminal always loses a few characters at the beginning of a line, you need to enable null insertion. A null is a character with ASCII code 0; and the TNC does not actually transmit nulls in this mode, since they are misinterpreted by some computer's terminal programs as a BREAK signal. The number of null intervals is set by the command NULLS, and null insertion after <cr>s and <lf>s is separately enabled by NUCR and NULF.

EDITING COMMANDS

Several characters are used to correct mistakes in the text typed into the TNC. Except in the transparent mode or if timed packets are in effect in the converse mode, no text characters are interpreted by the TNC until it receives a <cr> or the send-packet (in the converse mode). Until then, you can delete and retype characters or cancel the line completely.

Control characters are normally chosen as editing characters. You can disable editing functions by setting the character used for the function to $00. This prevents any character (even a null) from being matched to that function.

The key usually used to remove the last character typed on a line may be either a <delete> character (hexadecimal $7F) or a back space (hexadecimal $08 or CTRL-H). The character used is determined by the DELETE command, which has options ON (<delete>) and OFF (back space). The key used for rub out on your terminal may be labeled "back space", "delete", "rub out", or simply "←". You may have to experiment to find out if it produces a back space or a <DELETE> character.

Attempts to delete past the beginning of a line, or past the beginning of a packet in the converse mode, will have no effect. You can not delete a <cr> or any of the special characters not inserted into the text, such as the send-packet character or flow-control characters. These characters cause actions which take place immediately. Some home computers allow you to fix errors in input lines by backing the cursor up to the mistake, retyping the incorrect characters, and then spacing forward with an arrow key. The TNC does not have this kind of input editing feature; therefore, you must delete all the characters back to the ones you want to change and then retype the rest of the line.

If you wish to cancel an entire line, you can delete text back to the last-occurring <cr> by typing the character specified by the CANLINE command (default CTRL-X). In the converse mode, this character functions like the cancel-packet character unless you have set the send-packet character to something other than <cr>.

In the converse mode, you can delete text back to the last-occurring send-packet character by typing the character specified by the CANPAC command (default CTRL-Y). This will delete any intervening <cr>s. You can not cancel packets which have already been placed in the outgoing packet buffer.

The cancel-packet character has a special function in the command mode. Typing this character will cause the terminal output buffer to be flushed and all output to be diverted to the write-only memory. You can resume normal output by typing another cancel-packet character or by changing modes; that is, going into the converse mode. Echoing of input is not affected by this command. This feature makes it possible to get rid of any unwanted output which may occur.

You may occasionally wish to transmit one of the characters that you have assigned to a special function. The pass character is intended to increase the flexibility of the converse mode by providing this capability. To insert such a character into the input buffer, precede it with the character specified by the PASS command (default CTRL-V). You can send any character this way, including nonspecial characters and the pass character. Since the pass character is kept in the buffer until a packet is formed, two <delete>s are required to remove both the quoted character and the pass character. Note that the PASS character will cause only one special character to be inserted; therefore, you must type it again for each such character.

You can use the pass character in the command mode in order to include <cr>s in the beacon text.

SPECIAL OPERATING CONFIGURATIONS

The primary function of the TNC is to enable you to communicate with other amateurs via packet radio. The Heath TNC implements two protocols (sets of rules) for formatting messages to other TNCs. These protocols, AX.25 and VADCG, are both designed primarily for point-to-point, two-party communications. However, you can also use them to simulate the common amateur net or round-table type of contact. You can specify the AX.25 protocol with the command AX25 ON, or the VADCG protocol with the command AX25 OFF. Packets heard by the TNC will be ignored if they do not correspond to the selected protocol.

Earlier in this section, you learned how to set your call sign and issue the CONNECT command to talk to a specific station. These commands are the beginning of packet operation, which we will now discuss in more detail.

In order to establish a two-way connection, the TNC must know your station address and the address of the party you wish to talk to. To prepare your TNC for amateur radio operation, first establish your call sign as the station address by using the command MYCALL (or just MY). This sets the string used to identify packets transmitted by your station. If you plan to use the VADCG protocol when you operate, you must also set your VADCG address by using the command MYVADR. The VADCG address is a number from 1 to 31, which is used to identify VADCG packets once a connection is established. Neither protocol will work properly if there is more than one station on the air with a given address. To avoid conflict with other stations operating in your area, you must coordinate the VADCG address for your TNC with them. If you will have more than one station operating the AX.25 protocol using your call sign, you can give them different addresses using the substation ID (SSID) extension, a number from 0 to 15. This number is appended with a dash, as in this example:

```
MYCALL W8XYZ-3
```

If you don't specify the SSID extension, it will be 0. The extension does not affect the Morse code ID of your station.

The call sign specified by MYCALL is ordinarily used by the TNC for Morse code identification. This call sign is sent automatically every 9-1/2 minutes if you have sent a packet in the previous 9-1/2 minutes (or as soon thereafter as the channel is available). In some locations, the address string included in the packets may be considered adequate identification for legal purposes. However, notice that the call sign address within the packets is actually bit-shifted ASCII (occupying bits 1-7 of the byte rather than bits 0-6). Also, if you use the VADCG protocol, your call sign is included only in the initial packets establishing a connection. Make sure you identify your station in either Morse code or voice. This makes it easy for other operators to tell who is on the air without displaying every packet. It also improves relations with nonpacket operators using the same frequencies.

You can disable the automatic Morse code ID with the command CWID OFF. If you wish to use an identifier other than your call sign as your packet address, you can set the Morse code ID with the command IDTEXT. This also allows you to include other information along with the call sign in the ID. If you disable the automatic ID, you can still command the TNC to identify in Morse code at any time with the command ID.

You have already been introduced to the CONNECT command, which causes your packets to be sent to a specific station. If the station you wish to talk to is a little too far away for you to contact directly, you can use the digipeating feature of the TNC. This feature works differently, depending on whether you use the AX.25 or VADCG protocol. A digipeater accomplishes much the same task as an ordinary repeater in extending the range over which you can communicate. The difference is that your messages are copied and relayed by the digipeating packet station. This results in better quality of the signal received at the destination at the expense of some delay while the intermediate message is received and retransmitted.

To request digipeating under the AX.25 protocol, you must specify the intermediate packet station or stations which you want to relay your messages. You do this as part of the CONNECT command by using the VIA subcommand:

```
CONNECT WA8XYZ VIA N0XYZ-2, K7XYZ
```

You should list the intermediate stations in the order in which you want them to relay the packets as they go from your station to the destination station. In this example, your connect message to WA8XYZ will be repeated by N0XYZ-2 and then by K7XYZ. Reply packets from WA8XYZ will be relayed first by K7XYZ and then by N0XYZ-2.

You can specify as many as eight intermediate stations; however, keep in mind that using more than one digipeater is an extension to AX.25 and may not be compatible with other implementations of this protocol. The delay between your transmission and the receipt of a reply will naturally increase as more intermediate relays are used. Also, the possibility of losing information due to interference or noise on the channel increases.

You can specify intermediate digipeaters to be used for unconnected packets by using the UNPROTO command with the same format as the CONNECT command:

```
UNPROTO QST VIA N8XYZ
```

This will cause packets sent when you are not connected to another station to be sent to QST (rather than the default of CQ), digipeated by N8XYZ.

To request digipeating under VADCG protocol, give the command:

```
VRPT ON
```

This causes your packets to be repeated by any VADCG digipeater. For a Heath TNC to act as a VADCG digipeater, you must give the command:

```
VDIGIPEA ON
```

The VADCG protocol will not function properly if there is more than one VADCG digipeater in an area, since all digipeaters will simultaneously attempt to relay packets requesting digipeating.

For special applications, you can disable the TNC's ability to connect or to transmit. If you have left your TNC running to monitor channel activity in your absence, but you wish to inhibit it from transmitting, set XMITOK OFF. The TNC will perform normal operations in this condition, including formatting and "sending" packets; however, it will not key the transmitter. You may also wish to specify CONOK OFF. This prevents the TNC from accepting connect requests from other stations (although it does not stop you from initiating a connect request of your own).

If a connect request is received when CONOK is OFF, the TNC will send a "station busy" packet to the requesting station and display a message such as:

```
*** connect request: KB8XYZ
```

identifying the requesting station. If the TNC receives a "station busy" message in response to a connect request, it will display a message such as:

```
*** K0XYZ station busy
```

showing the call sign of the station you tried to connect to. These messages are also used if a TNC is connected to another station when a request is received.

In addition to transmitting information typed in from a data mode, you can command the TNC to send a specific message at regular intervals. This message is called a "beacon." You can use the function to send announcements or allow other packet users to test their equipment. To set the beacon text to your message, type the command:

```
BTEXT
```

Everything you enter on the command line following the space after BTEXT will be entered into your message string.

Use the BEACON EVERY command to set the interval between your beacon messages. If you wish the beacon to transmit at 30-minute intervals, for example, give the command:

```
BEACON EVERY 180
```

You can specify any value between 0 and 255 for n in the BEACON EVERY n command, where n specifies 10-second time intervals. A value of 0 is the default value and turns the beacon off, while 255 specifies 2550 seconds (or 42.5 minutes). If local activity is high on your operating frequency, it is wise to send regular beacon messages at 30-minute or longer intervals.

The beacon function also has a transmit-after mode, in which a beacon packet is only transmitted after activity is heard on the channel. You can use this feature to leave a message for other packet users. If someone initiates a connection (or sends anything, for that matter) on an otherwise idle channel, a beacon can be sent a short time later with a message such as "I'll be back on the air on packet after dinner — call me then." If the station is monitoring beacon packets (see monitor mode discussion on Page 2-14), the operator will see this message. No beacons are sent in this mode if there is a lot of packet activity on the channel, since the required period of quiet will not occur.

PACKET TIMING FUNCTIONS

Five adjustable timing parameters are provided for configuring the TNC to your particular radio environment. Some other parameters related to the timing parameters are discussed here as well.

The time delays that are required in switching from receive to transmit and from transmit to receive vary greatly among various amateur radio equipment. When two stations are sending packets back and forth, these delays must be allowed for. If data is sent before the transmitter is operating, the packet will not be transmitted properly. Similarly, if the receiving station has not had sufficient time since it stopped transmitting for the receiver to become active, data will be lost. The delay between transmitter keyup and the beginning of data transmission is controlled by the command TXDELAY. Ordinarily, this parameter should be set to the same value by all members of a local packet group, and it should be determined by the slowest pair of stations in the group.

If you are transmitting packets through an audio repeater, you may require a considerably longer keyup delay than what is required for direct communications. The command AXDELAY allows you to specify an additional keyup delay to allow the repeater receiver and transmitter to lock up. If the repeater has a long "hang time" and stays up for a while after the other station has stopped transmitting, you can make use of this time with the AXHANG command. If the TNC has detected channel activity recently enough that the repeater should still be "up," it will wait only a time that equals the TXDELAY before sending data, rather than adding an AXDELAY time as well.

The parameters set by TXDELAY, AXDELAY, and AXHANG are all specified as numbers between 0 and 15. The actual delay in milliseconds is a multiple of the input parameter, 40 ms per count for TXDELAY and 120 ms per count for AXDELAY and AXHANG. During the time the TNC is keying the transmitter but not sending data, it will transmit a synchronizing signal (flags). Thus, the total keyup delay will only be:

Keyup delay = TXDELAY*40 + AXDELAY*120

in milliseconds. If channel activity has been heard more recently than AXHANG*120 ms ago, the keyup delay will be:

Keyup delay = TXDELAY*40

in milliseconds. If it takes your radio an exceptionally long time to key up, you can use AXDELAY to augment the maximum delay available with TXDELAY by setting AXHANG to 0.

Both AX.25 and VADCG protocols provide for retransmitting packets if no acknowledgment is heard from the connected station within a certain period of time. A packet may not be acknowledged due to channel noise or "collision" with another packet transmission. Since there may be other stations on the channel, the receiving station may not be able to acknowledge the received packet immediately. The time lapse before the originating station retransmits the packet is set by the command FRACK (frame acknowledge time). The maximum number of retransmissions before the originating station terminates the connection is set by the command RETRY. The maximum number of transmissions of a packet is RETRY + 1, since the initial transmission does not count as a retransmission.

The frame-acknowledge time is automatically corrected for the additional time required for digipeating. An extra time delay is added for each transmission, which must be made after origination of the packet in order to deliver the packet and receive the acknowledgment. The time interval before the TNC retransmits an unacknowledged packet is therefore:

Retry interval = FRACK * (2*n + 1)

in seconds, where n is the number of calls in the digipeat field of the address.

Both the AX.25 and VADCG protocols specify that acknowledgments of digipeated packets be made from end to end. This means that intermediate digipeaters do not acknowledge the packets they digipeat. When the destination station receives the packet, it generates an acknowledgment which is sent through the reverse route used by the original packet. If there are several intermediate relays, the chance of either the original packet or the acknowledgment to being lost increases drastically. To help alleviate this problem, an automatic wait time can be imposed on any station not transmitting a digipeated packet. Any station ready to transmit a packet immediately after the carrier drops is required to wait for this time interval unless it will be transmitting one or more digipeated packets. This means that the chance of a collision involving a digipeated packet is reduced since, once a transmission begins, other stations will wait for a clear channel. The digipeat wait time is set by the command DWAIT, which specifies 40 ms intervals. If no digipeating is being done by anyone in the local area, you can set this parameter to 0. In any event, however, it should be set to the same value by all members of a local packet group.

In order to avoid unnecessary packet retries with the associated channel load, the TNC implements a collision-avoidance strategy which applies to all packets except those being relayed. On the second and subsequent transmission of a particular packet, the TNC waits an additional random time after detecting a clear channel before beginning transmission. This strategy is based on the assumption that packets not acknowledged suffered collisions with transmissions from other stations. If the random waiting time is spent, repeated collisions of transmissions by the same two stations can be prevented, since eventually they will wait different time periods and one station will capture the frequency. The random time is a multiple (0-15) of the TXDELAY time. This is because TXDELAY represents the interval during which a transmitter may have been keyed but can not yet be detected by other stations. The interval, in milliseconds, between the TNC detecting carrier-drop and beginning to transmit is:

Wait time = DWAIT * 40

for the first transmission of a packet. For subsequent transmissions of the same packet, the interval is:

Wait time = (DWAIT + r * TXDELAY) * 40

where r is a random number from 0 to 15.

Both AX.25 and VADCG protocols allow multiple packets to be transmitted before waiting for an acknowledgment. This permits more efficient channel use when large amounts of data are being transferred. The maximum number of packets which the TNC will send before waiting for acknowledgment is specified by MAXFRAME. Of course, the TNC will not wait until MAXFRAME packets have been entered before transmitting — this parameter is only used to limit the transmission if more than one packet is ready when the TNC begins to transmit. MAXFRAME, in combination with PACLEN, determines how much information can be sent in a single transmission. The best combination for efficient data transfer is determined partly by the channel quality and partly by the rate at which the terminal can process data. For a 1200 baud terminal data rate, you should start with a combination that produces about 300 characters outstanding at one time.

The radio data transmission rate is set by the command HBAUD. This command selects a baud rate from a table of standard rates similar to the command ABAUD for terminal baud rate. Note that there is no relationship between terminal baud rate and radio baud rate. Also, be aware that the baud rate table is different for HBAUD and ABAUD. A 400 baud option for radio baud rate is included for AMSAT operations, and only rates through 4800 baud are supported (9600 with high-speed clock option). In order to communicate with another packet station, you must use the same radio baud rates. The length of time required to send a given amount of information depends inversely on the baud rate, so that it takes four times as long to send a line at 300 baud as at 1200 baud. If you use slow radio baud rates, you should either limit the length of transmissions determined by MAXFRAME and PACLEN so that the hardware watchdog timer does not disrupt your transmissions, or disable or modify the watchdog. The Bell-202 compatible modem is the optimum design only for 1200 baud radio data rate. For HF operation, at low baud rates, you should consider configuring the modem for a different set of tones. The modem is not useful at rates higher than 1200 baud, although the TNC will provide data signals at up to 4800 baud with the standard clock; an external modem is required for such operation.

MONITOR FUNCTIONS

Although the AX.25 and VADCG protocols are primarily oriented toward setting up "circuits" between two stations, this is not the way many amateurs operate. The TNC can also operate in a mode more suitable for a "net" or "round-table" discussion with several participants, although reliable reception of your transmissions by every station can not be guaranteed. This is done by enabling the monitor functions.

Monitoring is enabled by the command MONITOR ON, and separate monitor functions are individually enabled. This set of functions allows you to see displayed packets from selected stations or classes of stations. You can list up to 10 call signs of stations to monitor with the commands MFROM and MTO. Packets are displayed if any of the call signs specified by MFROM appear in the "from" field of the packet address, or if any call signs specified by MTO appear in the "to" field. If you specify ALL in place of either of these lists, you will see all the packets your TNC receives. If you specify NONE in place of a list, that list will not be used to select monitored packets. Although a list of 10 call signs would be too long to store in the TNC's NOVRAM memory, if you have MTO or MFROM set to ALL when you give the PERM command, this information will be saved; otherwise, the NONE choice will be saved.

Monitored packet display is somewhat different from the display of connected packets. Each packet is displayed with the source and destination stations identified:

KC8XYZ>N9XYZ:Go ahead with the file transfer.

If a connected packet QSO is taking place on the frequency of your group conversation, you may wish to ignore all connected packets while your group operates in unconnected mode. The command MALL OFF will cause connected packets to be ignored. If you want to be able to monitor packet activity when your station is not connected but have the feature automatically disabled when you connect to someone, you should command MCON OFF. If you have MALL ON and MCON ON, and you are monitoring the station you are connected to, packets from that station will be displayed only in the monitor format and not in the usual manner with no station identification.

You can operate a group conversation with some data integrity by having the stations connect in pairs and set MALL ON and MCON ON. This does not insure that every packet is received at every station, but it does insure that a packet involved in a collision will be retried. You may occasionally see duplicate copies of a packet in this mode if the acknowledgment packet is lost. If you have an odd number of stations participating in this sort of conversation, one station can connect to itself via another station as digipeater. This station will have the disadvantage of seeing its own packets redisplayed. For example, suppose WB6XYZ, WD0XYZ, WA0XYZ, W1XYZ, and K4XYZ wish to carry on a group conversation. In order to make all the transmissions as reliable as possible, the following connections are made:

WB6XYZ connects to W1XYZ
WA0XYZ connects to K4XYZ
WD0XYZ connects to WD0XYZ via W1XYZ

If each station specifies MCON ON, MALL ON, and MTO ALL, each station will see the packets sent by all the others.

NOTE: Future revisions of the Heath TNC software may support multiple connects. As soon as these enhancements become available, the version number will appear in the HD-4040 catalog copy. The new ROMs will then be available from the Heath Company Parts Replacement department. For parts ordering information, please refer to "Customer Service" (inside the rear cover of this Manual).

HF AND OSCAR

When configured in the default mode, the Heath TNC is optimized for a local VHF FM environment. The modem is configured for best response at 1200 baud. The settings of MAXFRAME and PACLEN provide the possibility of several continuous frames of long data length. The type of data link offered by the average HF or OSCAR 10 path is very different. Lower signal-to-noise ratios require lower baud rates; noise spikes and fades require shorter packet lengths, and a higher rate of false carrier detects lowers the total, usable dynamic range in the audio input. The TNC hardware and software provide many methods to improve throughput in high noise environments. While you can obtain the best results through your own experimentation, some problem areas and hints are given below.

The TNC detects a busy channel through the lock-detect signal from the demodulator. The presence of a lock-detect signal is indicated by the Data Carrier Detect (DCD) LED. For best results, the LED should be off when no data is present. Decrease your receiver volume to the point where the LED is fully lit when data is present and only flickers occasionally at other times. In most cases, it will not be possible to keep the LED completely off. On a noisy channel, many spurious lock-detect signals may be generated. Each time DCD comes on, the TNC will start another DWAIT interval before it considers the channel to be clear, usually resulting in long periods of silence from your TNC. For this reason, you must set DWAIT to 0 for HF or OSCAR operation. You can disable the random retry wait by setting TXDELAY to 0 and using AXDELAY to force a suitable keyup delay. Note that the units for TXDELAY and AXDELAY are different, however.

If you are operating a full-duplex radio station (simultaneous transmit and receive), such as an OSCAR 10 station, you should set the parameter FULLDUP. While the TNC is always capable of full duplex operation, this parameter will cause the protocol to behave slightly differently in acknowledging packets. Also, the state of the DCD line will be ignored. You may be able to improve operation somewhat by disconnecting the DCD line at the modem connector (J5).

Intuition tells you that lower baud rates will reduce the number of packet retries. In fact, there is usually a small range between too fast and too slow. A slower packet is a larger target for fades and static crashes, and a complete packet must be received correctly.

You can improve the response of the modem for low baud rates by calibrating the modulator and demodulator tones for a narrow shift. This also permits the use of a narrow filter by the receiving station. You may also wish to move the demodulator center frequency to a lower value. You can change modem parameters and timing constants by using different parts on headers U34 and U35. Consult the EXAR notes listed in the Bibliography for the straightforward design procedure. The use of the calibration routines for special tones is discussed in the "Calibration" section.

The audio frequency response of the TNC in combination with an SSB transceiver is likely to be less than optimum. If you will operate your TNC on HF or OSCAR, and your transceiver does not require the 560 ohm load resistor (R59), do **not** install it, as it degrades the low frequency response. You can modify the input filtering provided by the MF-10 switched capacitor filter by changing the values of the components on dip header U30. Various alternate filter configurations are listed from time to time in TAPR's (Tucson Amateur Packet Radio's) newsletter, "Packet Status Register."

You can bypass the onboard modem completely by using J5. This allows you to use exotic modulation methods and higher baud rates. The interfaces available on J5 are TTL-type levels and **not** RS-232C. Be sure to refer to Page 5-1 in this Manual for more information.

COMMANDS AND MESSAGES

COMMAND SYNTAX

Many variable **parameters** are used by the TNC in its operation, such as your call sign, terminal type, display preferences, and the characteristics of your radio. In addition, you can command the TNC to perform several **tasks**, such as connecting to another station in order to start a conversation, disconnecting at the end of the QSO, saving information in NOVRAM, or displaying information about itself. You can change parameters and issue instructions to the TNC by typing **commands** composed of English-like words or word abbreviations called **keywords**, or by typing **variables** consisting of numbers or strings of characters that you select. You will probably never change some of these parameters; however, the TNC has been designed to give you maximum flexibility so you may adapt it to your particular environment.

In the following paragraphs, all commands are listed alphabetically. If a command has parameters, each parameter is described and the default value is given. The defaults are the EPROM's stored values, which you may load (instead of those in NOVRAM) by setting switch SW-1 to ON prior to performing a reset. Those parameters that can be saved and recalled after power-down are marked with an * (asterisk) at the right margin. Each parameter is described and the possible values are given. A more detailed discussion of many of the commands and their interrelationships is given in the "Operation" section. Enter the command to the TNC by typing it when you see the command-mode prompt,

```
cmd:
```

The command keywords and parameters are separated by spaces, and the TNC takes action after you press the RETURN key. You may enter keywords in upper- or lowercase. Except for the beacon and ID text string, everything you enter in the command mode is translated to uppercase before it is examined. You may abbreviate all commands and alphabetic parameters to the shortest unique string. These minimum abbreviations are shown in **bold** type in this section of the Manual.

There are several types of parameters. A parameter denoted as "n" is a number, and can be given either in decimal or in hexadecimal (base 16). When the TNC shows some of these parameters (those which set special characters), they will be given in hexadecimal. A hexadecimal number is distinguished from a decimal number by the "$" prefix that precedes it. The "digits" of a hexadecimal number represents powers of 16, analogous to the powers of 10 represented by a decimal number. The numbers 10 through 15 are denoted by the hexadecimal digits A through F. For example,

$1B = 1*16 + 11 = 27
$120 = 1*16*16 + 2*16 = 288

The TRACE command parameter is given as a bit-code. This means that several related values are simultaneously set by this command, and the parameter is formed by adding together the numbers corresponding to each value desired. You may find it convenient to think of this number in hexadecimal.

Many parameters are "flags," meaning that they have two possible values, **ON** and **OFF**, or **YES** and **NO**. All of the commands descriptions show **ON** and **OFF** as the options; however, you may type **YES** and **NO** instead. A few parameters are really flags, but rather than indicating that something is "on" or "off", they select one of two ways of performing a task. Some of these parameters have the values **EVERY** or **AFTER**, indicating how a time interval for a repeated action is to be treated. Others are **CONVERS** or **TRANS**, indicating operating modes for data transmission.

Several commands require call signs as parameters. While these parameters are normally amateur radio call signs, they may actually be any collection of numbers and at least one letter (up to six characters); they are used to identify stations sending and receiving packets. A call sign may additionally include an "extension," a decimal number from 0 to 15 that is used to distinguish two or more stations on the air with the same call sign (such as a base station and a repeater). You enter the call sign and extension, which are then displayed as call-ext; that is, W8XYZ-3. If you do not enter the extension, it is set to −0; also, extensions of −0 are not displayed by the TNC.

Several parameters are numerical codes for characters which perform special functions. The code is simply the ASCII character code for the desired character. These characters have **control characters** as default values. You enter a control character by holding down a special **control** key on the keyboard while you press the indicated key.

There are two commands, BTEXT and IDTEXT, which have a text string as parameters. This string can be any combination of letters, numbers, punctuations, or spaces up to 128 characters. You can even put characters with special meanings, such as RETURNs, into the string by preceding them with the "pass" character. The string ends when you type a (non-passed) RETURN.

In the following command descriptions, the keywords are shown in uppercase. User-supplied values are shown in lowercase. If you must choose a parameter from one or two values, the choices are separated by a vertical bar. Optional parameters are shown in square brackets. For example,

KEYWORD var A/B [C/D]

This means that the command KEYWORD requires a user-supplied variable "var" and either A or B. In addition, you can optionally specify C or D.

You can examine the value of any parameter by typing the command which sets this parameter followed by a RETURN. A special command, DISPLAY, allows you to see the values of all parameters or groups of related parameters.

USER COMMANDS

The commands listed in this section of your Manual are recognized by your TNC. In addition to the following description of each user command, you will find a summary of the most frequently used commands in the "Quick Reference Guide."

ABAUD n *default: not applicable* *

parameters

n selects a baud rate from the list below.

This command sets the baud rate used for input and output through the serial port. The parameter n selects one of the following baud rates:

0 Reserved for future use.

50	1800
75	2400
110 #	3600
135	4800
150	7200 ☆
300 #	9600 #☆
600	19200
1200 #	

— These baud rates are detected by the autobaud routine which operates when the TNC is initialized with the default parameters.

☆ — The TNC may not be able to perform its other functions while sending or receiving data at these baud rates. Use of baud rates higher than 4800 is not recommended for computer file-transfer applications.

The baud rate change will not take effect until you have issued a RESET command. The TNC enters an auto-baud routine and attempts to match the terminal baud rate if you use switch SW-1 to select default parameters. If you initialize the TNC with default parameters, the auto-baud routine will set ABAUD to reflect the baud rate it detects.

ABIT n *default: 1* *

parameters

n 1 – 2, specifying number of stop bits.

This value selects the number of stop bits used by the 6551 UART (U14). The number of stop bits will not change until you perform a reset.

AUTOLF ON/OFF *default: ON* *

parameters

ON A line-feed command is sent to the terminal after each RETURN.

OFF A line-feed command is not sent to the terminal after each RETURN.

This option should be ON when a terminal that does not automatically send a line feed after a RETURN is used. If AUTOLF is ON, a line-feed command is sent to the terminal after RETURNs in received packets as well as echoed RETURNs received from the terminal. This command only affects what is displayed, not the data sent in packets.

AWLEN *default: 7* *

parameters

n 7 – 8, specifying number of data bits per word.

This value defines the word length used by the 6551 UART. A word length change will not take effect until a reset is performed. Except in the transparent mode, only the low order seven bits of a word are kept. To use all eight data bits in the transparent mode, you must set AWLEN to 8.

AX25 ON/OFF *default: ON **

parameters

ON	The TNC operates in AX.25 protocol.
OFF	The TNC operates in VADCG protocol. NOTE: Also refer to the **MYVADR**, **VDIGIPEA ON/OFF**, and **VRPT ON/OFF** commands.

If you specify ON, the TNC will originate packets and other link operations in the AX.25 mode. Packets received in the VADCG protocol when AX25 is ON, or packets received in the AX.25 protocol when AX25 is OFF, will not be interpreted or displayed.

AXDELAY n *default: 0 **

parameters

n	0 – 15, specifying 120 ms intervals.

This value specifies a period of time to wait in addition to TXDELAY after keying the transmitter before data is sent. This will be used by groups using a standard "voice" repeater to extend the range of the local area net. Repeaters using slow mechanical relays, split sites, or combinations of both require some amount of time to get on the air. AXDELAY can also be used when the receiving TNC's rig has a very slow PLL or squelch.

AXHANG n *default: 0 **

parameters

n	0 – 15, specifying 120 ms intervals.

You can use this value to increase channel utilization when you use an audio repeater with a hang time greater than 120 ms. If the repeater squelch tail is long, it is not necessary to transmit if the repeater is still transmitting. If the TNC has detected a packet sent within the AXHANG period, it will not add AXDELAY to the key-up time.

BEACON [EVERY/AFTER] n *default: EVERY 0 **

parameters

EVERY	Send beacon every specified interval.
AFTER	Send beacon once after the specified interval with no link activity.
n	0 – 255, specifying 10 second intervals.

The beacon mode is turned on by this command, where n specifies a time in multiples of 10 seconds. A value of 0 for n turns the beacon off. If you use the optional keyword EVERY, a beacon packet is sent only every n × 10 seconds. If you use AFTER, a beacon is sent only after n × 10 seconds have passed with no activity detected on the RF channel; however, the beacon is sent only once until further activity is detected. You can use this mode to send announcements or test messages only when packet stations are on the air, and avoid adding unnecessary activity on the channel.

A beacon frame consists of the text specified by BTEXT in a packet addressed to "BEACON" and sent via the digipeat addresses specified by the UNPROTO command. These frames can be monitored from other TNCs by setting MONITOR to ON and MTO to BEACON.

BKONDEL ON/OFF *default: ON **

parameters

ON	The sequence backspace-space-backspace is echoed when you enter the DELETE character.
OFF	The backslash character (\\) is echoed when you enter the DELETE character.

This command determines the way the display is updated to reflect a character delete in the command mode or converse mode. The backspace-space-backspace sequence will properly update the screen of a video display.

BTEXT text

parameters

text — any combination of characters and spaces.

BTEXT specifies the content of the data portion of the beacon packet. The default text is:

AX.25 level 2 protocol software version X.Y.

where X.Y. represents the software version number. You can include the RETURN character in the text by using the pass character (default is CTRL-V) preceding RETURN. You can specify a maximum of 128 characters. The beacon text is not stored in NOVRAM.

CALIBRA

Use the CALIBRA command to transfer control to the hardware calibration routine. The commands available from this routine are described in the "Modem Calibration" section of the "Set-Up Instructions." You may calibrate the unit at any time without altering the current link state.

CANLINE n *default: $18 <CTRL-X> **

parameters

n 0 – $7F, specifying an ASCII character.

Use this command to change the cancel-line input editing command character.

CANPAC n *default: $19 <CTRL-Y> **

parameters

n 0 – $7F, specifying an ASCII character.

Use this command to change the cancel packet terminal editing command character.

This character functions as a cancel-output character in the command mode. Typing the cancel-output character a second time restores normal output.

CMDTIME n *default: 1 **

parameters

n 0 – 15, specifying one second intervals.

Use this command to set the transparent mode time-out value. In order to allow escape to the command mode from the transparent mode, while permitting any character to be sent as data, a guard time of CMDTIME seconds is set up. You must enter three characters within the CMDTIME of each other, with no intervening characters, after a delay of CMDTIME since you typed the last character. After a final delay of CMDTIME, the TNC will exit the transparent mode and enter the command mode. You should then see the following prompt:

```
cmd:
```

If CMDTIME is 0 (zero), the only exit from the transparent mode is a hardware reset.

COMMAND n *default: $03 <CTRL-C> **

parameters

n 0 – $7F, specifying an ASCII character.

Use this command to change the entry character for the command mode. You enter the command mode from the converse mode when you enter this character from the terminal. See the "Operation" section for information on how to use the command character in the transparent mode.

CONMODE CONVERS/TRANS *default: CONVERS **

parameters

CONVERS	Connects cause automatic entry to the converse mode.
TRANS	Connects cause automatic entry to the transparent mode.

CONMODE controls what mode the TNC will be placed in after a connect. The connect may result either from a connect request received over the air or a connect initiated by a CONNECT command. If the optional mode is given with the CONNECT command, the CONMODE parameter will not be used. If the TNC is already in the converse or transparent mode when the connection is completed, the mode will not be changed. If you have typed part of a command line when the connection is completed, the mode change will not take place until you complete the command or cancel the line.

CONNECT call1 [VIA call2[,call3...,call9]] [CONVERS/TRANS]

parameters

call1	Call sign of TNC to be connected to:
call2	Optional call sign of TNC to be digipeated through. You can specify as many as eight digipeat addresses.
CONVERS	Enter the converse mode upon successful connect.
TRANS	Enter the transparent mode upon successful connect.

Each call sign can have an optional extension designator specified as $-n$ immediately following the call sign, if AX25 is ON. The digipeat fields are specified in the order in which you want them to relay the packets to the destination.

This command does not change NOVRAM values, and it has immediate effect. It initiates a connect request to TNC **call1** optionally through digipeaters and, if successful, will enter the specified data transfer mode. An error message is returned if the TNC is in a connected state, or is already attempting to connect or disconnect. If there is no response to the connect request after RETRY attempts, the command is aborted, a message is typed, and the TNC remains in the command mode.

It you do not specify the optional mode parameter, the mode given in CONMODE is used. If the mode parameter is used, it overrides CONMODE.

If AX25 is ON, the VIA option is used to request digipeating. If AX25 is OFF, the status of the VRPT parameter controls digipeating. NOTE: Digipeating by specific stations can not be specified in the VADCG control.

Example (AX25 is ON):

cmd: CONNECT W8XYZ VIA K0XYZ-1,
WB0XYZ CONVERS

This commands the TNC to initiate a connect request to W8XYZ, with the connect packets and subsequent data packets to be digipeated through K0XYZ followed by WB0XYZ. Packets sent in the opposite direction access the digipeaters in the opposite order. Thus, packets from W8XYZ will first be repeated by WB0XYZ, then by K0XYZ-1.

CAUTION: Use of more than one digipeater is an extension of the AX.25 protocol and may not function properly with TNCs that do not support this enhancement.

CONOK ON/OFF *default: ON **

parameters

ON Connect requests from other TNCs will be automatically acknowledged.

OFF Connect requests from other TNCs will not be automatically acknowledged.

This command determines the action taken by the TNC when a connect request for it is received through the radio. ON will result in the request being acknowledged, the standard connect message will be sent through the terminal, and the data transfer mode specified by CONMode will be entered.

OFF will cause the message "connect request: <call>" to be sent to the terminal. You may then enter the command mode and enter your own connect command.

Example of OFF:

*** connect request: KC0XYZ

<CTRL-C>

CONNECT KC0XYZ

*** CONNECTED TO KC0XYZ

In this example, the TNC is assumed to be in the converse mode, but not currently connected to anyone. After receipt of the message, you enter the command mode and issue a connect command. If CONOK is ON, only the final connected message will appear.

Connect requests received from another station when the TNC is already connected or when CONOK is OFF cause the TNC to issue a DM packet (busy-signal). The connect request message will not appear if the TNC is in the transparent mode.

CONVERS

CONVERS has no options. It is an immediate command, and will cause exit from the command mode into the converse mode. Any possible link connections are unaffected.

CPACTIME ON/OFF H ON *default: OFF **

parameters

ON PACTIME is used in the converse mode.

OFF PACTIME is not used in the converse mode.

When CPACTIME is ON, the PACTIME parameter is used in the converse mode as well as in the transparent mode. This mode is normally used when a computer is attached to the TNC on the other end of the link but the full transparent mode is not desired. In this mode, characters are sent periodically as in the transparent mode; however, the local editing and echoing features of the converse mode are enabled.

CR should normally be off in this mode since, otherwise, the SENDPAC character is appended at random intervals as the input is packetized by the timer.

CR ON/OFF H OFF *default: ON **

parameters

ON The SENDPAC character, normally RETURN, is appended to all packets sent in the converse mode.

OFF The SENDPAC character is not appended to packets.

When CR is ON, all packets sent in the converse mode will include the SENDPAC character which forces the packet to be sent. Setting CR ON and SENDPAC $0D results in a natural conversation mode. Each line is sent when you enter a RETURN, and arrives at its destination with a RETURN at the end of the line. If AUTOLF is on at the other end, no overprinting occurs.

CWID ON/OFF *default: ON **

parameters

ON The TNC will send an ID after 9.5 minutes if it sent a packet during the previous 9.5-minute interval.

OFF The TNC will send an ID only when the ID command is entered.

If ON is specified, the TNC will ID after a disconnect operation.

IDTEXT will be used for the ID if it has been entered; otherwise, the callsign specified by MYCALL will be used.

DEBUG n *default: $05 <CTRL-E> **

parameters

n 0 – $7F, specifying an ASCII character.

This command is used to change the debug program entry character. When you enter that character in the command or converse mode, the resident debugger is entered. The debugger is described in the "Special Functions" section on Page 7-1 of this Manual.

DELETE ON/OFF *default: ON **

parameters

ON The delete character input editing character is <delete> ($7F).

OFF The delete character input editing character is <back space> ($08).

This command is used to change the input editing command for character deletion.

DIGIPEAT ON/OFF *default: ON **

parameters

ON The TNC will digipeat AX.25 packets if requested.

OFF The TNC will not digipeat AX.25 packets.

When this parameter is turned on, any packet received that has MYCALL in the digipeat list of its address field will be retransmitted. Each station included in the digipeat list relays the packet in the order specified, marking the packet so that it will not accidentally relay it twice (unless so requested), and so the stations will relay the packet in the correct order. Only the digipeating of AX.25 packets is controlled by this value; VADCG digipeating is controlled by VDIGIPEA. Digipeating takes place concurrently with other TNC operations and does not interfere with normal operation of a packet station.

DISCONNE

This command will immediately initiate a disconnect request with the currently connected station. A successful disconnect results in the display of:

```
*** DISCONNECTED
```

You may enter other commands while the disconnect is taking place, although connects are disallowed until the disconnect is completed.

If you exceed the RETRY count while waiting for the other side to acknowledge, the TNC moves to the disconnected state. If you enter a disconnect command while the TNC is disconnecting, the retry count is immediately exceeded. In both cases, the disconnect message is preceded by "retry count exceeded."

Disconnect messages are not displayed when the TNC is in the transparent mode.

DISPLAY [class]

parameters

class — optional parameter-class identifier, one of the following: CHARACTE, ID, LINK, MONITOR, TERMINAL, TIMING.

The display will cause all control parameters and their current values to be displayed. You can display individual parameters by entering the parameter name with no options. Also, you can display groups of related parameters by specifying the optional parameter-class.

DWAIT n — *default: 2 **

parameters

n — 0 – 15, specifying 40 ms intervals.

This value is used to avoid collisions with digipeated packets. The TNC will wait DWAIT * 40 ms after last hearing data on the channel before it begins its own keyup sequence. This value should be agreed on by all members of a local area when digipeaters are used in the area. The best value will be determined by experimentation, but will be a function of the key-up time (TXDELAY) of the digipeater.

This feature is made available to help alleviate the drastic reduction of throughput that occurs on a channel when digipeated packets suffer collisions. It is necessary because digipeated packets are not retried by the digipeater, but must be restarted by the originating station. If all stations specify DWAIT, and the right value of DWAIT is chosen, the digipeater will capture the frequency every time it has data to send, since digipeated packets are sent without this delay.

ECHO ON/OFF — *default: ON **

parameters

ON — Characters received from the terminal are echoed to the terminal by the TNC.

OFF — Characters are not echoed.

ECHO controls local echoing by the TNC when it is in the command or the converse mode. Local echoing is disabled in the transparent mode.

ESCAPE ON/OFF — *default: OFF **

parameters

ON — The ESC (escape) character ($1B) is output as "$" ($24).

OFF — The ESC character is output as ESC ($1B).

This command specifies the character which will be output when an <escape> character is to be sent to the terminal. The <escape> translation is disabled in the transparent mode.

FLOW ON/OFF — *default: ON **

parameters

ON — Input flow control is active.

OFF — Input flow control is disabled.

When FLOW is on, any character entered from the terminal will halt output to the terminal until a packet is forced (in the converse mode) or a line is completed (in the command mode), PACLEN is exceeded, or the terminal output buffer fills up. Cancelling the current command or packet or typing the redisplay-line character will also cause output to resume. FLOW is ignored in the transparent mode.

FLOW will keep received data from interfering with data entry. NOTE: Also see the comments on the CR command.

FRACK n *default: 4 **

parameters

n 0 – 15, specifying one second intervals.

After transmitting a packet requiring acknowledgment, the TNC waits FRACK seconds before incrementing the retry counter and sending it again. If the retry count specified by RETRY is exceeded, the current operation is aborted. If the packet address includes relay requests, the time between retries will be adjusted to:

Retry interval = FRACK * (2*m + 1)

where m is the number of intermediate relay stations.

When the retried packet is sent, a random wait time is added to any other wait times in use. This is to avoid lockups where two TNCs repeatedly collide with each other.

FULLDUP ON/OFF *default: OFF **

parameters

ON Full duplex mode is enabled.

OFF Full duplex mode is disabled.

When FULLDUP is OFF, the TNC makes use of the carrier-detect signal from the modem to avoid collisions, and acknowledges multiple packets with a single acknowledgment. When FULLDUP is ON, the TNC ignores the carrier-detect signal and acknowledges packets individually. The latter mode is useful only for full-duplex radio operation, such as through OSCAR 10.

HBAUD n *default: 8 (1200 baud) **

parameters

n selects a baud rate from the list below.

This command specifies the baud rate used for radio packet communications. This value has **no relationship** to the terminal baud rate selected by ABAUD. In order to communicate with other packet stations, the radio baud rates must be the same. Note that this table does not correspond to the ABAUD rate table because of the addition of the special 400 baud rate.

0 Reserved for future use.

50	600
75	1200
110	1800 *
135	2400 *
150	3600 *
300	4800 *
400	

* — These baud rates are not supported by the on-board Bell-202 compatible modem.

ID

ID will send the CW identification the next time the frequency is clear. If there is an ID request already pending, this command is ignored. You can use this command to send a CW identification even if the automatic ID feature is disabled.

IDTEXT will be used for the ID if you have entered it; otherwise, the callsign specified by MYCALL will be used.

IDTEXT text

parameters

text — any combination of characters and spaces.

IDTEXT specifies the ID sent by the ID commands. A maximum of 128 characters can be specified. If the first character is set to & or %, the call sign set by MYCALL will be sent.

LCOK ON/OFF *default: ON **

parameters

ON — The terminal is capable of receiving lowercase ASCII characters.

OFF — The terminal is not capable of receiving lowercase ASCII characters.

If LCOK is OFF, lowercase characters will be translated to uppercase before being sent to the terminal. This case translation is disabled in the transparent mode. Input characters and echoes are not case translated.

LFADD ON/OFF *default: OFF **

parameters

ON — A line-feed character is added to outgoing packets following each RETURN transmitted in the packet.

OFF — No line-feed is added to outgoing packets.

This function is similar to AUTOLF, except that the line-feed characters are added to outgoing packets rather than to the text displayed locally. This feature is included in order to maintain compatibility with other packet radio controllers. If the person you are talking to reports overprinting of packets from your station, you should set LFADD ON. This character insertion is disabled in the transparent mode.

MALL ON/OFF *default: OFF **

parameters

ON — Monitored packets include both "connected" packets and "unconnected" packets.

OFF — Monitored packets include only "unconnected" packets.

This command determines the class of packets which are monitored. If MALL is OFF, only otherwise eligible packets (as determined by MTO and MFROM commands) sent by other TNCs in the unconnected mode are displayed. This is the normal manner of operation when this TNC is being used to talk to a group of TNCs that are all unconnected.

If MALL is ON, all otherwise eligible frames are displayed, including those sent between two other connected TNCs. You may use this mode for diagnostic purposes or "reading the mail."

MAXFRAME n H 2 *default: 4 **

parameters

n — 1 – 7, signifying a number of packet frames.

MAXFRAME sets an upper limit on the number of unacknowledged frames which the TNC can have outstanding at any one time. This is also the maximum number of adjoining frames which can be sent during any given transmission. If some but not all of the outstanding frames are acknowledged, a smaller number may be transmitted the next time, or new frames may be included in the retransmission, so that the total unacknowledged does not exceed MAXFRAME.

MCON ON/OFF *default: OFF **

parameters

ON — The monitor mode remains active when the TNC is connected.

OFF — The monitor mode is off while the TNC is connected.

If MCON is ON, the TNC will observe the MONITOR command while the TNC is connected to another TNC. If MCON is OFF, the display of monitored packets is suspended when a connect occurs, and is resumed when the TNC is disconnected. If MCON is on and the station connected to is selected by MFROM or MTO, you would see only the packets displayed by the monitor function.

MFROM call1[,call2...,call10] *default: NONE* **

parameters

call Call-sign list. Up to ten calls, separated by commas.

MFROM establishes a list of FROM call-signs to the monitor. If MONITOR is on, any packet heard (which has as its FROM address any of the calls in the MFROM list) will be displayed.

NOTE: There are two special calls. If you use either one, it must be the only call in the list. ALL means display all packets heard regardless of their FROM address. NONE means do not display packets based on the contents in either the MTO list or the MFROM list. This means that if either list is ALL, all packets will be displayed.

** If you specify anything other than NONE, ALL will be stored in NOVRAM by the PERM command.

MONITOR ON/OFF *default: ON* *

parameters

ON Monitor mode is on.

OFF Monitor mode is off.

If the monitor mode is on, and the TNC is not in the transparent mode, packets not addressed to this TNC may be displayed. The addresses in the packets are displayed along with the data portion of the packet. For example:

```
KB8XYZ>W0XYZ-3: I'm ready to transfer the file now.
```

The calls are separated by a ">" and the call sign extension field is displayed if it is other than 0. The MALL, MTO, and MFROM commands determine which packets are to be monitored. The MCON command controls the action of the monitor mode when the TNC is connected. All monitor functions are disabled in the transparent mode. To completely enable the monitor mode, you must specify a TO or FROM list using MTO or MFROM.

MTO call1(,call2...,call10) *default: ALL* **

parameters

call Call sign list. Up to ten calls, separated by commas.

MTO establishes a list of TO call signs to the monitor. If MONITOR is on, any packet heard will be displayed if it has any of the calls in the MTO list as its TO address. Refer to the discussion under MFROM for special calls and a discussion of the interaction of the MTO command with other monitor mode commands.

** If anything other than NON is specified, ALL will be stored in NOVRAM by the PERM command.

MYCALL call(-n) *no default* *

parameters

call Call sign assigned to this TNC.

n 0 – 15, optionally specified call sign extension.

This command tells the TNC what its call sign is. This call sign will be placed in the FROM address field for all packets originated by it, and it will respond to frames with this call sign in the TO or digipeat fields as appropriate. MYCALL will be used for Morse code ID unless another string is specified by IDTEXT.

The call sign in the default parameter list is blank, and must be changed for proper operation of the protocols. The default for the extension is 0, and is not required to be changed.

MYVADR n *default: 31* *

parameters

n 0 – 31, signifying a VADCG address byte.

This command selects the address used when the unit is operating in the VADCG mode. The address is translated to the "to be digipeated" range by the VRPT command. Choose VADCG addresses by coordinating with other packet operators in your area to avoid duplicate addresses.

NUCR ON/OFF *default: OFF **

parameters

ON Nulls are sent to the terminal following <cr> characters.

OFF Nulls are not sent to the terminal following <cr> characters.

This command enables a transmission delay following any <cr> sent to the terminal. The length of the delay is determined by the command NULLS. This delay is required by some hardcopy terminals.

NULF ON/OFF *default: ON **

parameters

ON Nulls are sent to the terminal following the line-feed characters.

OFF Nulls are not sent to the terminal following the line-feed characters.

This command enables a transmission delay following any line feed sent to the terminal. The length of the delay is determined by the command NULLS. This delay is required by some display terminals.

NULLS n *default: 0**

parameters

n 0 – 30, the (even) number of nulls to send after <cr> or <lf>.

This command specifies the number of nulls to send to the terminal after a <cr> or <lf> is sent. In addition to setting this parameter value, you must set NUCR and NULF to indicate whether nulls are to be sent after <cr>, <lf>, or both. Devices requiring nulls after <cr> are typically hard-copy devices requiring time for "carriage movement." Devices requiring nulls after <lf> are typically CRTs which scroll slowly. NULLS is valid only in the converse and command modes. If you specify an odd number, it will be rounded down one.

PACLEN n *default: 128 **

parameter

n 1 – 256, the maximum length of the data portion of a packet.

The TNC will automatically transmit a packet when the number of bytes input for a packet reaches PACLEN. This value is used in both the converse and transparent modes.

WARNING: Allowing more than 128 characters of data is an extension of both the AX.25 and VADCG protocols and may not function properly when used to communicate with TNCs that do not support this enhancement.

PACTIME (EVERY/AFTER) n *default: AFTER 4 **

parameters

n 0 – 15, specifies 1/4-second intervals.

EVERY Time-out occurs every n seconds.

AFTER Time-out occurs after n seconds with no other input.

This parameter is always used in the transparent mode, and will also be used in the converse mode if CPACTIME ON is specified. When EVERY is specified, input bytes are packaged and queued for transmission every n/4 seconds. When AFTER is specified, bytes are packaged when input from the terminal stops for n seconds. In neither case is a zero-length packet produced, and the timer is not started until a new byte is entered. If EVERY or AFTER is not given, the current state is retained.

PARITY n *default: 3 (space parity)* *

parameters

n 0 – 4, selecting a parity option from the table below.

This command sets the parity mode for the terminal output according to the following table:

n	Parity
0	odd
1	even
2	mark
3	space
4	none

If PARITY choices 0-3 are chosen together with AWLEN=8, only one stop bit is available and ABIT will be automatically set to 1. The parity bit is automatically stripped on input and not checked. If your terminal transmits seven data bits and a parity bit, all eight bits can be transmitted in the transparent mode if you set AWLEN 8 and PARITY 4.

PASS n *default: $16 <CTRL-V>* *

parameter

n 0 – $7F, specifying an ASCII character.

This command selects the ASCII character used for the pass input editing command.

PERM

PERM is an immediate command. It causes any NOVRAM values changed since the last PERM command to be made permanent; all values are burned into the NOVRAM. Since you can not undo this process by turning the TNC off, be sure you have selected the correct values.

You can reverse each bit of the NOVRAM chip by the burning procedure a minimum of 10,000 times. According to our information, giving the PERM command when no values have changed does not affect the life expectancy of the chip.

PROGRAM

PROGRAM is an immediate command. It enters the EPROM Programmer routine. This routine supports the TAPR EPROM Programmer attachment, which allows programming of EPROMS through the parallel I/O port. This routine is documented with the EPROM Programmer kit. Should you accidently enter this routine, type <RETURN>until you see the Option: prompt. Exit by typing **R** followed by a <RETURN>. Then, following the next prompt, enter any character.

REDISPLA n *default: $12 <CTRL-R>* *

parameters

n 0 – $7F, specifying an ASCII character.

This command is used to change the redisplay-line input editing character.

RESET

This command is used to perform a soft reset. Any NOVRAM parameters changed but not made permanent are retained if switch SW-1 is OFF. Values made permanent are restored via a hardware reset (toggling switch SW-3) or power off/on.

RETRY n *default: 10* *

parameter

n 0 – 15, specifying number of packet retries.

The AX.25 and VADCG protocols allow for retries, i.e., retransmission of frames that are not acknowledged. Frames are retransmitted RETRY times before the operation is aborted. The time between retries is specified by FRACK. A value of 0 specifies an infinite number of retries. If the number of retries is exceeded, the TNC goes to the disconnected state (with an informative message if not in the transparent mode). Also see the FRACK command.

SCREENL n *default: 80 **

parameters

n 0 – 255, specifying the screen or platen width of the terminal.

This value is used to properly format the terminal output. A <cr> <lf> sequence is sent to the terminal at the end of a line in the command and converse modes when n characters have been printed. A value of 0 inhibits this action.

SENDPAC n $01 ↑A *default: $0D RETURN **

Selects the character that will force a packet to be sent when entered in the converse mode.

START n *default: $11 <CTRL-Q> **

parameters

n 0 – $7F, specifying an ASCII character.

Selects the character used to restart the output **from** the TNC **to** the terminal. The output is stopped with the STOP character.

STOP n *default: $13 <CTRL-S> **

parameters

n 0 – $7F, specifying an ASCII character.

Selects the character used to stop the output **from** the TNC **to** the terminal. The output is restarted with the START character.

TRACE n *default: $1000*

parameters

n a 16-bit value.

TRACE is used to set protocol debugging functions. Each bit enables a different option; the bit values are described here. If the bit in the position indicated below is set, the option is enabled.

2xxx	Dump data as well as header.
1xxx	Dump input and output frames in case of FRMR condition.
x8xx	Dump all outgoing frames.
x4xx	Dump incoming frames that look useful.
x2xx	Show "before" and "after" states when the link state changes.
x1xx	Dump all incoming frames.
xx8x	Never send Final bit.
xx4x	Dump any digipeated frames that don't get monitored.
xx2x	Allow digipeated frames to get monitored.

All other bits are currently undefined.

TRANS

This command causes immediate exit from the command mode to the transparent mode. The current link state is not affected.

TXDELAY n *default: 4 **

parameters

n 1 – 16, specifying 40 ms intervals.

This value tells the TNC how long to wait after keying up the transmitter before sending data. Some start-up time is required by all transmitters to put a signal on the air; some need more, some need less. In general, crystal-controlled rigs with diode antenna switching do not need much time, synthesized rigs need time for PLL lockup, and rigs with mechanical T/R relays will need time for physical movement. Determine the correct value for a particular rig by experimentation. The proper setting of this value may also be effected by the requirements of the station you are communicating with. This parameter should be locally agreed upon.

TXFLOW ON/OFF *default: OFF **

parameters

ON The XFLOW parameter is used in the transparent mode.

OFF The XFLOW parameter is ignored in the transparent mode.

When ON, XFLOW is used to determine the type of flow control used in the transparent mode. When OFF, software flow control is not used, i.e., XFLOW is treated as OFF. If TXFLOW and XFLOW are both ON, the TNC will use the XON and XOFF characters to control input from the terminal. However, only hardware flow control is available to the terminal to control output from the TNC, and all input from it remains fully transparent.

UNPROTO call1 [VIA call2[,call3...,call9]] *default: CQ*

parameters

call1 Call sign to be placed in the TO address field.

call2 Optional digipeater call.

This command is used to set the RPT and TO address fields of packets sent in the unconnected (unprotocol) mode. Unconnected packets are sent as unsequenced I frames with TO and RPT fields taken from UNPROTO **call1** through **call9** options. A special call, NONE, is interpreted as "no one in particular", i.e., CQ. When NONE is specified, unconnected packets are sent to CQ. These packets sent from other TNCs can be monitored by setting MONITOR ON and MTO CQ or MTO ALL. As in the case of the CONNECT command, up to eight digipeater calls may be specified. Also, see the BEACON command.

VDIGIPEA ON/OFF *default: OFF **

parameters

ON This TNC is a VADCG digipeater.

OFF This TNC is not a VADCG digipeater.

When ON, VDIGIPEA causes this TNC to retransmit any VADCG frames it receives that have an address byte value in the digipeat range. Digipeating occurs concurrently with other operations of the TNC. Only one TNC in an area should be a VADCG repeater.

VRPT ON/OFF *default: OFF* *

parameters

ON VADCG address is translated into the digipeat range.

OFF VADCG address is not modified.

This parameter is used to request that packets originated by this TNC in the VADCG mode be digipeated. It has no effect on operation in the AX.25 mode.

XFLOW ON/OFF *default: ON* *

parameters

ON XON/XOFF flow control enabled.

OFF XON/XOFF flow control disabled; hardware flow control is enabled.

When XFLOW is on, the device connected to the terminal port is assumed to respond to flow control characters XON and XOFF. When XFLOW is off, the TNC will only respond to hardware flow control (CTS) and will communicate flow control commands via RTS.

XMITOK ON/OFF *default: ON* *

parameters

ON Transmit functions are enabled.

OFF Transmit functions are disabled.

When XMITOK is off, transmitting is inhibited. All other functions of the unit remain the same.

XOFF n *default: $13 <CTRL-S>* *

parameters

n 0 to $7F, specifying an ASCII character.

This command selects the character sent **by** the TNC **to** the terminal to stop input from that device.

XON n *default: $11 <CTRL-Q>* *

parameters

n 0 to $7F, specifying an ASCII character.

This command selects the character sent **by** the TNC **to** the terminal to restart input from that device.

MESSAGES

The first message you should see on your screen after issuing a RESET command or performing a hardware reset is a sign-on message.

```
Heath Company Amateur Packet Radio
AX.25 level 2 version X.Y
```

This message appears when you have performed a reset with DIP switch SW-1 set on. It indicates that the TNC has been initialized with the default parameters stored in ROM. X.Y refers to the software version number.

```
Heath packet radio
```

This sign-on message appears when you have performed a reset with DIP switch SW-1 set off. It indicates that the TNC has been initialized with the parameter values stored in NOVRAM.

```
RAM size is nnnn
```

This message appears after the sign-on message and indicates that the RAM has been successfully verified and found to be the indicated (hexadecimal number) length in bytes. If you have an 8K RAM installed in the U7 socket, the RAM size should be 2000.

```
High RAM size is nnnn
```

This message is printed if you have a RAM chip installed in the U8 socket.

```
HDLC can't init
```

This is a hardware diagnostic that will appear at the time of reset if integrated circuit U17 (HDLC) can not be commanded. It indicates difficulties with either U17 (WD-1935) or U6 (6522).

```
PIA can't init
```

This is a hardware diagnostic that will appear at the time of reset if PIA U13 (6520) cannot be commanded.

```
UART can't init
```

This message will never appear — if the UART cannot initialize, there is nothing to type on. If you see LEDs D1 and D2 blinking every couple of seconds when the TNC should be signing on, it may indicate problems with UART U14 (6551).

Messages displayed in response to the command mode input are discussed below. Messages, as they appear on terminals with lowercase displays, are given at the left margin. Messages which are responses to input errors in commands from the user will try to point out the problem by typing a $ under that part of the line. If your input line is messy because of deleted characters echoed as \ or incoming packets, the pointer will not line up properly.

```
cmd:
```

This is the command mode's prompt for input. Any characters you enter after the TNC prints "cmd:" will be used as command input and not packet data.

```
EH?
```

This is the TNC's generalized "I don't understand" message. A dollar sign ($) is used to point to the offending character. It will also appear if a required input item is missing; for example:

```
C WD8XYZ VIA
            $
EH?
```

In this example, the required call sign after the VIA option is missing. Most commands that receive an EH? error are ignored. In a few cases, part of the command may be accepted and acted upon, as described under the message "Input ignored".

Value out of range

If the syntax of the command is legal but the value specified is too large or too small for this command, the "value out of range" message is used. A $ is used to point to the bad value.

Input ignored

Since the command processor was kept small and simple, it will sometimes change parameters before it completes some of the more involved commands. In some cases, options at the beginning of the command will have been acted on before a syntax error near the end of the line is reached. When this occurs, "Input ignored" is used to show what part of the line was ignored. The dollar sign points to the boundary. Characters to the left were used; the character pointed to and those to the right were not; that is, the line was processed as if a <cr> was entered at the $. Example:

```
MTO QST,WB9XYZ K9XYZ
              $
Input ignored
```

The command is processed as if it were MTO QST, WB9XYZ and the K9XYZ is ignored.

was

Whenever one of the NOVRAM values is changed, the previous value is displayed. Example:

```
AX25 OFF
was ON
```

Not while connected

The AX25 parameter can not be changed if the TNC is connected to another TNC. This message is printed if such an attempt is made.

Too many packets pending

You have given a CONVERS or TRANS command after exiting the transparent mode but before all packets have been acknowledged. Wait until the TNC quits retrying packets; then enter the command again.

Link state is:

This message is output in response to the CONNECT and DISCONNECT commands if the state of the link does not permit the requested action. It is prefaced by either "Can't CONNECT" or "Can't DISCONNECT" as appropriate.

A CONNECT command with no options will display the current link state. The states are:

DISCONNECTED

No connection exists. Connects are legal; disconnects are not.

CONNECT in progress

Connect request has been issued. Another connect is illegal; a disconnect will abort the attempt.

CONNECTED to <CALLSIGN>

The TNC is connected. Connects are illegal, disconnects start the disconnect process.

FRMR in progress

The TNC is connected but a protocol error exists. This should never happen when two TNCs are connected. An improper implementation of the AX.25 protocol could cause this state to be entered. The TNC will attempt to resynchronize frame numbers with the TNC on the other end although a disconnect may result. Connects are not legal in this state, and a disconnect will start the disconnect process.

DISCONNECT in progress

A disconnect has already been issued. Connects are not legal in this state, and a second disconnect will cause a "retry count exceeded" condition.

YOUR HEATHKIT 90-DAY FULL WARRANTY

During your first ninety (90) days of ownership, Heath Company will replace or repair free of charge — as soon as practical — any parts which are defective, either in materials or workmanship. You can obtain parts directly from Heath Company by writing us or telephoning us at (616) 982-3571. And we'll pay shipping charges to get those parts to you — anywhere in the world.

We warrant that, during the first ninety (90) days of ownership, our products, when correctly assembled, calibrated, adjusted, and used in accordance with our printed instructions, will meet published specifications.

If a defective part or error in design has caused your Heathkit product to malfunction during the warranty period, through no fault of yours, we will service it free upon delivery at your expense to the Heath factory, Benton Harbor, Michigan, or to any Heathkit Electronic Center (units of Schlumberger Products Corporation), or through any of our authorized overseas distributors.

You will receive free consultation on any problem you might encounter in the assembly or use of your Heathkit product. Just drop us a line or give us a call. Sorry, we cannot accept collect calls.

Our warranty, both expressed and implied, does not cover damage caused by use of corrosive solder, defective tools, incorrect assembly, misuse, fire, customer-made modifications, flood or acts of God, nor does it include reimbursement for customer assembly or setup time. The warranty covers only Heath products and is not extended to non-Heath allied equipment or components used in conjunction with our products or uses of our products for purposes other than as advertised.

And if you are dissatisfied with our service — warranty or otherwise — or our products, write directly to our Director of Customer Services, Heath Company, Benton Harbor, Michigan, 49022 He'll make certain your problems receive immediate, personal attention.

HEATH COMPANY
BENTON HARBOR, MI. 49022

Prices and specifications subject to change without notice.

instructions

FOR THE

HEATH
Schlumberger

HIGH VOLTAGE PROBE
(X100 FOR 11MΩ INPUTS)

MODEL IMA-100-11

HEATH

MODEL	SERIES NO.
IMA-100-11	00628

597-1424

HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022
a Schlumberger company

PRINTED IN U.S.A.

PROTOCOLS

AN OVERVIEW

EXPLANATION OF PROTOCOL

This material is intended to give an overview of the protocols used to transmit data by the Heath software. References are given to more detailed information required by those wishing to implement these protocols on other hardware. The material presented below is somewhat tutorial in nature for those who have not had previous exposure to layered network protocols, but it presumes some knowledge of general communications hardware and software. If you are already well versed in networking, you may wish to skip this material and refer to "AX.25 Protocol Specification" and "VADCG Protocol Specification."

The Heath TNC hardware and software architecture is organized in accordance with the International Standards Organization (ISO) layered network model. The model describes seven levels and is officially known as the ISO Reference Model of Open Systems Interconnection, or simply the ISO Model. The model and many other interesting topics are discussed in "Computer Networks" by Andrew S. Tanenbaum.

The ISO model provides for layered processes, each supplying a set of services to a higher level process. The Heath TNC currently implements the first two layers, the Physical Layer and the Data Link layer.

Physical Layer

The duty of the physical layer, layer one, is to provide for the transmission and reception of data at the bit level. It is concerned only with how each bit is physically transmitted; that is, voltages on a hardwire line or modem tones on phone or RF links.

The physical layer of the Heath TNC is compatible with the VADCG TNC. The actual modem interface is compatible with the Bell 202 standard, which is similar to the CCITT (International Telegraph and Telephone Consultation Committee) V.23 standard. Any other hardware device compatible with the Bell 202 standard will be compatible with the Heath TNC, at least at level one of the ISO reference model.

Data Link Layer

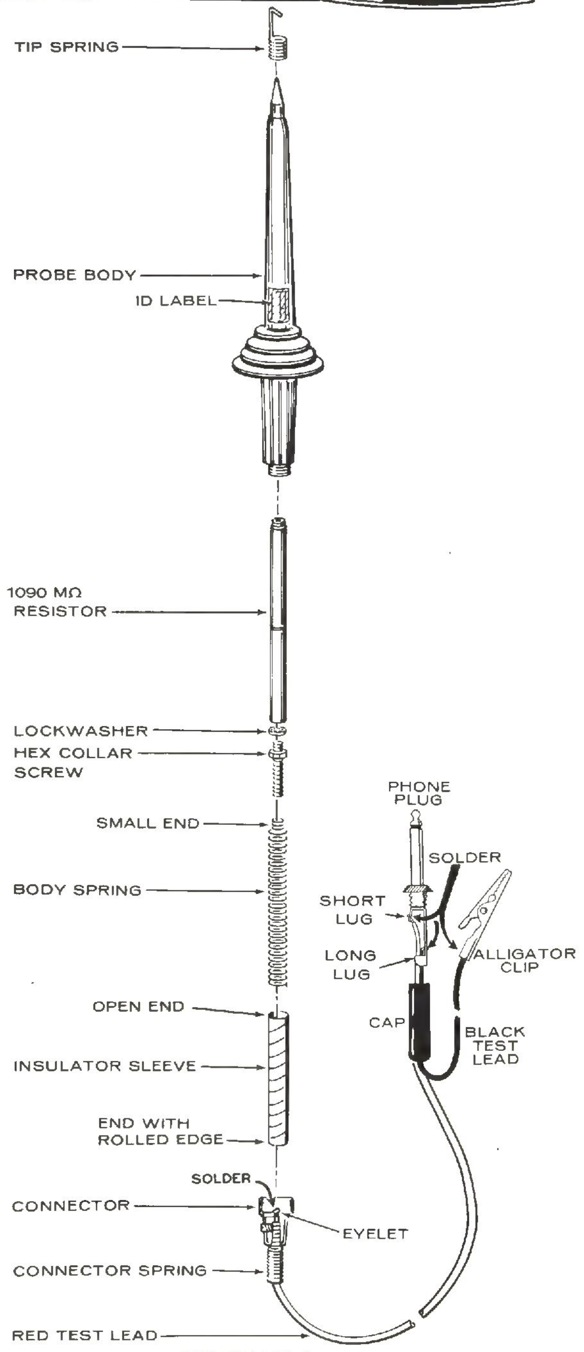
The duty of the data link layer is to supply an error-free stream of data to higher levels. Since level one simply passes any bits received to level two and is unaware of the content or overlying structure of the data, transmission errors are not detectable at level one. Level two carries the responsibility of detecting and rejecting bad data, retransmitting rejected data, and detecting the reception of duplicate data.

PARTS LIST

CAUTION: Do not unpack or handle the 1090 MΩ resistor until you are instructed to do so in a step. Moisture from your fingers can change its resistance, and thus affect its accuracy.

PART No.	QTY	DESCRIPTION	PRICE Each
432-1	1	Connector	.70
476-2	1	Probe body	3.25
2-47	1	1090 megohm resistor	5.00
250-6	1	Hex collar screw	.15
260-1	1	Alligator clip	.10
258-2	1	Tip spring	.15
258-3	1	Body spring	.30
70-1	1	Insulator sleeve	.15
438-3	1	Phone plug	.65
341-1	1	Length black test lead	.10
341-2	1	Length red test lead	.10
390-1176	1	ID label	
391-34	1	Blue and white label	
597-260	1	Parts Order Form	
	1	Instruction sheet (see Page 1 for part number.)	
		Solder (Additional 3' rolls of solder, #331-6, can be ordered for 15 cents each.)	

The above prices apply only on purchases from the Heath Company where shipment is to a U.S.A. destination. Add 10% (minimum 25 cents) to the price when you order from a Heathkit Electronic Center to cover local sales tax, postage and handling. Outside the U.S.A., parts and service are available from your local Heathkit source and will reflect additional transportation, taxes, duties and rates of exchange. Prices and specifications subject to change without notice.



PICTORIAL 1

ASSEMBLY INSTRUCTIONS

NOTE: In the next step, do not handle the resistor with your bare hands. Finger prints can change the resistance enough to cause inaccurate readings. Use a soft dry cloth between your hands and the resistor.

( ) Locate the 1090 megohm resistor. Remove and discard the screw at one end of the resistor, but keep the lockwasher.

( ) Place the lockwasher on the short end of the hex collar screw; then thread this end of the screw into the resistor as shown. Now screw the small end of the body spring onto the long end of the hex collar screw.

( ) Slip the body spring into the open end of the insulator sleeve.

( ) Insert the resistor and body spring and insulator sleeve assembly into the probe body with the resistor toward the tip of the probe. See Pictorial 1.

( ) Remove 1/4" of insulation from one end of the red test lead and 1/2" of insulation from one end of the black test lead.

( ) Twist the fine bare wires at the end of each lead. Then melt a small amount of solder to the ends to hold the small strands together.

( ) Push the prepared end of each lead through the phone plug cap. Then solder the black lead to the long phone plug lug and the red lead to the short phone plug lug.

( ) After the wires have completely cooled, use pliers to bend the tabs on the phone plug over lightly to secure the two wires. Be sure you do not cut through the insulation by pinching the wires too hard with the tabs. Screw the cap onto the phone plug.

( ) Remove 1/2" of insulation from the other end of the black test lead and twist the fine wires together. Then melt a small amount of solder to the end to hold the small strands together.

( ) Solder the prepared end of the black test lead to the alligator clip.

( ) Remove 3/8" of insulation from the other end of the red test lead and twist the fine wires together. Then melt a small amount of solder to the end to hold the small strands together.

( ) Insert this end through the connector spring and solder it to the eyelet in the connector as shown. Cut off the excess lead lengths from the soldered connection.

( ) Screw the connector and test lead assembly to the probe body. This compresses the body spring and insures the proper contact between the resistor and tip, and between the body spring and test lead assembly.

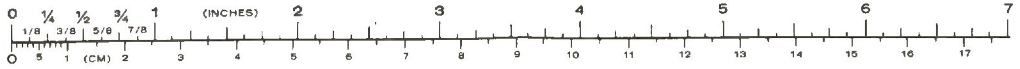
( ) Install the tip spring by pressing the spiral end of the spring onto the tip of the probe.

( ) Carefully peel away the paper backing from the ID label. Then press the label onto the probe body as shown.

NOTE: The blue and white label that you will install in the following step shows the Model number and Production Series number of your kit. Refer to these numbers in any communications you have with the Heath Company about this kit.

( ) Carefully peel away the paper backing from the blue and white label. Then press the label onto the first page of these instructions.

This completes the assembly. Read the next section, "Using the High Voltage Probe," before you attempt to use the probe. Then connect the probe to your DC voltmeter in place of the regular DC test probe.



USING THE HIGH VOLTAGE PROBE

CAUTION: HIGH VOLTAGES ARE EXTREMELY DANGEROUS. NEVER MEASURE DC VOLTAGES IN EXCESS OF 30,000 VOLTS.

This probe allows you to make high voltage measurements as safely as possible. ALWAYS CONNECT THE DC VOLTMETER AND THE PROBE IN THE FOLLOWING ORDER TO AVOID SHOCK HAZARD:

Connect the phone plug of the probe to the voltmeter in place of the regular test probe. Do not connect the probe tip to the circuit under test before you connect the phone plug to the voltmeter and connect the ground lead to the chassis of the unit under test.*

Wherever possible, contact the high voltage by hooking the tip spring to the terminal under test. This should be done with the power turned off. Then without touching the probe, turn power on, take the reading, turn the power off, carefully discharge any high voltage capacitors which may be in the circuit, and remove the probe from the circuit.

*The conductors inside the handle and the test lead assembly never carry more than 300 volts when the probe is properly connected. These parts will be exposed to the full 30,000 volts if the phone plug of the probe is not connected to the voltmeter and if the ground lead is not connected to the ground point of the circuit under test.

When this Probe is connected to an 11 megohm DC voltmeter, the voltage ranges and the input resistance are increased by a factor of 100. Thus the 10-, 150-, and 300-volt ranges become 1000-, 15,000-, and 30,000-volt ranges respectively. The input resistance will increase from 11 megohms to 1100 megohms. This allows you to make measurements in high resistance circuits with negligible loading.

NOTE: Although multiplying a 500-volt range by 100 gives a range of 50,000 volts, never use the probe on DC voltages above 30,000 volts.

PROTOCOLS

AN OVERVIEW

EXPLANATION OF PROTOCOL

This material is intended to give an overview of the protocols used to transmit data by the Heath software. References are given to more detailed information required by those wishing to implement these protocols on other hardware. The material presented below is somewhat tutorial in nature for those who have not had previous exposure to layered network protocols, but it presumes some knowledge of general communications hardware and software. If you are already well versed in networking, you may wish to skip this material and refer to "AX.25 Protocol Specification" and "VADCG Protocol Specification."

The Heath TNC hardware and software architecture is organized in accordance with the International Standards Organization (ISO) layered network model. The model describes seven levels and is officially known as the ISO Reference Model of Open Systems Interconnection, or simply the ISO Model. The model and many other interesting topics are discussed in "Computer Networks" by Andrew S. Tanenbaum.

The ISO model provides for layered processes, each supplying a set of services to a higher level process. The Heath TNC currently implements the first two layers, the Physical Layer and the Data Link layer.

Physical Layer

The duty of the physical layer, layer one, is to provide for the transmission and reception of data at the bit level. It is concerned only with how each bit is physically transmitted; that is, voltages on a hardwire line or modem tones on phone or RF links.

The physical layer of the Heath TNC is compatible with the VADCG TNC. The actual modem interface is compatible with the Bell 202 standard, which is similar to the CCITT (International Telegraph and Telephone Consultation Committee) V.23 standard. Any other hardware device compatible with the Bell 202 standard will be compatible with the Heath TNC, at least at level one of the ISO reference model.

Data Link Layer

The duty of the data link layer is to supply an error-free stream of data to higher levels. Since level one simply passes any bits received to level two and is unaware of the content or overlying structure of the data, transmission errors are not detectable at level one. Level two carries the responsibility of detecting and rejecting bad data, retransmitting rejected data, and detecting the reception of duplicate data.

Level two accomplishes this task by partitioning data to be transferred by level one into individual frames, each with its own error detection field and frame identification fields. The Heath TNC supports two level-two layers, the VADCG and AX.25 protocols. Each of these protocols is based on HDLC, the High Level Data Link Control protocol defined by the ISO.

HDLC FRAMES

Exact knowledge of the format of HDLC frames has been made largely unnecessary by the advent of LSI and VLSI communications chips which interface directly with the level one hardware. The level two software need only supply data to fill in various fields and the chip takes care of the rest. For completeness however, an HDLC frame looks like this:

| FLAG | ADDRESS | CONTROL | DATA | FCS | FLAG |

FLAG — A unique bit sequence used to detect frame boundaries. A technique called "bit stuffing" is used to keep data from looking like a flag.

ADDRESS — A field normally specifying the destination address. The VADCG protocol uses a one-byte destination address; AX.25 uses 14 or 21 bytes containing the actual call signs of the source, destination, and optionally a digipeater.

CONTROL — A byte which identifies the frame type. In both the VADCG and AX.25 protocols, the control field may include frame numbers in one or two 3-bit fields.

DATA — This field contains the actual information to be transferred. This field need not be present. Most frames used for link control only do not have data fields.

FCS — Frame Check Sequence, a 16-bit error detection field.

The communications chip recognizes the opening and closing flags and passes the address, control, and data fields to the software. The FCS field is a Frame Check Sequence computed by the transmitting chip and sent with the frame. The receiving chip recomputes the FCS based on the data received and rejects any frames where the received FCS does not match the computed FCS. This satisfies the level two task of bad data detection.

The communications chip used on the Heath TNC is a Western Digital 1935 running in the NRZI mode. The NRZI mode is a way of encoding bits so that a logic level transition is guaranteed to occur at least every 5-bit times. This allows two chips to synchronize clocks when data is transmitted, and is required when bit stream data is sent asynchronously, as is done by the Heath TNC. The Intel 8273, which is used on the VADCG TNC, and the Zilog 8530 are also compatible with the 1935.

The HDLC format supplied by the communications chip is common between the VADCG and AX.25 protocols. There are several other layer two concerns that are not handled by the chip. These items are duplicate frame detection, connection and disconnection of the level two layers on different TNCs, and buffer overrun avoidance. AX.25 and VADCG solve these problems in similar ways. The AX.25 protocol will be discussed first, and then the areas in which VADCG differs will be presented.

AX.25 LEVEL TWO

AX.25 is based on the Balanced Link Access Procedure (LAPB) of the CCITT X.25 standard. LAPB in turn conforms to the HDLC standard. Two extensions are made to LAPB in AX.25, the extended address field and the unnumbered information (UI) frame. In LAPB, addresses are limited to eight bits, while AX.25 uses either 112 or 168 bits, containing the originator's call sign, the destination call sign, and an optional digipeater (simplex digital repeater) call sign.

The UI frame is used to send information bypassing the normal flow control and acknowledgment protocol. This data is not acknowledgable; however, it can be transmitted by layer two at any time without fear of disturbing higher layers. It is used by the Heath TNC for beacon frames and for sending information frames when the TNC is not connected to another TNC; that is, CQ and QST activities.

The exact specifications for AX.25 are supplied in "AX.25 Protocol Specification." The Heath implementation makes three deviations from that specification. These deviations are detailed below.

DM Frame — This frame is sent whenever a non-SABM frame is received when the TNC is in the disconnected state. Heath has expanded this definition. A Heath TNC will send a DM frame in the following additional cases:

1. When an SABM frame is received and the TNC is in the disconnect state and the CONOK flag is off.
2. When the TNC is connected and an SABM frame is received from a third TNC. The DM is sent to the third user.

If a DM frame is received by a TNC in response to an SABM frame sent by it, that TNC will print:

```
*** <call> busy
```

Address Field — This field has a single digipeater call sign slot specified. Heath has extended the address field to allow up to eight digipeater call signs. Only as many digipeater subfields as needed are sent. Only the final byte of the final digipeater subfield has its "E" bit set. The meaning of the "H" bit is extended from "This frame has been been repeated" to "The frame has been repeated by this digipeater." Thus, when a frame is received, the digipeater list is scanned beginning with the subfield closest to the start of the frame, looking for the first digipeater address with H set to 0. If that subfield is the current TNC, the frame is repeated, first setting the H bit in the subfield to 1. If all digipeater "H" bits are on, then the frame has been completely repeated and the destination address can be searched. The destination TNC will reverse the digipeater list when packets are sent in the other direction.

If the VIA option of the CONNECT command is limited to one call by the user, the Heath TNC will generate address fields in compliance with the current specification.

Poll/Final Bit — The handling of this bit is an area of controversy. X.25, from which AX.25 was taken, defines the uses of of the P/F bit. The AX.25 environment is not quite the same, and the P/F bit becomes harder to pin down. In fact, some amateurs have pointed out the need for a second bit in addition to the P/F bit to make error recovery work in all cases. The Heath TNC code does not use the P/F bit to perform its error recovery. Instead, all packet retries are based on the timers, RR, and REJ frames already defined by AX.25. The Heath TNC will never generate a frame with the P/F bit set. For compatibility with other software which may be using the P/F bit as specified, the Heath TNC will generate an RR frame with the final bit set in response to a frame with the poll bit set.

NOTE: All of the above items are invisible to the Heath TNC user and are mentioned only for the benefit of those who may be writing software.

The following paragraphs list the frame types used by AX.25 and describe their purpose. The material is intended only for those who wish to have a general idea on what takes place on an AX.25 link. If you wish to implement this material, refer to "AX.25 Protocol Specification." The control field contents are given as they appear in memory after data is received; that is, the high order data bit is at the left and the low order bit is at the right. Some texts choose to list the bits in the order in which they are transmitted, which is low order bit first. The "AX.25 Protocol Specification" reproduced later in this section uses this format. The data is presented in the "as in memory" form here because that is how it appears in the trace dump format enabled by the TRACE command.

The control bytes listed below are presented in hexadecimal form with the "x" character used to signify four bits which may be any value, depending on what acknowledge functions the packet is performing. Usually "x" is a frame number. Frame numbers fit in three bits and are used to ensure that frames are received in order and that no frames are missed. Since only three bits are available, the frame number is counted modulo 8. This is why the MAXFRAME parameter has a ceiling of seven; that is, no more than seven frames can be "in flight" (transmitted but unacknowledged) at one time. A short description of the use of the frames is given after the table.

x1	RR	— Receive Ready
x5	RNR	— Receive Not Ready
x9	REJ	— Reject
03	UI	— Unnumbered Information
0F	DM	— Disconnected Mode
2F	SABM	— Connect request
43	DISC	— Disconnect request
63	UA	— Unnumbered Acknowledge
87	FRMR	— Frame reject
even	I	— Any frame ending in an even number (including A, C, and E) is an information frame.

I — This and the UI frames are the only frame types containing user data. The control byte contains this frame's number and the number of the next frame expected to be received from the other end of the link.

RR — Usually used to acknowledge receipt of an I frame. The RR function can also be performed if an I frame is sent with an updated "expected next frame number" field.

RNR — Used when the buffer space on the receiving side is full.

REJ — Used to request retransmission of frames starting from "x." Missed frames are detected by receiving a frame number larger than that expected.

DM — Sent in response to any frame received other than a connect request (SABM) when the TNC is disconnected. Sent in response to an SABM whenever the TNC is on the air but cannot connect to the requesting user; for example, if the TNC is already connected to someone else or if CONOK is OFF.

SABM — Set Asynchronous Balanced Mode – initiates a connect.

DISC — Initiates a disconnect.

UA — Sent to acknowledge receipt of an SABM or DISC.

FRMR — Sent when an abnormal condition develops; that is, the protocol byte received is undefined or not proper protocol at the time received.

UI — An I frame without a frame number. It is not acknowledged.

VADCG LEVEL TWO

The VADCG level two protocol is also based on HDLC and is therefore similar to AX.25. VADCG is not based on LAPB however, so many procedures are different, most notably in the connect and disconnect sequences. VADCG does not define a REJ frame type. VADCG frame types and control bytes are listed below in the same format as for AX.25 above. The detailed VADCG description appears in "VADCG Protocol Specification."

x1	RR	— Receive Ready
x5	RNR	— Receive Not Ready
03	UI	— Unnumbered Information
17		— Connect request
07		— Connect Acknowledge
53		— Disconnect Request
43		— Disconnect Acknowledge
even	I	— Any frame ending in an even number (including A, C, and E) is an information frame.

Frame use is as in the AX.25 protocol except that connect and disconnect request frames have the P/F (bit 4) set, and are acknowledged by the same control byte with the P/F bit turned off.

CHANNEL AND TIMING FUNCTIONS

The following discussions mention parameters which are set by various commands. The time values selected are discussed in the "Operation" section.

An important part of any packet radio control is the means by which many stations make efficient use of an RF channel, achieving maximum throughput with minimum interference. The basis for this time domain multiplexing is CSMA (Carrier-Sensed Multiple Access) with collision detection and collision avoidance.

CSMA means simply that no station will transmit if the frequency is in use. The TNC continually monitors for the presence of an audio data carrier on the frequency and transmits only if there is no carrier. (Note that the RF carrier is not detected.) In order to make detection of a busy channel more reliable, the TNC sends an audio signal (continuous flags) any time the transmitter is keyed up and a packet is not being sent, as during the transmitter key-up delay (TXDELAY), or while a slow audio repeater is being keyed (AXDELAY).

By itself, CSMA is not enough to insure a minimum (or even low) interference rate due to the likelihood or simultaneous keyup by two or more stations. This is where collision detection and collision avoidance comes in. The TNC detects a collision by the absence of an ACK signal from the station it is sending to. The receiving station does not acknowledge the frame that suffered the collision, since either the FCS was incorrect or the packet was not heard. There are other possible reasons for nonreceipt of the packet, but the TNC's response is based on the assumption of a collision.

After transmitting a packet, the TNC waits a "reasonable" length of time (frame acknowledge — FRACK) for an acknowledgment. "Reasonable" is determined by the link activity, packet length, whether the packet is being digipeated, and other time-related factors.

If no ACK signal is received, the packet must be resent. If the unacknowledged frame was lost because of a collision, the presumption is that there is at least one other packet station out there that also lost a frame and will probably have exactly the same criterion for deciding when to retry the transmission as this station is using.

In order to avoid a second collision, the collision avoidance protocol calls for the stations retrying transmissions to wait a random time interval after hearing the frequency become clear before they key their transmitters. There must be enough different random wait times to provide a reasonable chance of two or more stations selecting different values. In addition, the difference between adjacent time values must be similar to the key-up time delay of typical stations on the frequency. This is the time lapse after a station keys its transmitter before other stations detect its presence on the channel, and is a function of the keying circuitry of the transmitter and the signal detection circuitry of the receiver. The random time has been chosen to be a multiple (0-15) of the transmitting station's key-up delay (TXDELAY). This is reasonable if one's own key-up delay is similar to that of other stations on the channel.

One other factor must be taken into consideration in optimizing data throughput. The currently implemented link protocols provide for relaying (digipeating) of packets. The acknowledgment procedure for such packets is that the relay station simply repeats packets without acknowledgment to the sending station. The receiving station sends its ACK signal back through the same digipeaters to the originating station. Since the digipeated packets are not acknowledged to the digipeater, an unsuccessful transmission must be retried from scratch by the originating station. In order to help alleviate the congestion of the frequency that tends to result when digipeated packets suffer collisions, the digipeater is given first shot at the frequency every time it becomes clear. Other stations, instead of transmitting as soon as they hear the channel clear, **must** wait a short time (DWAIT). This restriction applies to all stations except the digipeater, which is permitted to transmit relayed packets immediately. This prevents digipeated packets from suffering collisions except on transmission by the originating station.

AX.25 PROTOCOL SPECIFICATION

The following document is reproduced as a reference for those interested in the link-level protocol specified as a standard at the AMSAT packet conference of October 8-10, 1982. The Heath AX.25 level 2 protocol has followed this set of specifications closely.

The Heath Company gratefully acknowledges AMRAD for permission to reproduce this document.

Protocol Specification for Level 2 (link level).

Version 1.1, October 10, 1982

INTRODUCTION

The purpose of this document is to establish a standard protocol to be used at layer 2 of the ISO open systems interconnection reference model (OSI-RM) (commonly referred to as the link level) that will work effectively in the amateur radio environment with a minimum of overhead.

This protocol conforms with the ISO Standards 3309, 4335 (including DAD1&2), and 6256 high-level data link control (HDLC), and uses terminology found within that document.

This protocol also follows, in principle, the level 2 protocol used in the CCITT standard X.25. The only deviations from the letter of this standard are the extension of the address field and the inclusion of an additional frame, the unnumbered information (UI) response frame, which was taken from the HDLC standard.

This protocol is designed to work in either half- or full-duplex radio environments.

This standard has been written to work equally well for either point-to-point connections or connections through a network controller or other larger device.

This standard is not responsible for defining the operation of any other layer of the ISO OSI-RM.

DEFINITIONS

Two basic types of devices are used in packet networking. One is called "data circuit-terminating equipment" (DCE), which is usually a larger device such as a Metropolitan Network Controller (MNC) or some other device smart enough to handle the link connection. The other type is the data terminal equipment (DTE) device. The DTE is usually the originator of a connection, and could be considered the terminal end of the data link.

FRAME STRUCTURE

All transmissions shall be sent in frames. A frame shall be formatted as shown in Figure 1, which shows how unsequenced and supervisory frames are constructed, while Figure 2 describes an information frame.

First bit
sent

Flag	Address	Control	FCS	Flag
01111110	112/168 bits	8 bits	16 bits	01111110

Figure 1
U and S frame construction

First bit
sent

Flag	Address	Control	PID	Info	FCS	Flag
01111110	112/168 bits	8 bits	8 bits	N*8bits	16 bits	01111110

Figure 2
Information frame construction

FIELD DEFINITIONS

Flag Field

All frames shall begin and end with a flag, which is defined as one 0 followed by six 1s and another 0. A single flag may be used as the closing flag for one frame and the opening flag of the next frame.

Address Field

The address field in this protocol deviates from the letter of the present (CCITT yellow book) X.25 standard. This protocol uses extended addressing, and has both the source and destination addresses, where X.25 specifies only single-octet address. X.25 also sends either the DCE or DTE address, depending on who is sending it and whether a command or response is sent.

NOTE: The reason this document recommends sending both the destination and source addresses is to allow multiple DCE/DTE links to share the same data channel. There are two possible ways of accomplishing this. One way is to modify the connection establishing procedure to make sure both ends of the link know who they are connected to, and no other stations can accidentally foul up the link. The other way is to include both addresses in every frame, insuring that neither end of a link would ever get confused. In the long run, the additional overhead needed for sending both addresses in all frames seems worth tolerating in order to simplify link establishment and control procedures, and to avoid central assignment of brief addresses.

The encoding of the address field will be discussed later in this document.

Control Field

The control field consists of one octet. It is responsible for informing the stations on the link what type of frame is being sent, and is also where link control functions are transferred.

The contents of the control field are discussed later in this section.

Protocol Identifier (PID) Field

The Protocol Identifier field is one octet in length and is used to specify what type of protocol is being used at the next level (level 3). At this time, the following identifiers have been assigned:

12345678	(Bit Order of Transmission)
XXXX0000	No layer 3 protocol implemented.
XXXX01XX	AX.25 Level 3 protocol.
XXXX10XX	AX.25 Level 3 protocol.
XXXX1111	Next octet contains more identification information.

Where "X" is a don't care bit.

NOTE: Additional PID fields will be assigned as they become necessary.

Information Field

If an information field exists, it is totally transparent at the end-to-end points. It is bit stuffed over the link however, to prevent flags from accidentally appearing, which would cause an early frame ending, and errors.

The maximum length of the information field is 256 octets. This will allow 128 actual user-data octets with room for higher layer overhead. Larger lengths may be used by bilateral agreement.

FRAME CHECK SEQUENCE

The frame check sequence (FCS) consists of 16 bits generated in accordance with ISO 3309 (HDLC).

Bit Stuffing

Whenever a frame is being transmitted, all fields except for the flags will be checked to be sure that no more than five contiguous 1 bits exist. Any time that five contiguous 1 bits are detected, the transmitter must add a 0 bit after the fifth 1 bit. This added 0 bit will be detected at the receive end of the link and automatically deleted.

This bit-stuffing technique is necessary to insure that a flag sequence does not accidentally appear anywhere but at the beginning and end of the frames.

Order of Bit Transmission

All fields of each frame shall be sent starting with the least significant bit except for the FCS, which shall be sent starting with the highest order bit first, in accordance with ISO 3309.

Frame Abort

When a frame must be aborted, at least 15 contiguous ones must be sent, with no bit-stuffing 0s added.

Invalid Frames

Any frame consisting of less than 32 bits, or not bounded by opening and closing flags, or not consisting of an integral number of octets, should be considered an invalid frame by the link layer.

NONREPEATER (NORMAL) ADDRESS FIELD GENERATION

The address field is encoded as shown in Figure 3. This encoding system places both the destination and the source amateur radio call signs in the address field. The destination address is the address of the station this frame is being sent to. The source address is the address of the actual sender of the frame.

There is an extra octet at the end of each address subfield that allows room for a Secondary-Station IDentifier (SSID) and also reserves three bits for future expansion. The SSID allows one amateur to put up several packet stations and have them individually addressable at level 2. This is necessary or useful for functions such as repeaters, hosts, multiple terminals, etc.

A1 through A14 are the 14 octets that make up the two address subfields in the address field. The destination address is seven octets long (A1 through A7) and is sent first. This will allow it to be compared with the receiving station's address while the rest of the frame is being received. The source address is then sent in A8 through A14. Both of these address subfields are the same format, so just the destination subfield encoding will be shown here.

Destination Address							Source Address						
A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14

Figure 3
Address field encoding

Destination Sub-Field Encoding

Figure 4 shows how an amateur radio call sign is placed into the destination address subfield in octets 1 through 7 of the address field.

Where:

1. The first (low-order) bit sent, designated "E", of each octet is the HDLC address extender bit. This bit shall be a 0 for all but the last octet in the address field where it is set to 1.

2. The bits marked "R" are reserved bits, which may be used in an agreed upon manner in individual local networks. If they are not used, they should be set to 1.

3. "[A]" is the ASCII character of the amateur radio call sign to be encoded into the address octets. It is standard seven bit ASCII (uppercase letters only) that has been bit shifted left once to accommodate the HDLC extender bit. A1 is the first character of the call sign. If the call sign is less than six characters long, it will be left justified and padded at the trailing end with ASCII spaces (20 hexadecimal).

4. The SSID field is a Secondary Station ID that will allow amateurs to operate more than one packet station. The operation of the SSID field is left vague at this point, and is up to individual stations how this field is defined. Some suggested definitions for this field are as follows:

   0000-0111 Normal Packet Stations.
   1111 All-Call sub-address.
   L M
   S S
   B B

The all-call sub-address is useful when a station is requesting a connection to any of the destination station's equipment or if the SSID of the destination station is unknown.

A 1	A 2	A 3	A 4	A 5	A 6	A 7
E[W]	E[B]	E[4]	E[J]	E[F]	E[I]	E|SSID|RR0
01110101	00100001	00010110	01101001	00110001	01001001	0|SSID|110

LSB >>Order of Bit Transmission>> MSB

Figure 4
Callsign encoded into address field

Level 2 Repeater Address Encoding

When there is a level 2 repeater in operation, the HDLC address field is extended to include a third address subfield, which contains the address of the repeater that should repeat that frame. The position of the repeater address is shown in Figure 5.

Destination Address	Source Address	Repeater Address
56 bits (7 octets)	56 bits (7 octets)	56 bits (7 octets)
A1 to A7	A8 to A14	A15 to A21

Figure 5
Repeater address field encoding

The repeater address sub-field is encoded similar to the destination and source address sub-fields with the exception of the last octet, where an additional flag bit is added. This flag bit, called the H bit, is set to 0 by the source station; it is changed to a 1 by the repeater when it repeats a frame to indicate the sent frame has been repeated. This allows a station that might see both the frame originally sent by the source station and the repeated frame to distinguish between the two, and accept only the repeated frame. The encoding of the repeater address sub-field is shown in Figure 6.

Where:

1. "E" is the HDLC extender bit as mentioned earlier.
2. "H" is the has-been-repeated bit. If H=0, the frame has not been repeated; while if H=1, the frame has been repeated.

NOTE: Some of the advantages to using the WB4JFI addressing scheme are:

1. Every packet station will have a unique fixed address that does not change every time a new network is logged into.
2. Relocating to a new area will not cause major (or minor) problems.
3. Allows for more than 62 or 31 users at a time.
4. No local packet guru is needed to assign addresses with attendant concerns of backup and transfer during failure.
5. Direct or network operation requires no change of address.
6. All the problems with dynamic allocation/deallocation are eliminated.
7. Reduces local co-network interference due to users in overlapping local network RF domains with the same address fields.
8. With every frame having both the destination and source addresses in them, it will be a lot easier to set up and run multiple connections on the same data channel without having problems arise as to who is sending what frames to whom.

A15	A16	A17	A18	A19	A20	A21		
E[W]	E[B]	E[4]	E[J]	E[F]	E[I]	E	SSID	RRH
01110101	00100001	00010110	01101001	00110001	11001001	1	SSID	110

Figure 6
Repeater sub-address encoding

CONTROL FIELD FORMATS

The control field is used to convey commands and responses regarding the control and status of the data link.

The control field of this protocol uses the X.25 standard as a starting point, and adds an additional control field from HDLC to allow the protocol to work effectively during point-to-multipoint operation.

There are three basic formats of the control fields: the Information format (I frames), the numbered Supervisory format (S frames), and Unnumbered control frames (U frames). Figure 7 shows the basic format of these fields. Bit 1 is the first bit transmitted; bit 8 the last.

Control Field Type	Control Field Bits 1	2 3 4	5	6 7 8
I Frame	0	N(S)	P/F	N(R)
S Frame	1	0|S S	P/F	N(R)
U Frame	1	1|M M	P/F	M M M

Figure 7
Control field formats

Where:

1. N(S) is the send sequence number (bit 2 = low order bit).
2. N(R) is the receive sequence number (bit 6 = low order bit).
3. S means the supervisory functions bits.
4. M means the unnumbered modifier bits.
5. P/F is the Poll/Final bit.

Information Frame Control Field Definition

I frames have bit 1 of the control field set to 0. N(S) is the sender's send sequence number (the sequence number of this frame). N(R) is the sender's receive sequence number (the sequence number of the next expected received frame). The poll/final bit (P/F) will be discussed in a later section.

Supervisory Frame Control Field Definition

The S frame has the control field's bit 1 set high, and bit 2 set low. S frames provide supervisory link control such as acknowledging or requesting retransmission of I frames, and link level window control. Since S frames don't have an information field, the sender's send variable and the receiver's receive variable are not incremented.

Unnumbered Frame Control Field Definition

U frames are distinguished by having both bits 1 and 2 of the control field set to 1. U frames are used to extend the number of link supervisory functions beyond those allowed as S frames. U frames are responsible for the setting up and tearing down of the data link, along with other miscellaneous functions. Some U frames may contain an information field.

Control Field Parameters

Sequence Numbers and Variables — For the basic (nonextended) control field, every I frame shall be assigned a sequential number varying from 0 to 7. This will allow up to seven outstanding I frames at a time.

Send State Variable V(S) — The send state variable is an internal variable (never sent) that contains the next sequential number to be assigned to the next transmitted I frame. This variable is updated with each successive I frame sent.

Send Sequence Number N(S) — The send sequence number is found only in I frames. It is the sequence number of the I frame being sent. Just prior to the sending of the I frame, N(S) is updated to equal the send state variable.

Receive State Variable V(R) — The receive state variable is an internal variable that contains the number of the next expected I frame to be received. This variable is updated upon the reception of an error-free I frame whose send sequence number equals the present receive state variable value.

Receive Sequence Number N(R) — Both I and S frames contain N(R), the sequence number of the next expected received I frame. Prior to sending an I or S frame, this variable is updated to equal that of the receive state variable. Transmission of this updated N(R) implicitly acknowledges the proper reception of all I frames up to and including N(R)-1.

Poll/Final Bit — The poll/final bit can be used with all types of frames. It is used in a command (poll mode) to request an immediate reply to a frame. The reply to this poll is indicated by setting the P/F bit (final mode) in the appropriate response frame. Only one poll command is allowed per direction at a time. Implementation of the P/F bit will be discussed later in this recommendation.

CONTROL FIELD ENCODING

Information Frames — The information frame control field is encoded as shown in Figure 8. Information frames are used to convey user data across the link. These frames are sequentially numbered to maintain control of their passage along the link. The I frame control field used here conforms with both the X.25 and ADCCP standards.

Control Field Bits

1	2	3	4	5	6	7	8
0	N (S)			P/F	N (R)		

Figure 8
Frame Control Field

Supervisory Frames

The supervisory frame control fields are encoded as shown in Figure 9. S frames are used in this standard only as response frames. These fields conform with both X.25 and ADCCP (except for SREJ not being implemented from ADCCP).

Control Field Response		Control Field Bits 1	2	3	4	5	6	7	8
Receive Ready	RR	1	0	0	0	P/F	N(R)		
Receive Not Ready	RNR	1	0	1	0	P/F	N(R)		
Reject	REJ	1	0	0	1	P/F	N(R)		

Figure 9
S frame control fields

VADCG PROTOCOL SPECIFICATION

This document is reproduced from "Packet Radio HDLC Protocol Notes" by Hank Magnuski, KA6M, August 15, 1981. The Heath Company gratefully acknowledges the author for permission to reproduce this document.

The protocol used by the Vancouver Digital Communications Group on their controller board, and also used by the packet radio repeater, is based on a subset of HDLC standard protocol. In this protocol, the standard unit of information is the frame:

SYNC	FLAG	ADDR	CNTL	TEXT	FCS1	FCS2	FLAG

where,

- SYNC — Preframe synchronization, idle flags or 0s.
- FLAG — Start of frame, bit pattern 01111110.
- ADDR — Address byte, hexadecimal 00 to FF.
- CNTL — Control byte, which indicates type of frame and other information.
- TEXT — Optional information field.
- FCS1 — First byte of frame check sequence (CRC).
- FCS2 — Second byte of frame check sequence.
- FLAG — Closing flag.

Other Features Used:

- Bit stuffing — Provides fully transparent transmission of data.
- NRZI encode — 0s cause transition which allows clock recovery.
- Multiframe — Up to seven frames permitted in a single transmission.

Types of Frames:

Non-Sequenced-
Information — Used for Connect/Disconnect.
Supervisory — Used for window and flow control.
Information — Used for transmission of text.

NSI Frames:

FLAG ADDR CNTL FMCALL TOCALL FCS1 FCS2 FLAG

ADDR — Address of calling station (assigned to each station).

CNTL — 17H – connect request.
07H – disconnect acknowledge.
53H – disconnect request.
43H – disconnect acknowledge.

The poll/final (P/F) bit, 10H, is used to force a response from the receiving station. It is used here and in other frame types for this function.

FMCALL — Call of station originating the frame (six characters).

TOCALL — Call of station receiving the frame (six characters).

The call sign is left-justified in the field and padded with trailing blanks if the call is shorter than six characters.

Supervisory Frames:

FLAG ADDR CNTL FCS1 FCS2 FLAG

ADDR Address of sender

CNTL

7 6 5	4	3	2	1	0	
NR	P/F	0	0	0	1	Receive ready
NR	P/F	0	1	0	1	Receive not ready

NR Sequence count of next expected I-frame P/F Poll/final bit.

Information Frames:

FLAG ADDR CNTL TEXT FCS1 FCS2 FLAG

ADDR Address of sender.

CNTL

7 6 5	4	3 2 1	0	
NR	P/F	NS	0	I-frame

NR Sequence count of next expected I-frame P/F Poll/final bit.

NS Sequence count of this I-frame.

TEXT Text field, 128 bytes maximum, ASCII code.

Timeouts:

T1 Receive timeout, 2-3 seconds.

T1S Frame timeout, time for frame of maximum length.

TR Delay time (random) prior to transmission of first frame of a sequence, 150 milliseconds to 1.25 seconds.

EXTERNAL MODEM

The Heathkit TNC design includes provisions to completely bypass the on-board modem. This allows the TNC to be used with higher-speed or special-purpose modems, experimentation with modem techniques, and so forth. The following information is intended primarily for those who wish to interface external modems to the TNC. Familiarity with modem and serial data channel terms is assumed.

20-pin connector J5 is available for disconnecting the onboard modem, and allows connection of an external modem at TTL interface levels. A TTL high level is greater than 2.4 volts but less than 5.25 volts, while a TTL low level is greater than −0.4 volt but less than 0.8 volt. CAUTION: Do NOT connect an RS-232C level modem directly to J5.

Normally, jumpers are installed to connect pins 1-2, 5-6, 7-8, 9-10, 11-12, 13-14, 17-18 and 19-20. If you wish to use the on-board modem, all of these jumpers must be installed.

J5 CONNECTOR PINOUTS

Pin 1 — Carrier Detect In

This pin tells HDLC controller U17 that a valid data carrier has been detected. It should be pulled high when no carrier is detected and low when a carrier is present. This line must be implemented to use the software in the TNC unless the software release notes indicate otherwise.

Pin 2 — Carrier Detect Out

This is an output from the on-board modem and meets the requirements outlined for pin 1, above. It is normally jumpered to pin 1 when the on-board modem is used.

Pin 3 — $\overline{\text{CD1}}$

This pin is normally tied to ground via pull-down resistor R76, and tells the HDLC controller to interrupt the µP (microprocessor) when a negative-going edge is applied to Carrier Detect In, pin 1. Tying this pin high will disable this edge. This pin will normally be left unconnected.

Pin 4 — $\overline{\text{CD0}}$

This pin is normally tied to ground via pull-down resistor R77 and tells the HDLC controller to interrupt the uP when a positive-going edge is applied to Carrier Detect In, pin 1. Tying this pin high will disable this edge. This pin will normally be left unconnected.

Pin 5 — $\overline{\text{MSCOT}}$

This is an output line from the HDLC controller. Unless indicated by the software release notes, it is used to key the attached transmitter and must be connected for proper operation of the radio link. This pin is high when the transmitter is commanded off and low when the transmitter is to be keyed.

Pin 6 — Xmtr Key In

This is an input to the on-board modem; it conforms to the specifications outlined on Page 5-1 for pin 5. The on-board modem features a hardware "watchdog" timer to protect the packet channel from a runaway TNC that always tries to key the transmitter. The time constant is about one minute, but can be changed by selection of R28 on header U35.

Pin 7 — $\overline{\text{DSR}}$

This is an input to the HDLC controller; it is used to tell the TNC that the attached modem is ready for operation. The on-board modem has no initialization time and simply returns this line to pin 8, described below. This pin must be satisfied for the TNC to operate properly, unless the software release notes indicate otherwise.

Pin 8 — $\overline{\text{DTR}}$

This is an output from the HDLC controller to the modem, which tells the TNC that the HDLC port is ready for operation. If the modem has no use for this line, it should be returned to pin 7 as mentioned above.

Pin 9 — $\overline{\text{RTS}}$

This is an output from the HDLC controller; it tells the attached modem that the HDLC port has data to send. It is used as a handshake with $\overline{\text{CTS}}$, pin 10 (below), to synchronize the sending of data from the TNC when the modem requires it. The line will be high when the HDLC controller has nothing to send to the modem and low when it has data.

Some external modems may use this line; the on-board modem does not and simply returns it to pin 10.

Pin 10 — $\overline{\text{CTS}}$

This is an input to the HDLC controller; it tells it that the modem is ready to accept data. It must be connected to allow proper operation of the HDLC controller, unless the software release notes indicate otherwise. The pin must be high to indicate the modem is not ready for data, and low to indicate the modem is ready to accept data.

The on-board modem simply returns this line to pin 9.

Pin 11 — $\overline{\text{TC}}$

This is the transmitter clock line. While various duty cycles widths are acceptable, a square-wave clock is preferred. If the DPLL is enabled (pin 15, below), the clock frequency must be 32 times the data rate (38.4 kHz for 1200 baud); otherwise the clock must be equal to the data rate (1200 Hz for 1200 baud).

Pin 12 — Clock Out

This line is tied to the on-board HDLC clock generator. It runs at 32 times the data rate for the HDLC port and provides a square-wave signal under software rate control.

Pin 13 — $\overline{\text{RC}}$

This is an input to the HDLC controller of the receive data rate clock. The same restrictions apply to this pin as apply to pin 11, above.

Pin 14 — Clock Out

This pin is physically tied to pin 12, described above.

Pin 15 — $\overline{32X}$

This is an input to the HDLC controller; it is used to select the on-chip digital phase-locked loop (DPLL) for data clock recovery. When the pin is held low, the DPLL is selected and the supplied $\overline{TC}$ and $\overline{RC}$ clocks must be 32 times the desired data rate. Pull-down resistor R79 is provided to set the default value of this pin to enable the DPLL. When this line is high, the supplied clocks must be equal to the desired data rate AND the receive clock must be synchronous with and in a certain time relationship to the received data. See the Western Digital data sheet listed in the Bibliography for details.

Pin 16 — $\overline{NRZI}$

This is an input to the HDLC controller; it tells it to format its output, and decode its input data stream, as NRZI (non-return to zero, inverted) when low and as NRZ when high. Normal packet radio usage to date has used the NRZI format for data as standard, and pull-down resistor R78 is provided to configure the default state of this pin.

Pin 17 — RXD

This is the received data input to the HDLC controller from the modem.

Pin 18 — Receive Data Out

This pin provides receive data from the on-board modem to the HDLC controller.

Pin 19 — TXD and $\overline{MISCIN}$

The TXD pin is the HDLC controller's transmitted data output to the modem. The format will be NRZ or NRZI, depending on the state of that control line (see pin 16, above).

The $\overline{MISCIN}$ input pin is used in conjunction with the on-board modem and control logic on the TNC to ensure the FSK ID is "right side up" when sent.

Pin 20 — TX Data Input

This input line accepts data to be transmitted by the modem.

In addition to the modem disconnect, three other lines are made available to the user from the HDLC controller, $\overline{RI}$, $\overline{RI1}$ and $\overline{RI0}$. These lines are normally disabled by pullup resistors R73, R74, and R75 respectively, and are used to program interrupt response by the HDLC controller to a "ringing" signal supplied by an external modem. Since these lines are not needed in a radio application, they have been disabled and the software ignores them. For further details, refer to the manufacturer's data sheet for the HDLC controller.

If you elect to use an off-board modem, be sure to properly shield the connecting cables, etc., as the TNC may be susceptible to RFI.

TUNING INDICATOR INTERFACE

In order to facilitate communications on HF and OSCAR, the Heath Company TNC includes space for a 5-pin plug (J7 — not supplied) for connecting a tuning indicator. You could use either an oscilloscope or a specialized LED-style unit for this purpose. Please refer to the Exar Application Note (listed in the "Bibliography" at the back of this Manual) for details on functions of the XR2211 signals available on this connector.

CONNECTOR PINOUTS

Pin 1 — Ground

This pin is the TNC's analog ground reference. Do **NOT** use this pin to sink appreciable current or the modem's weak-signal performance may be compromised.

Pin 2 — Loop-Data Filter Output

This pin is connected to the output of the XR2211 PLL data filter. It is a high-impedance source; therefore, be sure that no extraneous signals or low-impedance loads are attached to it.

Pin 3 — Demodulator Reference Voltage

The internal XR2211 data comparator reference voltage is available on this pin. By comparing this value with the signal present on pin 2, you will be able to properly tune your radio. As for pin 2, you must carefully shield this pin from noise, as it has a high internal impedance.

Pin 4 — Data Carrier Detect

This pin is an open-collector output that drops to near ground potential when valid data is not present.

Pin 5 — +12 Volts

This pin is a +12 VDC source. Do **NOT** use this pin to source more than a few milliamperes of current; otherwise, degradation of the on-board modem's weak-signal performance may result.

SPECIAL FUNCTIONS

TRACE FUNCTION

The trace function is a protocol debugging tool. It allows you to examine the frame structure of sent and received packets. If you report difficulties with the software, you may wish to include output from the trace function with such a report. Use the TRACE command to individually enable trace options.

Among the frame dumping options are dump frames sent, dump frames used (frames addressed to this TNC), dump all frames read, dump digipeated frames, and dump FRMR frames. Selecting all options will result in some frames being dumped more than once. Keep in mind that enabling all trace functions on a TNC, whose terminal baud rate is no faster than the HDLC baud rate, will quickly fill up the terminal output buffer. Refer to "Editing Commands" in the "Operation" section of this Manual for fast relief from this condition.

The frame dump output contains three sections and will be properly formatted if the screen is at least 80 characters wide. Unless the option to dump the entire frame is enabled, the trace dump will show the header only.

The first section (left-most column) contains the hexadecimal representation of the actual bytes transmitted or received. (If you examine the I/O buffers used by the transmitting and receiving routines, you will notice no apparent resemblance to this data. The data is complemented prior to being stored in the buffer to compensate for the reversed logic of the WD-1935 chip. Convert 0s to 1s and vice versa.)

The second section (middle column) of the dump displays the packet contents as ASCII characters, after right-shifting each byte by one bit. You should be able to easily identify the structure of the addresses used from this section; AX.25 protocol calls for address fields consisting of ASCII call signs left-shifted by one bit.

The third section (right column) displays the packet as ASCII characters. The header information in this section will look funny, but you should be able to read the message, if any, from this section.

Debug Program

This program was written as a debugging tool for the authors of the software. If you wish to modify the TNC software or examine the hardware I/O addresses, you may find the commands described here useful. In addition, you may wish to use this program to examine memory locations as supplementary information to accompany reports of software bugs.

You may enter the debug program by typing the character specified by the DEBUG command, by default <CTRL-E>. Use the same character to exit the debug program. Following the exit command, the message "Bye" will be displayed.

When you enter the debug mode, a prompt symbol (:) will be sent to the terminal. This prompt will also be sent to the terminal after completion of a debug mode command, except the exit command.

NOTE: All numbers input to the debug program are in hexadecimal, and the prefix ($) is optional. This is different from the convention employed in the command mode. The program does not parse numbers to insure that they are within appropriate bounds. If you type a number which is too large, the low-order part of the number will be used and no warning will be given.

The debug program operates with a very rudimentary and intolerant parser. Instructions which are not recognized do not produce warnings or error messages; they simply are not executed. Parsing of numeric input stops when an invalid (nonnumeric) character is encountered.

The following is a summary of the debug mode commands. Start each command by entering a special command character, and terminate it by entering <RETURN>. The command characters are @, #, >, T, L, and Q. User-supplied variables are identified here in lowercase, and explained in the following text. The text enclosed in square brackets is self-explanatory.

Display Storage

@address

where "address" is the address whose contents you wish to see displayed.

The program will respond with:

```
address content [no <cr>]
```

"Content" is the value read by the processor when the specified location is addressed. (NOTE: Some peripheral chips address different registers on read and write access of the same location.) Typing a <cr> will cause the program to display the next address and contents. Terminate this mode by typing "Q".

Example:

```
:@AC00
AC00 56
AC01 41
AC02 4C Q
:
```

NOTE: Following the prompt symbol, you typed the characters "@AC00" followed by three <cr>s. This requested the contents of three bytes starting at hexadecimal AC00. The storage location that was asked for contained the ASCII characters "VAL". "VAL" was then sent to the terminal as six ASCII characters showing the hexadecimal representation of the storage content. Finally, you terminated the display function by typing "Q" and a last <cr>.

There is no restriction as to what address you specify. You can specify RAM, ROM, some hardware device, or a nonexistent address. The result of asking for the contents of a nonexistent address is not predictable and probably not very useful. Keep in mind that reading some I/O locations may affect hardware status. Not all of the address lines are used in addressing the I/O chips, so a given location may be accessed by more than one address. The memory socket for U8 is mapped for 16K of RAM or ROM. If you use this socket for a memory smaller than 16K, each location will be accessed by more than one address.

Write to Storage

@address = data

where "address" is the address of the location you want to put the data into, and "data" is the hexadecimal representation of one byte of data. The program will respond by prompting you with the next address and an " = " sign:

```
nextaddress = [no <cr>]
```

You can load subsequent addresses by typing only the data to be stored. Typing a <cr> only will enter 0 into the location. Terminate this mode by typing "Q".

If you try to write to a nonexistent address or to a read-only address, no warning will be given. If you write to a hardware control register, you may affect the behavior of the TNC in an unexpected way.

Example:

```
:@1F00=48
1F01=49
1F02=Q
:
```

The storage area starting with 1F00 has been filled with the ASCII characters for "HI".

Registers

You can examine the contents of the registers as they appeared just prior to entering the debug mode. To examine the register contents, enter:

```
#register
```

Here, "register" is a register code with the following explanation: C (condition code), A, B, D (AB), G (direct page), X, Y, U, S, P (program counter). The contents of D, X, Y, U, S, and P will be displayed as double-byte quantities. The program will respond with:

```
#register content
:
```

To alter register contents, enter:

```
#register=data
```

where "data" is the single-byte quantity to be stored in registers C, A, B, or G, or a double-byte quantity to be stored in registers D, X, Y, U, S, or P. The program will respond with a prompt symbol. The values set in this procedure will be moved into the actual registers upon exit from the debug program. Altering register contents can produce bizarre results.

Start Execution at Address

>address

where "address" is the location in storage which contains the next instruction you want executed. A call (JSR) to that location will be executed. If the instruction sequence executed terminates with an RTS instruction, the execution will return to the debug program and the prompt symbol will be typed. If the routine starting at address requires initialization of registers prior to entry, you should supply a routine to do so. The register contents on entry to the debugger are not loaded for the subroutine call, and are not affected by the call.

If you wish to load test routines into RAM for execution using this command, you should be careful not to write over areas used by the TNC operating system. For the version 3.0 release, the RAM area from $1800 to $2000 is unused, as well as any RAM occupying the U8 socket.

Copy

Tsource destination

where "source" is the source address range expressed in one of the following ways:

address
address!length
address:endsource

and "destination" is the beginning of the destination address range. The source and destination fields are separated by a space.

List

Lsource

will copy addresses and contents from the address(es) specified by "source" to the terminal. The format of "source" is the same as for the copy command.

BIBLIOGRAPHY

American Telephone and Telegraph Company, "Operations Systems Network Communications Protocol Specification BX.25," Issue 2, Publication 54001, 1979.

David W. Borden, K8MMO, and Paul L. Rinaldo, W4RI, "The Making of an Amateur Packet-Radio Network," QST, October 1981, pp. 28-30.

Exar Integrated Systems, "Phase-locked Loop Data Book," Exar Applications Note AN-01, P. O. Box 62229, Sunnyvale, CA 94088, 1981.

Terry Fox, WB4JFI, "AX.25 Level 2 Protocol," Proceedings of the Second ARRL Amateur Radio Computer Networking Conference, March 19, 1983, p. 21.

David Henderson, KD4NL, "Design Decisions for the TAPR TNC Link Level," Proceedings of the Second ARRL Amateur Radio Computer Networking Conference, March 19, 1983, p. 21.

International Standards Organization, "Reference Model of Open Systems Architecture," Document ISO/TC97/SC16/N227, June 1979.

Lyle Johnson, WA7GXD, "Using the Xicor NOVRAM," QEX, August 1982, p. 4.

Lyle Johnson, WA7GXD, "Join the Packet-Radio Revolution," 73 Magazine, September, 1983, pp. 19-24.

Lyle Johnson, WA7GXD, "Join the Packet-Radio Revolution — Part II," 73 Magazine, October, 1983, pp. 20-31.

Lyle Johnson, WA7GXD, "Join the Packet-Radio Revolution — Part III," 73 Magazine, January, 1984, pp. 36-44.

Margaret Morrison, KV7D, "Real-Time Low-Level Software on the TAPR TNC," Proceedings of the Second ARRL Amateur Radio Computer Networking Conference, March 19, 1983, p. 21.

Margaret Morrison, KV7D, and Dan Morrison, KV7B, "Designing the TAPR TNC Audio Input Filter," Proceedings of the Second ARRL Amateur Radio Computer Networking Conference, March 19, 1983, p. 54 (Available from ARRL Headquarters for $9).

Margaret Morrison, KV7D, and Dan Morrison, KV7B, "Amateur Packet Radio: part 1," Ham Radio, July 1983, pp. 14-18.

Margaret Morrison, KV7D, Dan Morrison, KV7B, and Lyle Johnson, WA7GXD, "Amateur Packet Radio: part 2," Ham Radio, August 1983, pp. 18-29.

National Semiconductor, "Technical Product Information, MF10", 2900 Semiconductor Drive, Santa Clara, CA 95051.

Harold E. Price, NK6K, "Multi-use Design Considerations for the TAPR TNC," Proceedings of the Second ARRL Amateur Radio Computer Networking Conference, March 19, 1983, p. 62.

Andrew S. Tanenbaum, "Computer Networks," Prentice-Hall, 1981.

Western Digital, "Western Digital WD193X Controller," Components Group, 2445 McCabe Way, Irvine, CA 92714.

INDEX

Heathkit®

Heathkit®

EAST BAY RADIO CLUB W6CUS 145090 MAILBOX

116110 LONG KEYDOWN

ARRL PAPER

TMC 2 BOOK

CUSTOMER SERVICE

REPLACEMENT PARTS

Please provide complete information when you request replacements from either the factory or Heath Electronic Centers. Be certain to include the **HEATH** part number exactly as it appears in the parts list.

ORDERING FROM THE FACTORY

Print all of the information requested on the parts order form furnished with this product and mail it to Heath. For telephone orders (parts only) dial 616 982-3571. If you are unable to locate an order form, write us a letter or card including:

- Heath part number.
- Model number.
- Date of purchase.
- Location purchased or invoice number.
- Nature of the defect.
- Your payment or authorization for COD shipment of parts not covered by warranty.

Mail letters to: Heath Company
Benton Harbor
MI 49022
Attn: Parts Replacement

Retain original parts until you receive replacements. Parts that should be returned to the factory will be listed on your packing slip.

OBTAINING REPLACEMENTS FROM HEATH ELECTRONIC CENTERS

For your convenience, "over the counter" replacement parts are available from the Heath Electronic Centers listed in your catalog. Be sure to bring in the original part and purchase invoice when you request a warranty replacement from a Heath Electronic Center.

TECHNICAL CONSULTATION

Need help with your kit? — Self-Service? — Construction? — Operation? — Call or write for assistance. you'll find our Technical Consultants eager to help with just about any technical problem except "customizing" for unique applications.

The effectiveness of our consultation service depends on the information you furnish. Be sure to tell us:

- The Model number and Series number from the blue and white label.
- The date of purchase.
- An exact description of the difficulty.
- Everything you have done in attempting to correct the problem.

Also include switch positions, connections to other units, operating procedures, voltage readings, and any other information you think might be helpful.

Please do not send parts for testing, unless this is specifically requested by our Consultants.

Hints: Telephone traffic is lightest at midweek — please be sure your Manual and notes are on hand when you call.

Heathkit Electronic Center facilities are also available for telephone or "walk-in" personal assistance.

REPAIR SERVICE

Service facilities are available, if they are needed, to repair your completed kit. (Kits that have been modified, soldered with paste flux or acid core solder, cannot be accepted for repair.)

If it is convenient, personally deliver your kit to a Heathkit Electronic Center. For warranty parts replacement, supply a copy of the invoice or sales slip.

If you prefer to ship your kit to the factory, attach a letter containing the following information directly to the unit:

- Your name and address.
- Date of purchase and invoice number.
- Copies of all correspondence relevant to the service of the kit.
- A brief description of the difficulty.
- Authorization to return your kit COD for the service and shipping charges. (This will reduce the possibility of delay.)

Check the equipment to see that all screws and parts are secured. (Do not include any wooden cabinets or color television picture tubes, as these are easily damaged in shipment. Do not include the kit Manual.) Place the equipment in a strong carton with at least THREE INCHES of *resilient* packing material (shredded paper, excelsior, etc.) on all sides. Use additional packing material where there are protrusions (control sticks, large knobs, etc.). If the unit weighs over 15 lbs., place this carton in another one with 3/4" of packing material between the two.

Seal the carton with reinforced gummed tape, tie it with a strong cord, and mark it "Fragile" on at least two sides. Remember, the carrier will not accept liability for shipping damage if the unit is insufficiently packed. Ship by prepaid express, United Parcel Service, or insured Parcel Post to:

Heath Company
Service Department
Benton Harbor, Michigan 49022

WB6FMC-1 145.03 Questa Peak
WD6ZD-1 145.01

mailbox { W6 IXU via WD6ZD-1
" via W6AMT-1

KB6SC 145.03

to call Cliff: on 145.03 or .05
145.03 is the BS packet.

HEATH ZENITH

mailbox W6IXU Arroyo Grande

C W6IXU V WD6Z-1
or C W6IXU V W6AMT-1 C-PRB V SFO

HEATH COMPANY • BENTON HARBOR, MICHIGAN
THE WORLD'S FINEST ELECTRONIC EQUIPMENT IN KIT FORM

LITHO IN U.S.A.