November 1983

FIRST EDITION
1983

First Printing - November, 1983
Second Printing - February, 1984
Third Printing - June, 1984
Fourth Printing - December, 1984

CONTENTS

Chapter V Hardware

Chapter VI Troubleshooting

Chapter VII Protocols

Appendix A Illustrations

ILLUSTRATIONS

CHAPTER I

INTRODUCTION

Welcome To Packet Radio

Packet radio is one of the newest forms of communication that has been made available to Amateurs. It represents a giant step forward in reliable communication. We are glad you have joined the growing ranks of packet radio users throughout the world, and hope you will find it useful and exciting.

The TAPR Terminal Node Controller (TNC) kit is the culmination of approximately two years of volunteer effort by many people, including nearly 170 participants in TAPR's "Beta Test," which involved the distribution of TNCs throughout the world for testing, experimentation, and criticism. As a result, we believe this is one of the most thoroughly tested projects to be done by a non-profit, volunteer-labor Amateur organization. We hope you share our enthusiasm!

Brief Description of Amateur Packet Radio

Communication via Amateur packet radio involves the transmission of digital data by means of radio, making use of typical ham transceivers. Unlike ordinary RTTY or ASCII, the data does not pass directly from the source to the radio or vice-versa. Instead, it is routed through a terminal node controller, where the special characteristics of this mode are implemented. TAPR's terminal node controller is a microprocessor based device which is connected to a source of digital data via its RS-232C serial data port, and to an ordinary Amateur radio transceiver via its radio port. Thus, it stands between the digital device and the radio.

Usually the TNC will be connected to a data terminal or computer via the data port, and all commands to the TNC as well as data to be transmitted will come from this device. The TNC handles all the chores involved in actually sending and receiving the data, including packaging the data into bursts called packets or frames, keying and modulating the transmitter, and listening for and demodulating packets from the receiver.

In fact, the TNC does many tasks associated with maintaining the data channel, such as checking the integrity of packets received, and acting as a relay station when so requested by other users of the channel. All of these functions are handled completely automatically by the software and hardware of the TNC. The user need only be concerned with producing and making use of the digital data, leaving the multitude of communications tasks to the TNC.

A typical packet QSO begins with one user commanding his TNC to issue a "connect-request" packet directed to the other station. The connect-request packet includes both users' call signs, so the receiving station recognizes that the connect-request is directed toward him. The receiving station acknowledges receipt of this packet and the connection is established. Each user sees

    *** CONNECTED to <CALLSIGN>

appear on his terminal, where <CALLSIGN> is the call sign of the other station. The TNC then automatically switches to data transfer mode. From then on all data is sent to the other station whenever a packet of information is completed.

Each packet is a "sandwich" consisting of, in order of transmission, header information, followed by the data (if any -- some packets contain no user data), followed by a very important 16 bit number called the Frame Check Sequence (FCS). It is this packaging of the data that sets packet radio apart from all previous types of Amateur communication, and which gives it special properties.

The two primary characteristics of packet radio communications are first, that it is "bursty" -- most of the time, channel occupants are not transmitting. (However, exceptions to this rule occur, for example, during computer-to-computer file transfers.) Second, the received data which is sent out the TNC's data port is essentially guaranteed to be error-free. This is due to error detection associated with the FCS following the data, which is used by the receiving station to determine the validity of the data. In the event the data is incorrect, the transmitting station will re-transmit the packet, up to a user-specified number of times, in an attempt to get the data through with no errors.

Packet radio QSOs are normally terminated when one of the stations issues a "disconnect-request" packet. This is acknowledged by the receiving station, and both users see

    *** DISCONNECTED

appear on their terminal.

At any time the TNC may be switched to "command mode" and one of more than sixty user settable parameters may be inspected or changed. These parameters control all facets of the TNC's operation, and many of them are stored in NOnVolatile Random Access Memory (NOVRAM) (trademark of Xicor Corp.) so they need not be continually reset at every power-up.

Protocols

The rules by which the software determines responses to received data, as well as to the user's commands and data, are known as the protocol. For example, when a TNC issues a connect-request packet it expects a reply in the form of a "connect-request acknowledged" packet. If all goes well the receiving station will issue such a packet, the connection will be established, and each user will see

    *** CONNECTED to <CALLSIGN>

appear at his terminal. However, if no proper response is received, the originating TNC will re-issue the request for a predetermined number of times and then give up. All this behavior is dictated by the protocol.

As another example, if two or more QSOs are taking place on the same channel (one of the great advantages of this mode) and a simultaneous transmission by two stations occurs (a collision), the subsequent time delays and retransmissions are determined by rules contained in the protocol.

While a great many of the tasks required of the protocol are determined once the basic ground rules are laid out (some are immediately obvious, others less so), most of the implementation details are completely arbitrary. In particular, the exact composition of the frame is very flexible, and no two designers would be expected to independently hit upon identical digital frames. On the other hand, no communication at all is possible without perfect agreement on just this point, among many others!

For this reason, a meeting hosted by AMSAT was held during October, 1982 which included most of the major packet radio interest groups, in order to decide upon a single mutually agreeable protocol for use on the AMICON channel on the then upcoming Phase III B satellite. Such agreement was reached, and the presently accepted protocol (one of two implemented on the TAPR TNC - the other is the VADCG protocol) is known as AX.25, named after the international commercial protocol X.25, upon which it is based. AX.25 is currently the most widely used protocol in Amateur packet radio throughout the world. A substantial portion of this manual describes the usage of the two protocols on the TNC and their formal definitions are contained in Appendices B and C.

Hardware

In addition to the TNC characteristics implemented in software, several hardware features are notable. The power supply on the board provides four different voltages: +5 volts, -5 volts, +12 volts and -12 volts. The audio circuitry obtains its +5 volt power from its own regulator, and the audio ground is tied to the digital ground at a single point. These features, along with generous bypassing, give high noise immunity to the TNC.

The packet structure outlined above, which is used by both protocols on the TNC, is generated in software. However, the production and reception of these frames as serial, synchronous streams of data is handled by a special purpose Very Large Scale Integrated (VLSI) circuit which relieves the processor of most of the onerous tasks involved in this process, including the jobs of automatic generation and inspection of the FCS value. Not only does this make writing the software more pleasant, it allows full duplex operation at data rates up to 4800 baud. (Modem modification or an outboard modem is required if the radio channel is to be operated above 1200 baud.)

Great attention was paid to the TNC modem design. As a result of many experiments, and suggestions received during our Beta test, the Beta version modem design has been modified in crucial areas, and the design on the present TNC should prove to be one of the most reliable, especially for demodulating signals in the presence of background noise. This is due in part to the input filter system along with a quiet power supply.

Furthermore, in order to make the TNC as versatile as possible, all critical passive components for both the modulator and demodulator are mounted on special headers which simply plug into ordinary IC sockets. This permits easy tinkering and experimentation, and allows complete modem reconfiguration in seconds. As an additional aid for those wishing to use altogether different modulation/demodulation schemes, all signals to and from the modem are coupled via a multi-pin connector which is normally shorted by jumpers. In the event a separate modem is desired, the jumpers may be removed and these signals passed to the off-board modem via a standard connector.

The details of the TNC's performance are described in this manual, from the exact meaning of each bit in a frame to the type of modulation and demodulation used for the radio port. In addition, the manual contains complete descriptions of each of the many commands available to the TNC user. These commands provide great flexibility in configuring the TNC to provide precisely the type of service each individual user requires.

What Is TAPR?

Tucson Amateur Packet Radio Corporation is a nonprofit R & D group whose primary goal has been, and continues to be, the development of packet radio technology for the Amateur community. It was formed in 1981, originally as a club, and incorporated in mid-1982. From the beginning, the intention was to create an inexpensive way for hams to participate in this mode of communication.

Prior to our Beta test, the only Amateur packet radio was done by dedicated, hard-core experimenters using home-brew systems or systems based on the pioneer VADCG TNC. With the distribution of the 170-odd TAPR Beta boards, packet radio was introduced at the "black box" level. The TNC in this kit incorporates all the hard-won improvements resulting from that test, and should prove highly reliable.

TAPR presently (December, 1984) consists of about 1200 members spread over many countries, and continues to grow. Our interests are diverse, including such projects as higher speed local-area-net operation, PACket SATellite (PACSAT), and inter-net linking. We are always looking for volunteers with interest in these areas, and encourage you to discuss your ideas with us.

The Packet Status Register is the official bi-monthly publication of TAPR and is distributed to members as a way to keep them up to date on the latest TAPR activities and other interesting aspects of packet radio.

Problem Reporting

Please attempt to solve locally as many of your kit building problems as possible. The distribution of this kit is due primarily to the efforts of a very few people, all of whom do this as a hobby, in their spare time and without pay. For this reason, we are unable to offer a repair service for inoperative kits. However, in the event you are really at an impasse (please take heed of the suggestions in Chapter VI, TROUBLESHOOTING) write us at the following address and we will attempt to offer you suggestions to get your TNC running.

Our address:

Tucson Amateur Packet Radio
P.O. Box 22888
Tucson, AZ 85734

PLEASE DO NOT RETURN TO US ANY CONSTRUCTED TNC OR PARTIALLY CONSTRUCTED KIT WITHOUT CONTACTING US FOR WRITTEN PERMISSION.

CHAPTER II

HOOKUP

Board Layout and Power Connections

Throughout this chapter, references made to parts located on the board use the following convention: viewing the board from the component side, the top is the edge closest to the RAM and EPROM memory bank (U7-U12). The label "Tucson Amateur Packet Radio" and most of the silk screen legends will be right side up if the top edge is away from you as you look at the board. The wire wrap area will be at top left and the three interface connectors will be at the lower edge closest to you. The reverse side of the board, which has no silk-screen legends, will also be referred to as the "solder side" of the board.

Pin 1 of all Integrated Circuits (ICs) is marked on the silk screen. In addition, a label for pin 1 for most ICs is included in the trace etching on the solder side of the board. IC pins are numbered consecutively down the left side of the chip, then back up the right side when viewed from the top. For example, pin 4Ø of a 4Ø-pin IC will be opposite pin 1. Pinouts for the modem disconnect, power connector, and all interface connectors are illustrated in Fig. 2.1.

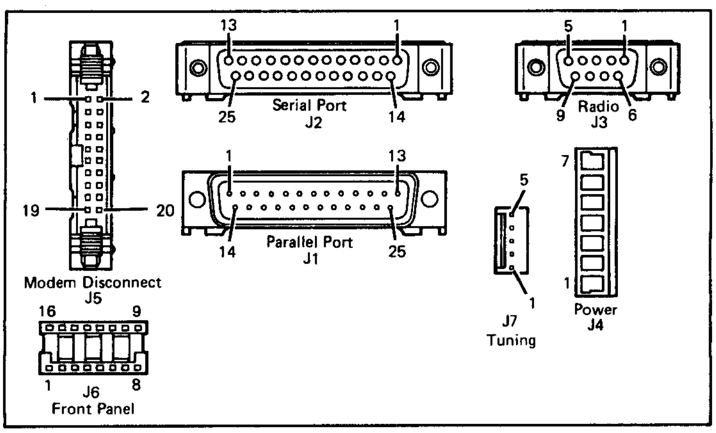


Fig. 2.1 Connector Pin Numbering

The 5-volt regulator, U24, may be mounted either on the TNC board with the heat sink provided or off board. If it is mounted off board, the regulator must be in good thermal contact with an appropriate heat sink. The power cable connects the board to the custom transformer secondary windings and to the 5-volt regulator if it is off board.

The power connector, J4, should be attached to the mating connector so that the nylon locking tab and the "wall" of the on-board half are touching. Make sure the connector is attached properly before applying power to the transformer.

WARNING! Be sure that there are no exposed power connector pins (i.e. misalignment of J4 and its mating connector) or severe damage to your TNC may result!

Jumper and Switch Settings

The onboard jumpers and DIP switches are described in Chapter V. For initial power-up of the TNC, the following switch configurations should be used:

Switch 1	ON
Switch 2	ON
Switch 3	OFF
Switch 4	ON

Switch 3 is the Reset switch. Toggling this switch ON then OFF will perform a hard reset of the TNC microprocessor.

The following jumper configurations should be used:

JP1	ON	JP6	left position
JP2	OFF	JP7	right position
JP3	OFF	JP8	left position
JP4	ON		
JP5	OFF		

Terminal Interfacing

Terminal interfacing is provided via J2, an RS-232C compatible connector. A detailed discussion of the RS-232C communication standard and the pinout of this connector are given in Chapter V. If you construct a cable to connect your computer or terminal to your TNC you should refer to this discussion. The TNC connector is wired exactly the same as that on a modem, so if you already have a cable which connects your terminal to a modem, you may use that cable.

Radio Interfacing

The TAPR TNC is designed for easy interfacing to a station transceiver or transmitter and receiver. All radio input and output (I/O) is handled through a single 9-pin connector, J3. A mating connector is provided with the TNC.

In general, the easiest transceivers to interface will have all return lines to negative ground, separate microphone and PTT input lines and clean audio sections with no spurious audio tones or audio harmonic distortion. The audio response resistor-programmed into the TNC modem input filter (U28) should be adequate for most receivers, but there may be a few "hard case" units. This is one reason that the input filter is configured via a 16-pin DIP header resistor pack (U3Ø). Note that, while simpler input coupling methods do exist and have been tried during the development of the TNC, the extensive filtering provided on this TNC permits operation in the presence of higher background noise than is possible with simple passive filters.

Due to the wide variety of audio input and output connectors in common Amateur Radio use, TAPR cannot provide a prewired cable for mating to your particular radio. However, detailed connection information for a number of popular transceivers is compiled in Appendix D and we suggest you look for your particular radio there. If your radio isn't listed in the Appendix, we request that you fill out the RADIO INTERFACING DATA page, located at the end of this chapter, and send a copy to TAPR. The following explanation of the connector pinout should be helpful to you in working out your own radio interface.

Radio Interface Pinout

Audio Input (pin 3):

This lead brings the audio from the radio to the TNC. It is normally connected to the headphone or speaker audio output of the receiver. The TNC input impedance as normally configured, is about 1Øk ohms, bypassed by Ø.Ø1 uF. If your receiver requires a lower impedance load, change C26 to 1Ø uF and add the load at R59. Most radios do not require this even though they normally drive a low impedance speaker.

If the radio's speaker is also connected, be alert to possible distortion it may introduce, particularly at high drive levels.

Common (pins 6,7,8 and 9):

These pins are tied directly to the negative return bus on the TNC and should be connected to the negative return bus on the radio. This common serves as a return for transmit control (PTT) and both send and receive audio. If a ribbon cable is used for radio connection and all four ground wires are used, every other wire in the cable should be a ground.

NOTE: Some Japanese HTs, such as the Yaesu FT-2Ø8R, use their positive supply bus as audio return. DO NOT connect this positive audio return to TNC common! Instead, the HT's negative common should be connected to the TNC. This should not cause any problems, as the radio power supply should be adequately bypassed for audio.

Transmit Key Line (PTT) (pin 4):

This lead is designed to connect to the PTT or other transmitter activation line on the attached radio. It will be connected to TNC common, or "ground" when the TNC attempts to key the transmitter. This line is rated at +3Ø volts dc open circuit maximum and Ø volts dc open circuit minimum. DO NOT connect it to a negative potential with respect to common. It can safely sink up to 25Ø mA in the closed (transmitter keyed) configuration. If the above ratings of the transmitter key line are exceeded, extensive damage may occur to your TNC. If your radio fails to meet the above criteria, a relay or other means of keying should be used.

The VFET used to key the transmitter has been very successful at operating transmitters previously thought to require a relay contact to properly key.

KNOW YOUR RADIO PTT CHARACTERISTICS BEFORE CONNECTING THE TNC!

Transmitter Audio (pin 5):

This lead provides an audio signal to the station transmitter microphone input.

The TNC audio output is AC coupled through C19, a non-polarized Ø.1 uF 12-volt capacitor OR a 1 uF polarized electrolytic capacitor (assembly option). If your radio has a microphone input impedance below about 3ØØØ ohms, you should probably use the 1 uF capacitor and you should be alert to polarity reversal problems. The TNC side of the capacitor is at a potential of about Ø volts dc. If your input impedance is above about 3ØØØ ohms, you should use the Ø.1 uF capacitor.

NOTE: Do not attempt to use the TNC to directly drive a so-called carbon microphone input circuit as this is a low impedance input that requires considerable drive current. While damage to your TNC is not likely, excessive distortion of the transmitted signal is probable, which will result in unreadable packet transmissions.

Receiver and Transmitter Audio Level Adjustments:

Before connecting a radio to the TNC, connect the terminal and apply power. Ensure that the board has been properly initialized and that the sign-on message appears (see Chapter III, Operation).

Starting with the radio volume control at minimum, open the squelch (if applicable) and advance the volume control. Observing LEDs D7 and D8 (marked RCVR AUDIO on the silk screen), continue advancing the volume until the LEDs begin to flicker and glow from the noise. This is probably the optimum setting of the audio level to your TNC. It should be used as a starting point for final adjustment.

The transmitter audio signal level is set by R33 and should typically be about 2Ø mV peak-to-peak at the microphone connector, or about Ø.6 volts p-p (Ø.2 volts RMS) at test point TP1, located just above U31. In lieu of using an oscilloscope for this adjustment, on-the-air monitoring of your packet transmissions by other stations will generally prove adequate. For further details, consult the Calibration section of Chapter V.

RADIO INTERFACING DATA

Station (Call Sign):

Radio Manufacturer:

Model:

Microphone Connector--
Make:
Model:
Mic Audio Pin no.:
Mic Key Pin no.:
Mic Common Pin no.:

Headphone/Speaker Connector--
Make:
Model:
Audio Pin no.:
Common Pin no.:

Microphone audio drive level:

Measured open-circuit voltage from PTT line to common:

Measured "key-down" current through PTT line to common:

Schematic of any adapter needed to mate PTT and microphone audio to radio:

Copy and forward to:

Tucson Amateur Packet Radio Corp.
P.O. Box 22888
Tucson, AZ 85734

CHAPTER III

OPERATION

Introduction

In order to use your Terminal Node Controller for packet radio communication, you must attach it to a radio and to a terminal or computer. This Chapter assumes that you have followed the instructions in Chapter II to get your station hooked up. If you have gotten the TNC to type a sign-on message on your terminal, you are ready to operate. We will call the terminal or computer a "terminal" for simplicity, except when we discuss features which are peculiar to computers. You can use your computer to emulate a terminal by running a "terminal program" or a "modem program" or a "communications support program". We will use the term "terminal program" to refer to this type of program.

A TAPR TNC running the TAPR/AMSAT AX.25 software has several operating modes. When the TNC is turned on or reset, it will be in 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 carriage-return (<cr>). 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 Command Mode, you will be able to see them, but to send packets you must instruct the TNC to enter a data mode. Two data modes are provided: Converse Mode, and Transparent Mode. In these modes, the information you type to the TNC is assembled into packets and transmitted on the radio.

In addition to these operating modes, a special Debug Mode is provided which allows the user to examine and change memory locations. This mode is primarily of use to the software developers, and to people trying to track down obscure hardware faults. A list of Debug Mode commands is given in Appendix E for those who are interested.

The remainder of this Chapter describes 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 they might be used. An alphabetical catalog of commands which describes the format and parameters of each is given in Chapter IV.

Terminal Characteristics

Most terminals use serial communications, in which each bit of a 7- or 8-bit character is sent in sequence over the same wire. Serial communications are usually described as being variations of the EIA standard called RS-232C. Serial data must be transmitted at a predetermined bit-per-second rate (baud). There are many standard rates, and unless the TNC and the terminal are using the same rate, they will not be able to communicate. The TNC supports all standard rates from 50 baud to 19,200 baud. Baud rates are selected from a table using the command ABAUD.

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 for best results. These are word length, parity, and number of stop bits. These characteristics are set using the commands AWLEN, ABIT, and PARITY. Serial data may represent ASCII data, which has 7 data bits per character, or binary data, which usually has 8 data bits per byte. Unless you are operating in 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.

Data may include a parity bit, which helps to detect data errors if there is noise on the data line. Parity may be even or odd. Even parity means that the parity bit will be set to a 1 or 0 to make the total number of 1s in the word even. The parity bit for odd parity is the reverse of that for even parity. Many terminals have an option of "no parity" which may mean different things for different manufacturers. This should mean that there is no parity bit included in the word, which would make the transmission one bit short. However, it may mean that the parity bit is there but not used for checking, and is always set to the same thing -- 0 (space parity) or 1 (mark parity). The TNC never checks the parity bit, but it can transmit any of these parity options for compatibility with any terminal. There are some restrictions on parity and stop-bit combinations.

The beginning and end of a word or character are marked by start and stop bits. All serial data has one start bit and one or two stop bits. The stop bits are the minimum interval between successive characters. Usually, the only terminals which require two stop bits are old-style slow (110 baud) teletypes. The TNC does not support the combination of 8 bits, parity, and 2 stop bits.

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

If you start up the TNC in the default-parameter mode, with Switch 1 set, the serial port will be initialized at 300 baud, space parity, 7-bit characters, and one stop bit. A message will be typed at 300 baud, and if an * (asterisk) is entered before the message is completed, the TNC will sign on at 300 baud. If this does not occur, an autobaud routine will start. If the user types *s at one of the baud rates supported by the autobaud routine, the TNC will eventually sign on at that rate. When the TNC signs on, ABAUD will correspond to the result of the autobaud routine. The other terminal attributes are not affected by the autobaud routine.

Getting Started

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

```
cmd:
```

The first thing you should do when you see this prompt is to set your call sign and save it. To do this, type

```
MYCALL WB9FLW
```

using your own call, of course. The TNC will respond with

```
was
```

indicating that the call sign was blank. Save this information by typing

```
PERM
```

This writes your call and the terminal baud rate into the NOVRAM so that they will be set automatically the next time you power your TNC up. You should move Switch 1 to the OFF position so that the TNC will use the NOVRAM information instead of the default values.

Once you have set your call sign, you are ready to try sending a packet. First, you should type the command CONVERS to enter Converse Mode. Now type a message to be transmitted as a packet:

Hello World!

End your message with a <cr>. If you watch the LEDs on the TNC as you do this, you should see several things happen. The PTT light will come on, indicating that the transmitter is being keyed. The TxD light will flicker on and off. If this is the first packet you have transmitted, and if CWID is ON, you will see the CWID light blink on and off as your call sign is transmitted in FSK. When the transmission stops, the PTT light will be off, and if a Morse Code ID was sent, the CWID light will be off. Don't worry 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.

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.

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 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 NK6K, for example, return to Command Mode by typing <ctl-C> (or the character you have specified with COMMAND) and type

CONNECT NK6K

(The symbol <ctl-C> means the character produced by typing "C" while simultaneously pressing the "control" key.) If NK6K is on the air, tuned to your frequency, and within range of your transmission, you should notice a message coming back to your TNC. If you have your radio attached to a speaker as well as the TNC, you will hear the packets;

otherwise, you can see the DCD LED light up, and you may see the two incoming-data lights flicker. When your connect request (the packet your TNC sent) has been acknowledged, the TNC will display the message

```
*** CONNECTED to NK6K
```

and automatically move to Converse Mode. If you now type a message, it will be formed into a packet and sent to NK6K. When you are through with the conversation, either you or the operator of the other station may initiate a disconnect. To do this, return to Command Mode and type the command

```
DISCONNE
```

After an exchange of packets, you will see the message

```
*** DISCONNECTED
```

which 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.

Now that you have gotten your packet radio station on the air, we will take a few pages to describe the TNC operation in more detail. The remainder of this chapter will help you get the most out of your TNC. You should also refer to the alphabetical list of commands in Chapter IV.

Operating Modes

Command Mode

Command Mode is used to enter commands which alter the TNC's operating parameters. All other modes (except Debug Mode) must be entered from Command Mode. When the TNC is in Command Mode, the characters, "cmd:" are 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 Command Mode after a reset or power-up. A reset can be accomplished by disconnecting power from the TNC for several seconds, by toggling the reset switch (Switch 3) on the board, or by issuing the command RESET. After a reset operation, all operating parameters of the TNC are re-initialized by the resident software.

The values of most parameters can be stored 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 the new values used upon reset, you can store them in NOVRAM with the command PERM. In order to use these values, you must make sure that Switch 1 is not set ON when the TNC is reset, as this causes the default parameters stored in the software to be loaded.

There are several ways to get to a data mode from Command Mode. You can type the command CONVERS or TRANS, depending on the data mode desired. 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.

There are two special modes which you can enter from Command Mode, Calibration Mode and Programmer Mode. These modes are actually special sub-programs. You can not send or receive packets in these modes (the packet controller chip is disabled), and you can only exit to Command Mode. Calibration Mode is discussed in detail in Chapter V. Programmer Mode is intended for using the TAPR EPROM Programmer Attachment, and is discussed in the manual for the attachment. If you accidentally get into the Programmer Mode, you will have to answer a few questions (<cr>s will do for answers) until you see the prompt "Option:". To return to Command Mode, type the letter "R" when you see this prompt.

Converse Mode

The data mode you will probably use most often for ordinary QSOs is Converse Mode. In Converse 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 command mode, there are nine characters which have special meanings to the TNC and are not ordinarily included in packets. These include input editing characters, which are discussed in a later section. Many of these special characters are control characters, which are typed by simultaneously holding down the "control" key and another key on the keyboard. The ASCII character set includes control characters corresponding to all the letters and a few other characters. As an example, a control C character is typed by holding down the control key and typing an upper or lower case letter C. This character is designated <ctl-C>. Some computers which do not have a control key, such as the Radio Shack Color Computer, have other methods of entering such characters. Also, some of the special-function keys (such as backspace and tab) may actually enter control characters. If you have no convenient way of typing control characters you may change these parameters to ordinary printing characters. Also, you should change any characters which will be used by your terminal program or computer operating system.

To get back to Command Mode from Converse Mode, you must type a special character. The default character is a <ctl-C>, and it may be changed with the COMMAND command. Typing the COMMAND character when the TNC is already in Command Mode does nothing.

Transparent Mode

An application for which packet radio is very well suited is transfer of large amounts of data between computers. For some types of data transfer operations, Converse Mode will work very well. However, you may want to send special information such as ready-to-run programs to another Amateur. 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 Converse Mode. For this type of application, you will want to use Transparent Mode. Transparent Mode is a data mode like Converse Mode, except that in this mode 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. The time intervals at which data is packetized may be changed with the PACTIME command.

The display characteristics of the TNC are also modified in Transparent Mode. Data is sent to the terminal from the TNC exactly as it is received over the radio channel, including all 8 bits of each byte received. All features such as line-feed (<lf>) and <cr> 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 Converse Mode are changed by entering Transparent Mode, and all display features are re-enabled when the TNC is returned to Command Mode. Most of the informative messages which appear in Converse Mode as the TNC moves between disconnected and connected states are also disabled.

In order to escape from Transparent Mode to Command Mode, you must follow a special procedure. After a time interval equal to PACTIME the last data typed will have been packetized for transmission (although it may not be transmitted yet). You must then wait an additional time, which is set by the command CMDTIME. Following this wait, you must type three <ctl-C>s (or whatever character is set by COMMAND) within an interval CMDTIME of each other. After a final CMDTIME interval in which no characters are typed, you will see the "cmd:" prompt. If any characters are typed 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 zero you will not be able to escape from Transparent Mode except by performing a hard reset (switch or power-down reset).

Flow Control

Whenever data is transferred to computers (home computers or TNCs), there is a chance that the data will be received faster than the computer can handle it. Some programs try to prevent this problem by providing data buffers for storing incoming data until the program is ready for it. However, this merely postpones the problem, since there is a finite amount of room in any buffer. 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 which are supported by the TNC. XON/XOFF flow control, sometimes called "software flow control," is accomplished by sending a special character (usually a <ctl-S>) to request that the output stop and another special character (usually a <ctl-Q>) to restart output. Hardware flow control may be used if both computers use the Request To Send (RTS) and Clear To Send (CTS) lines of the RS-232C standard.

The single greatest difficulty experienced during Beta Test was correct implementation of flow control on the user's computer. Many inexpensive and commonly used terminal programs and file transfer programs for home computers do not implement flow control in software, and many "RS-232C" ports do not support hardware flow control. Even if the RTS and CTS lines appear at the connector, software which 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, you should 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 should use XON/XOFF flow control. This method of flow control is enabled by setting XFLOW ON. The special flow control characters are set to <ctl-S> and <ctl-Q> by default, but they may be changed. The commands XON and XOFF select the characters which will be sent <u>to</u> the terminal <u>by</u> the TNC, and the commands START and STOP select the characters which should be sent <u>to</u> the TNC <u>by</u> the terminal. If you set the codes for these characters to zero, you will disable that function. If you set START or XON to zero, STOP or XOFF will be automatically disabled as well.

The TNC's input buffer may fill up in Command Mode if you try to type too long a command. In data mode, the buffer may fill up if you are using your computer to transfer data at a rate 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 person or computer at the other end has stopped 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 entirely, data will be lost. When the buffer empties out, the TNC will send a single XON character to the terminal.

A computer file transfer program can very easily be unable to process data fast enough to keep up with output from the TNC. In order to be sure of reading every character, a computer must respond to "interrupts" from its I/O devices. Some simple programs, especially those written in BASIC, may "poll" the input register for new data. If the polling is not done often enough, data may be lost. If the program enters a routine which will not allow it to check for data regularly, it should send a STOP character to the TNC. Some disk operating systems, such as Apple DOS, disable all interrupts when they access the disk. The program should always send a STOP character before a disk access. If you send a STOP (START) character to the TNC when it is already stopped (started), the character will be treated as data. If you have enabled echoes, and the STOP or START character is echoed, this may have undesired consequences (a control-S will lock some keyboards). If the STOP and START character are the same character, this character will "toggle" the output, turning it off if it is on, and on if it is off.

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

If you are using a terminal or terminal-emulating program, you may want to set XON and XOFF to something you can respond to, such as <ctl-G> (bell) or a printing character that you don't use.

XON/XOFF flow control is normally disabled in transparent mode, since all characters are treated as data. 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 Chapter V.

Display Options

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

In 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 Converse Mode.

A common occurrence when two stations are exchanging packets is for incoming packets to arrive when the user 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 (normally <ctl-R>), 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 Ø. The TNC does not carefully distinguish between printing and non-printing 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, a user terminates a line of typing with a single character, usually a <cr> (called 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 a <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.

A few terminals and many home computer monitors do not display lower case characters. Most of these terminals convert all characters to upper case for display, but some of them may use lower case character codes for special control functions. If you have such a terminal, you should disable lower case display with LCOK OFF. This does not affect the way echoed characters are displayed. This feature can be used to easily distinguish your outgoing messages from incoming packets on the screen. If you disable lower case display, incoming data will be converted to upper case, while your own messages will appear in lower case if you type them that way.

The conventional response to character deletion on display terminals is to output to the terminal a backspace, space, backspace sequence. This removes the character from the screen and leaves the cursor ready to type a new character in its place. On a hard-copy 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. The redisplay-line character can be used to see the corrected line.

The <escape> character (ASCII code 27 or hex $1B) is used by many terminal to control cursor movement and special display modes. (Note: throughout this manual the dollar sign symbol prefaces all hexadecimal numbers.) If this effect is not desired, <escape> translation should be enabled with ESCAPE OFF. This will cause all <escape> characters to be sent to the terminal as the dollar sign symbol ($). This does not affect <escape> characters which are transmitted in packets.

Some terminals echo characters typed in locally, before transmitting the character 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. Echo mode can be disabled with ECHO OFF.

A few terminals require particularly long times to respond to <cr>s or <lf>. Some hardcopy terminals, including old Silent 7ØØ 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 Ø, 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. Instead, the TNC waits the specified number of character time intervals before transmitting more data. 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 Transparent Mode or if timed packets are in effect in Convers Mode, no text characters are interpreted by the TNC until it receives a <cr> or the send-packet character (in Converse Mode). Until then, you can delete and retype characters or cancel the line completely.

Editing characters would ordinarily be chosen to be control characters or other non-printing characters. Editing functions can be disabled by setting the character used for the function to $ØØ. This prevents any character (even a null) from being matched to that function.

The key usually used to remove the last character on a line may be either a <delete> character (hex $7F) or a backspace (hex $Ø8 or <ctl-H>). The character used is determined by the DELETE command, which has options ON (<delete>) and OFF (backspace). The key used for rub out on your terminal may be labelled "back space", "delete", "rub out", or simply "<--". You may have to experiment to find out whether it produces a backspace or a <delete> character.

Attempts to delete past the beginning of a line, or, in Converse Mode, past the beginning of a packet, will have no effect. You can not delete a <cr> or any of the special characters which are 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 wrong characters, and then spacing forward with an arrow key. The TNC does not have this sort of input editing feature -- you must delete all the characters back to the ones you want to change and then retype the rest of the line.

If you want to cancel an entire line, text back to the last-occurring <cr> can be deleted by typing the character specified by the CANLINE command (default <ctl-X>). In Converse Mode, this character does the same thing as the cancel-packet character unless you have set the send-packet character to something other than a <cr>.

In Converse Mode, text back to the last-occurring send-packet character is deleted by typing the character specified by the CANPAC command (default <ctl-Y>). This will delete any intervening <cr>s. Packets which have already been placed in the outgoing packet buffer can not be cancelled.

The cancel-packet character has a special function in 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 (otherwise known as the Bit Pit). Normal output can be resumed by typing another cancel-packet character or by changing modes, e.g., going into Converse Mode. Echoing of input is not affected by this command. This feature makes it possible to get rid of unwanted output which may occur. For example, the TRACE command can very quickly produce enormous amounts of output which, because of the large TNC buffers, may not all be typed until long after the function has been disabled.

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 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 <ctl-V>). Any character (including non-special characters and the pass character) can be sent this way. 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 -- you must type it again for each such character.

The pass character can be used in Command Mode in order to include <cr>s in the beacon text.

Packet Operation

The primary function of the TNC is to enable you to communicate via packet radio with other Amateurs. The TAPR TNC implements two protocols, or 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, although they can be used to simulate the common Amateur net or roundtable type of contact. The protocol used by the TNC can be specified as AX.25 with the command AX25 ON, or as VADCG with the command AX25 OFF. Packets heard by the TNC not corresponding to the protocol selected will be ignored.

Earlier in this Chapter you found out 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 want to talk to. To prepare your TNC for operation, the first thing you should do is to establish your call sign as the station address using the command MYCALL. This sets the string used to identify packets transmitted by your station. If you will be operating using VADCG protocol, you must also set your VADCG address 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. The VADCG address of your TNC must be coordinated locally to insure that there will be no conflicts. 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 Ø to 15. This number is appended with a dash, as in this example:

```
MYCALL W3IWI-3
```

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

The call sign specified by MYCALL is ordinarily used by the TNC for Morse Code identification, which is sent automatically every 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, you should be aware that the call sign address within the packets is actually bit-shifted ASCII (occupying bits 1-7 of the byte rather than bits Ø-6). Also, if you use the VADCG protocol, your call sign is included only in the initial packets establishing a connection. TAPR recommends that 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, and also improves relations with non-packet operators using the same frequencies. The automatic Morse Code ID can be disabled by 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 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 want to talk to is a little too far away for you to contact directly, you can use the <u>digipeating</u> feature of the TNC. This feature works differently depending on whether the AX.25 protocol or VADCG protocol is used. A digipeater accomplishes much the same thing 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 achieves 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 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 WA7GXD VIA NØADI-2,KV7B
```

You should list the intermediate stations in the order in which you want them to relay the packets as they go <u>from</u> your station <u>to</u> the destination station. In this example, your connect message to WA7GXD will be repeated by NØADI-2 and then by KV7B. Reply packets from WA7GXD will be relayed first by KV7B and then by NØADI-2.

You can specify as many as eight intermediate stations; however, bear in mind that using more than one digipeater is a

TAPR extension to AX.25 and may not be compatible with other implementations of this protocol. In particular, the TAPR Beta Test TNCs were released with a software version which supports up to three intermediate stations. These TNCs will not function properly with their original software if other stations in the area send packets addressing more than three relay stations. The delay between your transmission and the receipt of a reply will naturally increase as more intermediate relays are used, and the likelihood of losing information due to interference or noise on the channel also increases.

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

    UNPROTO QST VIA NK6K

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 NK6K.

To request digipeating under VADCG protocol, give the command

    VRPT ON

This causes your packets to be repeated by <u>any</u> VADCG digipeater. For a TAPR TNC to act as a VADCG digipeater, the command

    VDIGIPEA ON

must be given. 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 you absence but want to inhibit it from transmitting, set XMITOK OFF. The TNC will conduct normal operations in this condition, including formatting and "sending" packets, but will not key the transmitter. You may also want 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: KD4NL

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:

    *** KD2S 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, the TNC can be commanded to send a specific message at regular intervals. This message is called a beacon, and the function can be used 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. To transmit the beacon at, for example, 10-second intervals, give the command

    BEACON EVERY 1

The beacon function also has a transmit-after mode, in which a beacon packet is only transmitted after activity is heard on the channel. This feature has been used 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 below) he 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 which are related to the timing parameters are discussed here as well.

Amateur radio equipment varies greatly in the time delays required in switching from receive to transmit and from transmit to receive. 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 greater keyup delay than 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 transmitting station has been unkeyed, 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 the TXDELAY time 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 your radio is exceptionally slow 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 <u>re</u>transmission.

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 be lost increases drastically. To help alleviate this problem, an automatic wait time can be imposed on any station <u>not</u> transmitting a digipeated packet. Any station which <u>is</u> ready to transmit a packet immediately after the carrier drops is required to wait for this time interval <u>unless</u> 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, this parameter can be set to 0, but in any event 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 which are 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 which were not acknowledged suffered collisions with transmissions from other stations. If the random wait is used, 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 transfered. 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 ten 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 ten 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:

    KV7D>NK6K: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 himself via another station as digipeater. This station will have the disadvantage of having to see his own packets re-displayed. For example, WB6YMH, WDØETZ, WAØTTW, W1HDX, and K4NTA wish to carry on a group conversation. In order to make all the transmissions as reliable as possible, the following connections are made.

    WB6YMH connects to W1HDX
    WAØTTW connects to K4NTA
    WDØETZ connects to WDØETZ via W1HDX

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

HF and OSCAR

When configured in the default mode, the TAPR 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 the best results can be obtained 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. Turn the volume on your receiver down to the point where the LED is on full when data is present and only gives an occasional flicker 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 <u>must</u> 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 will tell 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 <u>all</u> of a packet must be received correctly. Data rates as <u>high</u> as 1200 baud (HBAUD 1200) have been used on both 10 meters and through OSCAR 10 on a Maryland to New Zealand path using unmodified Beta TNCs with signal to noise ratio of 12-15 db.

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. Modem parameters and timing constants can be changed 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 Chapter V.

The audio frequency response of the TNC in combination with a 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 56Ø ohm load resistor (R59) you should not install it, as it degrades the low frequency response. The input filtering provided by the MF-1Ø switched capacitor filter can be modified by changing the values of the components on the dip header U3Ø. Various alternate filter configurations are listed from time to time in TAPR's newsletter, PSR.

The onboard modem can be completely bypassed using J5. This allows use of exotic modulation methods and higher baud rates. The interfaces available on J5 are TTL levels and not RS-232C. Be sure to refer to the hardware section for more information.

CHAPTER IV

COMMANDS AND MESSAGES

Command Syntax

There are many variable parameters which are used by the TNC in its operation, such as the user's call sign, terminal type, display preferences, and the characteristics of his radio. In addition, there are several actions the TNC can be commanded to perform, such as connecting to another station to start a conversation, disconnecting at the end of the QSO, saving information in NOVRAM, or displaying information about itself. The user changes parameters and issues instructions to the TNC by typing commands composed of English-like words or word abbreviations, which are called keywords, and variables which are numbers or strings of characters chosen by the user. You will probably never change some of these parameters; however, one of TAPR's goals is to allow each user maximum flexibility to adapt the TNC to his environment.

All commands are listed alphabetically below. If a command has parameters, each parameter is described and the default values are given. The defaults are the values stored in EPROM which may be loaded instead of those in NOVRAM by setting Switch 1 on prior to performing a reset. Those parameters which can be saved in NOVRAM 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 brief discussion of the command is then given. A more detailed discussion of many of the commands and their interrelationships is given in Chapter III. A command is entered 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 when a carriage return (<cr>) is typed. All keywords may be entered in upper or lower case. Except for the beacon and ID text string, everything entered in Command Mode is translated to upper case before being examined. All commands and alphabetic parameters may be abbreviated to the shortest unique string. These minimum abbreviations are shown underlined in the commands section of this chapter.

There are several parameter types. A parameter which is 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 hex. A hex number is distinguished from a decimal number by preceeding it with a "$" prefix. The "digits" of a hex number represent powers of 16, analogous to the powers of 1Ø represented by a decimal number. The numbers 1Ø through 15 are denoted by the hex digits A through F. For example,

```
$1B = 1*16 + 11 = 27
$12Ø = 1*16*16 + 2*16 + Ø = 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 YES and NO may be typed instead. A few parameters are really flags, but rather than indicating that something is "on" or "off", they select one of two ways of doing things. 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 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", which is a decimal number from Ø to 15 used to distinguish two or more stations on the air with the same Amateur call (such as a base station and a repeater). The call sign and extension are entered and displayed as call-ext, e.g., KØPFX-3. If the extension is not entered, it is set to -Ø, and extensions of -Ø 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 as default values control characters. Control characters are entered by holding down a special control key on the keyboard while typing the indicated key.

There are two commands, BTEXT and IDTEXT, which have as parameters a text string. 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 carriage returns into the string by preceding them with the "pass" character. The string ends when you type a (non-passed) carriage return.

In the command descriptions below, the keywords are shown in upper case. User-supplied values are shown in lower case. If a parameter must be chosen from one of 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, the user can optionally specify either C or D.

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

Commands

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:

Ø Reserved for future use.

5Ø		18ØØ	
75		24ØØ	
11Ø	#	36ØØ	
135		48ØØ	#
15Ø		72ØØ	*
3ØØ	#	96ØØ	#*
6ØØ		192ØØ	*
12ØØ	#		

- 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 performing input or output at these baud rates. Use of baud rates higher than 48ØØ is not recommended for computer file-transfer applications.

The baud rate change will not take effect until a RESET command is issued. The TNC enters an autobaud routine and attempts to match the terminal baud rate if Switch 1 is used to select default parameters. If the TNC is initialized with default parameters, the autobaud 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. The number of stop bits will not change until a reset is performed.

AUTOLF ON|OFF default: ON *

parameters

ON A <lf> is sent to the terminal after each <cr>.

OFF A <lf> is not sent to the terminal after each <cr>.

This option should be ON when a terminal is used that does not automatically send a linefeed after a carriage return. If AUTOLF is ON, a <lf> is sent to the terminal after <cr>s in received packets as well as echoed <cr>s received from the terminal. This command only affects what is displayed, not the data sent in packets.

AWLEN n 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 affect until a reset is performed. Except in Transparent Mode, only the low order 7 bits of a word are kept. To use all 8 data bits in Transparent Mode, AWLEN must be set to 8.

AX25 ON|OFF default: ON *

parameters

ON The TNC operates in AX.25 protocol.

OFF The TNC operates in VADCG protocol.

If ON is specified, the TNC will originate packets and other link operations in 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: Ø *

parameters

n Ø - 15, specifying 12Ø 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 RF on the air. AXDELAY can also be used when the receiving TNC's rig has a very slow PLL or squelch.

AXHANG n default: Ø *

parameters

n Ø - 15, specifying 12Ø ms intervals.

This value can be used to increase channel utilization when an audio repeater with a hang time greater than 12Ø ms is used. If the repeater squelch tail is long, it is not necessary to wait for AXDELAY after keying the transmitter if the repeater is still transmitting. If the TNC has heard a packet sent within the AXHANG period, it will not add AXDELAY to the keyup time.

BEACON [EVERY|AFTER] n default: EVERY Ø *

parameters

EVERY Send beacon every specified interval.

AFTER Send beacon once after the specified interval with no link activity.

n Ø - 255, specifying 1Ø second intervals.

Beacon mode is turned on by this command. n specifies a time in multiples of 1Ø seconds. A value of Ø for n turns the beacon off. If the optional keyword EVERY is used, a beacon packet is sent every n*10 seconds. If AFTER is used, 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. This mode can be used 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 ON and MTO BEACON.

BKONDEL ON|OFF   default: ON *

parameters

ON   The sequence backspace-space-backspace is echoed when the DELETE character is entered.

OFF The backslash character (\) is echoed when the DELETE character is entered.

This command determines the way the display is updated to reflect a character delete in 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:

TAPR/AMSAT AX.25 level 2 protocol software version X.Y

where X.Y represents the software version number. The <cr> character can be included in the text by using the pass character (default is <ctl-V>) preceding <cr>. A maximum of 128 characters can be specified. The beacon text is not stored in NOVRAM.

CALIBRA

The CALIBRA command is used to transfer control to the hardware calibration routine. The commands available from this routine are described in the modem calibration section of Chapter V. Calibration may be performed at any time without altering the current link state.

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

parameters

n Ø - $7F, specifying an ASCII character.

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

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

parameters

n Ø - $7F, specifying an ASCII character.

This command is used to change the cancel packet terminal editing command character.

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

CMDTIME n default: 1 *

parameters

n Ø - 15, specifying 1 second intervals.

This command sets the Transparent Mode timeout value. In order to allow escape to Command Mode from Transparent Mode while permitting any character to be sent as data, a guard time of CMDTIME seconds is set up. Three COMMAND characters must be entered within CMDTIME of each other, with no intervening characters, after a delay of CMDTIME since the last characters were typed. After a final delay of CMDTIME, the TNC will exit

Transparent Mode and enter Command Mode. You should then see the prompt

cmd:

If CMDTIME is zero, the only exit from Transparent Mode is a hardware reset.

COMMAND n default: $Ø3 <ctl-C> *

parameters

n Ø - $7F, specifying an ASCII character.

This command is used to change the Command Mode entry character. Command Mode is entered from Converse Mode when this character is entered from the terminal. See Chapter III for information on how the command character is used in Transparent Mode.

CONMODE CONVERS|TRANS default: CONVERS *

parameters

CONVERS Connects cause automatic entry to Converse Mode.

TRANS Connects cause automatic entry to 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 parameter is given with the CONNECT command, the CONMODE parameter will not be used. If the TNC is already in 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. As many as eight digipeat addresses can be specified.

CONVERS Enter Converse Mode upon successful connect.

TRANS Enter 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 no response to the connect request occurs after RETRY attempts, the command is aborted, a message is typed, and the TNC remains in Command Mode.

If the optional mode parameter is not specified, the mode given in CONMODE in 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. Digipeating by specific stations can not be specified in the VADCG protocol.

Example (AX25 is ON):

CONNECT WA7GXD VIA NØADI-1,WDØETZ CONVERS

This commands the TNC to initiate a connect request to WA7GXD, with the connect packets and subsequent data packets to be digipeated through NØADI-1 followed by

WDØETZ. Packets sent in the opposite direction access the digipeaters in the opposite order. Thus, packets from WA7GXD will first be repeated by WDØETZ, then by NØADI-1.

WARNING: Use of more than one digipeater is a TAPR 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 though the radio. ON will result in the request being ACKed, the standard connect message will be output through the terminal, and the data transfer mode specified by CONMode will be entered.

OFF will cause the message "connect request: <call>" to be output to the terminal. The user may then enter Command Mode and issue his own connect command.

Example of OFF:

*** connect request: KD4NL

<ctl-C>

CONNECT KD4NL

*** CONNECTED TO KD4NL

In this example, the TNC is assumed to be in Converse Mode, but not currently connected to anyone. After receipt of the message, the user enters Command Mode and issues 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 Transparent Mode.

CONVERS

CONVERS has no options. It is an immediate command, and will cause exit from Command Mode into Converse Mode. Any link connections are not affected.

CPACTIME ON|OFF default: OFF *

parameters

ON PACTIME is used in Converse Mode.

OFF PACTIME is not used in Converse Mode.

When CPACTIME is ON, the PACTIME parameter is used in Converse Mode as well as in Transparent Mode. This mode is normally used when a computer is attached to the TNC on the other end of the link, but full Transparent Mode is not desired. In this mode, characters are sent periodically as in Transparent Mode, but the local editing and echoing features of 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 default: ON *

parameters

ON The SENDPAC character, normally <cr>, is appended to all packets sent in Converse Mode.

OFF The SENDPAC character is not appended to packets.

When CR is ON, all packets sent in Converse Mode will include the SENDPAC character which forces the packet to be sent. Setting CR ON and SENDPAC $ØD results in a natural conversation mode. Each line is sent when a <cr> is entered, and arrives at its destination with a <cr> 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: $Ø5 <ctl-E> *

parameters

n Ø - $7F, specifying an ASCII character.

This command is used to change the debug program entry character. When that character is entered in Command or Converse Mode, the resident debugger is entered. The debugger is described in Appendix E.

DELETE ON|OFF default: ON *

parameters

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

OFF The delete character input editing character is <backspace> ($Ø8).

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

DISPLAY [class]

parameters

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

Display will cause all control parameters and their current values to be displayed. Individual parameters can be displayed by entering the parameter name with no options. Groups of related parameters can be displayed 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 keyup 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 Command or Converse Mode. Local echoing is disabled in Transparent Mode.

ESCAPE ON|OFF default: OFF *

parameters

ON The <escape> character ($1B) is output as "$" ($24).

OFF The <escape> character is output as <escape> ($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 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 Converse Mode) or a line is completed (in 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 Transparent Mode.

FLOW will keep received data from interfering with data entry. See also the comments on the CR command.

FRACK n default: 4 *

parameters

n 1 - 16, specifying 1 second intervals.

After transmitting a packet requiring acknowledgment, the TNC will wait FRACK seconds before incrementing the retry counter and sending it again. If the retry count specified by RETRY is exceed, 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.

Ø Reserved for future use.

5Ø	6ØØ
75	12ØØ
11Ø	18ØØ *
135	24ØØ *
15Ø	36ØØ *
3ØØ	48ØØ *
4ØØ	

* - These baud rates are not supported by the on-board Bell-2Ø2 compatible modem.

ID

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

IDTEXT will be used for the ID if it has been entered; 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. See Appendix A for valid CW characters.

LCOK ON|OFF default: ON *

parameters

ON The terminal is capable of receiving lower case ASCII characters.

OFF The terminal is not capable of receiving lower case ASCII characters.

If LCOK is OFF, lower case characters will be translated to upper case before being output to the terminal. This case translation is disabled in Transparent Mode. Input characters and echoes are not case translated.

LFADD ON|OFF   default: OFF *

parameters

ON A <lf> character is added to outgoing packets following each <cr> transmitted in the packet.

OFF No <lf> is added to outgoing packets.

This function is similar to AUTOLF, except that the <lf> characters are added to outgoing packets rather than to 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 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 all of which are unconnected.

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

MAXFRAME n 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 contiguous 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 unACKed does not exceed MAXFRAME.

MCON ON|OFF default: OFF *

parameters

ON Monitor mode remains active when TNC is connected.

OFF 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 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.

There are two special calls. If either is used, 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 of the FROM field. A packet is displayed if it appears in either the MTO list or the MFROM list. This means that if either list is ALL then all packets will be displayed.

** If anything other than NONE is specified, 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 monitor mode is on, and the TNC is not in Transparent Mode, packets not addressed to this TNC may be displayed. The addresses in the packet are displayed along with the data portion of the packet, e.g.:

KV7D>W5FD-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 monitor mode when the TNC is connected. All monitor functions are disabled in Transparent Mode. To completely enable monitor mode, a TO or FROM list must be specified 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 monitor. If MONITOR is on, any packet heard which has as its TO address any of the calls in the MTO list will be displayed. 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 NONE 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 Ø - 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 ID unless another string is specified by IDTEXT.

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

MYVADR n default: 31 *

parameters

n Ø - 31, signifying a VADCG address byte.

This command selects the address used when the board is operating in the VADCG mode. The address is translated to the "to be digipeated" range by the VRPT command. Choice of VADCG addresses should be coordinated 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 <lf> characters.

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

This command enables a transmission delay following any <lf> 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, NUCR and NULF must be set 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 Converse and Command Modes. If an odd number is specified, the number 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 Converse and Transparent Modes.

WARNING: Allowing more than 128 characters of data is a TAPR 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 Ø - 15, specifies 1/4 second intervals.

EVERY Timeout occurs every n seconds.

AFTER Timeout occurs after n seconds with no other input.

This parameter is always used in Transparent Mode, and will also be used in 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 Ø - 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
Ø	odd
1	even
2	mark
3	space
4	none

If PARITY choices Ø-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 7 data bits and a parity bit, all 8 bits can be transmitted in Transparent Mode by setting AWLEN 8 and PARITY 4.

PASS n default $16 <ctl-V> *

parameter

n Ø - $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. As this process can not be undone by turning the TNC off, care should be taken to see that the correct values have been selected.

Each bit of the NOVRAM chip can be reversed by the burning procedure a minimum of 1Ø,ØØØ 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 accidentally enter this routine, type <cr> until you see the Option: prompt. Exit by typing R followed by a <cr>. Then, following the next prompt, enter any character.

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

parameters

n   Ø - $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 PERMed are retained if Switch 1 is OFF. PERMed values are restored via a hardware reset (toggling DIP switch 3) or power off/on.

RETRY n default: 1Ø *

parameter

n   Ø - 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 re-transmitted RETRY times before the operation is aborted. The time between retries is specified by FRACK. A value of zero 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 Transparent Mode). See also the FRACK command.

SCREENL n   default: 8Ø *

parameters

n   Ø - 255, specifying the screen or platen width of the terminal.

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

SENDPAC n   default: $ØD <cr> *

parameters

n   Ø - $7F, specifying an ASCII character.

Selects the character that will force a packet to be sent when entered in Converse Mode.

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

parameters

n   Ø - $7F, specifying an ASCII character.

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

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

parameters

n   Ø - $7F, specifying an ASCII character.

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

TRACE n default: $1ØØØ

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. The frame dump output is described in Appendix E.

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 Command Mode into Transparent Mode. The current link state is not affected.

TXDELAY n default: 4 *

parameters

n 1 - 16, specifying 4Ø ms intervals.

This value tells the TNC how long to wait after keying up the transmitter before sending data. Some startup 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 don't need much time, synthesized rigs need time for PLL lockup, and rigs with mechanical T/R relays will need time for physical movement. The correct value for a particular rig should be determined 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 Transparent Mode.

OFF The XFLOW parameter is ignored in Transparent Mode.

When ON, XFLOW is used to determine the type of flow control used during 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. See also 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 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 board remain the same.

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

parameters

n Ø 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 <ctl-Q> *

parameters

n Ø 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.

Tucson Amateur Packet Radio Corporation
TAPR/AMSAT AX.25 level 2 version X.Y

> This message appears when you have performed a reset with DIP switch 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.

TAPR packet radio

> This sign-on message appears when a reset has been performed with DIP switch 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 (hex number) length in bytes. If you have an 8k RAM 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 in the U8 socket.

HDLC can't init

> This is a hardware diagnostic that will appear at the time of reset if the HDLC (U17) chip can not be commanded. It indicates difficulties with either the WD-1933 chip or the 6522.

PIA can't init

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

UART can't init

This message doesn't ever appear -- if the UART can't init, there is nothing to type on. If you see LEDs D1 and D2 blinking every couple seconds when the TNC should be signing on, this indicates problems with the 6551 (U14) UART.

Messages displayed in response to Command Mode input are discussed below. The message as it appears on terminals with lower case displays is 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 entered 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, e.g.:

```
C KV7B V
       $
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 parser was kept small and simple, it will sometimes change parameters before it completes parsing 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, i.e., the line was parsed as if a <cr> was entered at the $. Example:

```
MTO QST,WB9FLW K9NG
              $
Input ignored
```

The command is parsed as if it were MTO QST,WB9FLW and the K9NG 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 an attempt is made.

too many packets pending

You have given a CONVERS or TRANS command after exiting 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 has occurred. This should never happen when two TAPR TNCs are connected. An improper implementation of the AX.25 protocol could cause this state to be entered. The TNC will attempt to re-synchronize 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.

CHAPTER V

HARDWARE

Overview

This discussion of the TNC hardware is intended to show the functional interrelationships of the major components of the TNC. Please refer to the schematic and block diagrams in Appendix A for the following discussion.

For the purpose of this discussion, the TNC will be considered as eight distinct subsystems. These are the Central Processing Unit (CPU), the system clock, serial Input-Output (I/O), memory (including Random Access Memory (RAM), Read Only Memory (ROM), and NOnVolatile Random Access Memory (NOVRAM)), the High level Data Link Controller (HDLC), the MOdulator/DEModulator (MODEM), the Versatile Interface Adapter (VIA), and finally, the parallel user port.

TNC Specifications

Processor 68Ø9

Clocks
: Processor master clock input frequency: 3.6864 MHz jumper selectable to 7.3728 MHz (requires higher speed parts). If you wish to use this option, contact TAPR for software alterations.

  UART clock: 1.8432 MHz.

  MF-1Ø switched-capacitor filter clock: 115.2 kHz.

  System E and Q clocks: 921.6 kHz standard, 1.8432 MHz with high speed option.

Memory
: All memory in industry-standard Byte-Wide sockets.

  Standard complement of ROM: 32k = 4 x 2764

  Standard complement of RAM: 8k = 1 x 6264

Modulator: EXAR 2206 modulator circuit plus related components to produce up to 1200 baud 1200 Hz/2200 Hz coherent FSK.

All important passive modem components mounted on DIP headers for ease in substitution and modification.

Built-in modem calibration system requires no additional equipment.

An external modem may be attached via a single connector which completely bypasses the on-board modem.

Support for an external modem tuning indicator is provided via a separate connector.

NOVRAM Xicor XD2212

This electrically erasable, programmable memory is configured as two switch selectable banks of 64 bytes each, enabling two independent TNC configurations to be stored.

Protocols AX.25 and VADCG (Vancouver Amateur Digital Communications Group) with full duplex radio data link operation.

Operating Modes

Command mode: accepts commands via digital port.

Conversation mode: accepts digital data, transmits and receives packets, permits terminal editing features (character delete, line delete, input packet delete, output packet delete, redisplay input) and escape to command mode via special characters trapped by the TNC. Optional use of packet completion timer as in Transparent mode.

Transparent mode: accepts digital data, transmits packets via packet completion timer only, and receives packets. No local editing features permitted. Escape to command mode via specially timed character sequence.

Functional Description of TNC

CPU

The CPU is the heart of the TNC. It manages the operation of all the other system components. It does this by issuing commands on the control bus and addresses on the address bus so that it can send and receive data between itself and the various other system components via the data bus. It also retrieves its own instructions from ROM in the memory area in this manner. These instructions stored in ROM form the program which enables the TNC to implement the communications protocol and to communicate with the user through an I/O port.

The CPU consists of four main components. First, the "brains" of the outfit is U5, the 6809 microprocessor. This is the device that executes the instructions of the program. Second, a 74S288 programmable ROM (PROM) (U4) is the address decoder. It generates chip select signals to enable the proper device on the data bus at the proper time. Third, logic gates U3 and U36 generate the proper read, write, and memory refresh timing signals. Finally, sections D and E of inverter U1 buffer the system "E" clock for the other users of this clock signal in the system.

System Clock

The system clock generates all the proper timing signals for the CPU and its peripheral devices (serial I/O, HDLC, MODEM). The system clock consists of U1A, B, and F which make up the clock oscillator, and U2 which divides the clock oscillator frequency to generate the various different frequencies required by the clock users mentioned above.

Serial Input/Output

The serial I/O section consists of U14, a 6551 Universal Asynchronous Receiver Transmitter (UART), along with U14 and U15, which are level translators. The UART receives data in parallel format from the CPU and sends it in serial format to the terminal (or computer) for communication with the user of the TNC. The UART also receives serial data from the terminal and sends it in parallel format to the CPU on the system data bus. The level translators convert the signals from the TTL levels used by the TNC logic circuitry to the +/-12 volt levels compatible with the RS-232C interface standard.

Memory

Memory is the component of the system in which instructions and data are stored. EPROM is used for program instructions and data which need not be changed during the operation of the system or lost when power is off.

RAM is used for data which will change, such as incoming and outgoing packets and system scratch variables. The TNC has a third type of memory, the NOVRAM. It is used for storage of infrequently changed parameters such as callsign, terminal baud rate, timing parameters, etc.

HDLC

The HDLC controller, U17, is a very important and complex device which performs several useful functions for the TNC. First, it helps format data received from the CPU into packets for transmission over the air. Second, it generates and checks the Frame Check Sequence (FCS) which is used for error detection. Third, it processes formatted packets received off the air to check for errors and ship the required data to the CPU for further processing under the communications protocol. This chip unloads a major portion of the work involved in implementing the lower layers of the protocol from the CPU, and thus allows the CPU to spend its time implementing the higher levels of the protocol.

Modem

The modem consists of two main parts. The MOdulator converts the serial binary data from the HDLC controller to two tones (normally 1200 & 2200 Hz) which are used to modulate the radio carrier when the packet is actually sent out over the air. The DEModulator converts the audio tones received off the air to serial binary data for the HDLC. The modulator consists of an EXAR 2206 frequency shift keyed signal generator (U19) and an output buffer U31A. The demodulator consists of U18, U20, U28, and U31B. U18 is an EXAR 2211 phase-locked-loop FSK demodulator. U20A and U28 filter and frequency compensate the received audio for use by U18. U20B provides a clean signal to the HDLC controller for use during calibration of the modem. U31B is used to drive the LEDs which indicate proper audio level for the demodulator.

VIA

The 6522 Versatile Interface Adapter (VIA) performs several different functions. First, it is used to communicate with the NOVRAM. Second, the VIA provides a means by which the CPU can read the status of DIP switch 1. Third, it provides a programmable baud clock for the HDLC controller. Fourth, it is used to key the modem tones for the Morse code identification function. Finally, it is used to provide a system software real-time clock, used by the software in timing channel operations and other functions.

Parallel Port

Finally, we come to the parallel port. The parallel port (as you might suspect) passes parallel data between the CPU and external devices which require a parallel data format. It is currently used to interface the TAPR EPROM programmer to the TNC and to output diagnostic and informative signals from the TNC. The parallel port consists of U13, a 652Ø Peripheral Interface Adapter (PIA).

NOTE: We recommend that you have the schematic diagram available for reference during the following discussions. We have used the convention of denoting a negative-true logic level by a trailing "*" (e.g., IRQ* means IRQ is true at logic Ø). Further note that TTL logic levels are defined as greater than +2.4 volts but less than +5.25 volts for a "1" and less than +Ø.8 volts but greater than -Ø.4 volts for a "Ø".

Clock and System Timing

The clock and system timing section is concerned with the generation of signals needed for clocking the CPU and serial user port baud rate, generating the power-on reset signal for on-board Large Scale Integrated circuits (LSI), address decoding for chip selection by the CPU, and conversion of CPU control bus signals to formats required by the non-family peripheral devices (memory bank and HDLC controller). This circuitry is Shottky-TTL based.

The primary system clock is crystal-controlled at 7.3728 Mhz by Y1, using U1A and U1F (74LSØ4) biased into linear operation by R12 and R13. The clock is buffered by U1B and passed to divider U2 (74LS393), a dual 4-stage binary counter. U2 provides a 1.8432 MHz clock for UART U14 (6551), and 115.2 kHz for U28 (MF-1Ø), the modem input filter. Two taps are provided to U5 (68Ø9) via jumper JP7 for quadrature clock generation at the bus frequency. The standard tap provides a clock of 3.6864 MHz, yielding a bus frequency of 921.6 kHz. The 7.3728 MHz tap provides a bus frequency of 1.8432 MHz when using optional high-speed components to support radio link data rates in excess of 4800 baud.

The power-on reset generator timing is controlled by RC network R8-C12. After power is applied, a delay of some 50 to 200 ms occurs before the Schmitt trigger (composed of U1C, U3A, R87 and R88) fires. The positive-going output from pin 3 of U3A (74LS00) and applied to the CPU (6809) as well as being distributed by the system control bus as RES* (reset).

A system clock on the control bus is provided by buffering the E phase output from U5 (6809) pin 34 through U1E and U1D (74LS14).

The HDLC controller and system memory require a separate RD* (read) and WR* (write) control signal not directly provided by the 6809. RD* is generated by NANDing R/W* and E from U5 (6809) through gate U36B (74LS10). Similarly, WR* is generated by NANDing E with R*/W (by inverting R/W* with U3D) via U36A (74LS10). The additional input to these RD* and WR* gates is tied to system RES* to drive RD* and WR* high during reset as may be required by certain RAMs used at U7 (see memory bank, below).

Address decoding is provided by U4 (74S288), a high-speed Shottky bipolar PROM. The memory map programmed into U4 is:

Address	(HEX)	Line	Size	Function	Chips Selected
0000	1FFF	CS0*	8k	RAM	U7
2000	27FF	CS2*	2k	I/O	U6,U13,U14
2800	2FFF	CS3*	2k	HDLC	U17
4000	7FFF	CS1*	16k	RAM/ROM	U8
8000	9FFF	CS4*	8k	(EP)ROM	U9
A000	BFFF	CS5*	8k	(EP)ROM	U10
C000	DFFF	CS6*	8k	(EP)ROM	U11
E000	FFFF	CS7*	8k	(EP)ROM	U12

The system used is called incomplete address decoding, which means that an IC that requires only a few bytes of address space is allocated more than it needs, which makes it accessible at multiple addresses. The primary addresses for the various I/O chips (which are the addresses actually used in the software routines) are:

IC	Description	Address Range (Hex)
U6	6522 VIA	2040-204F
U13	6520 PIA	2020-2023
U14	6551 UART	2010-2013
U17	1933 HDLC	2800-2807

In addition, certain logic has been provided for use with so-called "smart RAMs" which use dynamic RAM technology to achieve high density memory at lower cost than static RAM of comparable capacity. Specifically, many such memories require a pulse on pin 1 telling them when it is "safe" to perform an internal refresh operation without potential conflict with access requests by the CPU (below). This line is called refresh enable (REFEN*).

REFEN* is asserted (made true -- a logic "0" in this case) when RES* is false, CS0* is false and the E clock is high. This only occurs when the TNC is not in a hardware reset cycle, the chip (U7) is not selected, and the microprocessor is far enough into its current cycle to ensure that no glitches will occur on the CS0* line.

The REFEN* circuitry design was based on preliminary (limited) information and may not work with "smart RAMs" of any particular manufacturer.

Central Processing Unit

A 6809 microprocessor acts as the system CPU. This part is readily available, multiply-sourced and widely accepted for application in dedicated-function controllers as well as general purpose low-end computing devices. It executes the logical algorithms located in system EPROM.

The processor has an internal 2-phase clock generator and provides control, address and data bus input/output for family peripheral devices. The control lines are modified for non-family devices as explained above.

The 6809 has capabilities for position-independent code and is designed to support multiple stacks, making it very efficient for executing block structured high level languages (such as Pascal, which is used to implement the VADCG and AX.25 protocols running on your TNC). Information on the 6809 instruction set and addressing modes is available from the manufacturers.

Serial User Port

The serial user port is designed to provide a full-duplex RS-232C interface for the user's terminal or personal computer. Full baud rate selection from 50 baud to 19.2 kilobaud is supported by the port. EIA RS-232C levels and transition rates are implemented as well. The on-board connector is a DB-25S to conform with the RS-232C physical interface connector recommendations for modems (or Data Communications Equipment - DCE).

The serial I/O port is controlled by U14 (6551), a complete LSI Universal Asynchronous Receiver-Transmitter (UART) with an internal, software-controlled baud-rate generator. The transmitter and receiver are doubly-buffered and capable of interrupt-driven operation. The 6551 supports hardware flow control via its CTS* (Clear To Send - for transmit) and RTS* (Request To Send - for receive) pins.

The TTL I/O levels of the 6551 are translated to and from RS-232C levels and sense by U15 (1488) and U16 (1489A) buffers. U15's output transition rate of change is limited by 330 pf capacitors (C9-C11) to ensure RS-232C compliance. D17 and D18 are incorporated to prevent power supply problems when abnormal conditions exist on the RS-232C lines.

All input lines to the TNC's RS-232C port are pulled to proper levels by 33k resistors (R1, R3 and R4). This sets the correct defaults to allow the TNC to work with so-called "2-wire" serial data ports or other RS-232C subset implementations on the attached terminal. R1 asserts CTS* so the TNC will send data to the terminal. R2 (6.8 k - also an output) asserts DCD* to allow the TNC to accept user data.

Parallel User Port

The parallel user port is designed as a general-purpose I/O port. It is currently used to output status information and debugging information (see Chapter V and Appendix E). In addition, it is designed to support accessories such as the TAPR EPROM programmer.

U13 (6520/6820/6821) is a 6809 family parallel interface chip that provides two TTL-level, 8-bit handshaking ports. It allows for easy interface to computers, ASCII keyboards, and other devices that may not support RS-232C serial I/O. Due to the lack of universal standards for such interfaces, the I/O levels are left as TTL, and the user is required to make such level changes as may be necessary for his equipment.

System Port, Timers, Indicators and NOVRAM

The system port consists of a dedicated-function parallel I/O device which communicates with the NOVRAM used for system configuration, the DIP switches, the HDLC controller RESET, and the indicator driver interface for various LED-monitored system functions. It also provides timers for HDLC baud rate generation, software timing functions, Morse ID, and calibration of the modem frequencies. This section also includes the LED drivers.

The primary component in the system interface is U6 (6522), a very powerful 6809 family LSI part. The 6522 incorporates a pair of 8-bit programmable I/O ports, four control lines (for handshaking or other use), two 16-bit programmable timer/counters and an 8-bit shift register.

The A port of the 6522 is used to read DIP switch S1 at PA0 during initial system configuration. The remaining lines are used for 4-bit data bus I/O to and from the NOVRAM XD2212 (U27), the high-order address line to U27 (A6 via PA1) and as the WE* (write) and STORE* (store) control lines to U27.

The 6522 B port is used to control the address bus (A0 through A5) of U27 and to provide output from Timer 1 (normally the HDLC baud rate generator) at PB6, and input to Timer 2 at PB7 during modem calibration.

The control lines are used as follows:

CA1	not used
CA2	HDLC reset and LED driver for modem calibration
CB1	not used
CB2	Morse code ID output and LED driver for modem calibration

U27 is a non-volatile RAM. It is used to store system parameters that are normally not changed such as call sign, terminal attributes and timing parameters, but which remain user alterable. This allows configuration changes for a given session only, or on a "permanent" basis.

The NOVRAM is a special type of integrated circuit (IC) memory that has two modes, volatile (temporary) and non-volatile (permanent) storage.

In the volatile mode, NOVRAM looks to the TNC like any other kind of temporary data storage memory. If you turn the TNC off, then back on again, the data values stored in the volatile (RAM) portion of the NOVRAM chip are completely random.

In the non-volatile mode, NOVRAM still looks like ordinary RAM except that, when a power-off-then-on cycle occurs, the data in the NOVRAM is initialized from a form of permanent memory (called electrically erasable programmable read only memory -- EEPROM). This EEPROM, a sort of "shadow memory" for the RAM portion of the NOVRAM, is where you normally save your call sign and other parameters by setting the values, then performing a PERM command.

Note that, while the EEPROM portion of the NOVRAM chip is called "permanent" you can still change it as many times as you like (a minimum of 10,000 times) via the PERM command provided on the TNC.

The NOVRAM interface consists of the following:

WE*	write-enable: store data to RAM function (volatile)
STORE*	store: save data to ROM (non-volatile) 10,000 cycles minimum
CS*	chip enable: always true
Q0-Q3	data bus: 4-bit parallel interface
A0-A6	address bus: 128 4-bit locations (512 bits)
ARRAY RECALL*	overrides STORE* if Vcc drops below 3 volts: transfers contents of ROM to RAM (no cycle limit)

U21 (7406) consists of 6 inverters with open-collector NPN transistors capable of sinking sufficient current to drive the LED indicators.

HDLC and Modem

The WD1933B controller is used to implement the HDLC standard bit oriented protocol including CRC check sum and zero bit insertion. The modem consists of a phase-coherent AFSK modulator operating with 1200 and 2200 hz tones (for Bell 202 compatibility), a PLL demodulator, and the necessary analog circuitry for interface with most common Amateur communications equipment. A 60-second hardware "watchdog" timer is inserted in series with the transmitter activation line to prevent accidental RF channel lockout.

U17 (WD1933B) is an LSI HDLC protocol controller with on-board PLL for clock recovery from NRZI-encoded data. This chip is interfaced to the system data and address busses directly, and to the control bus signals described above. Note that all data is inverted on transfer to or from the CPU, and the address assignments likewise reflect the inverted status of the address bus. The DRQO, DRQI and INTRQ interrupts are inverted via open-collector inverters U25A, B and C (7406) for interface to the 6809 IRQ* input. U17 is wired for selection of NRZI format and PLL clock recovery.

The transmit control signal from the HDLC chip MISCOT* pin is fed through U25F and C13 to U26 (NE555), a monostable multivibrator with an approximate 1-minute time period established by R28 (on header U35) and C14. The output from U26 drives Q1 (VN10KM), a power field-effect transistor (VFET) which in turn keys the transmitter via J3 pin 4.

The modem is based on the EXAR 2206/2211 chip set for FSK operation. U19 (XR2206) provides phase-coherent low distortion sine wave output at 1200 hz and 2200 hz for a 1200-baud Bell 202 compatible modulator. The output signal is buffered by U31A (LM1458) and sent to the transceiver interface connector. R33 is used to adjust the AC output level of the tone from a few millivolts to a few volts peak-to-peak. R36 sets the 2200 Hz frequency and R38 the 1200 Hz. Input sources are selected by push-on, pull-off jumpers to allow HDLC data (JP4) or a fixed logic 1 or 0 (JP5). The fixed levels are for calibration of the modem tone frequencies.

Morse code ID is accomplished by use of FSK tones. When not transmitting, the tone output is nulled. R31 is adjusted to provide a null in the 2206 output when pin 11 of U25 (7406) is in the low, or non-transmitting state. This allows voice override without disconnecting the TNC from the transmitter's microphone input.

The demodulator is based on the EXAR 2211 PLL demodulator which provides both data and a lock-detect output. R43 is used to set the unlocked frequency of the device for best demodulation of the input frequencies used. To satisfy the requirement for input tone balance to the 2211, a switched-capacitor filter based on the MF-1Ø (U28) is provided. The passband of the filter is tailored for typical 2-meter FM radios. The filter characteristics are determined by a resistor network plugged into a 16-pin header (U3Ø) to allow other filter characteristics to be selected easily. The input level to the filter is monitored by LEDs D7 and D8 at the output of buffer U31B (LM1458).

Jumpers are provided to allow for calibration of the modem (JP1 through JP5) and for disconnecting the modem from the HDLC chip (via J5). This flexibility allows the use of external modems for specialized applications and experimenting, as well as using the on-board modem for other purposes. Finally, there is a modem tuning indicator connector (J7).

Memory Bank

The memory bank consists of six JEDEC-standard 28-pin "Byte-wide" sockets. One site (U7) is mapped for an 8k static RAM and four (U9-U12) are mapped as up to 16k EPROM or static RAM sites. The remaining site may be configured, via jumpers JP6 and JP8, to support a 2k through 16k EPROM, EEPROM or RAM. The configuration supports 2716, 2732, 2764 and 27128-type EPROM devices, but not 2532 or 2564-type EPROMs.

The JEDEC standard chosen (there are two) incorporates two-line control of the memory chip (CE* and OE*). This simplifies the controller timing circuitry and allows multiple-sourced memory chips to be used. The TNC is address-mapped for one 8k-by-8 static RAM chip, U7 (6264) to occupy low memory (ØØØØ-1FFF hex) and four 8k-by-8 chips (U9-12) to occupy high memory (8ØØØ through FFFF hex). U8 is mapped into the 16k space from 4ØØØ through 7FFF hex. A replacement of the address decoder (U4) will allow other memory mappings to be used with no board changes.

Power Supply

The power supply provides the needed voltages at the required current levels for proper TNC operation. Specifically, +12, -12 and -5 volts are supplied for the modem and

RS-232C circuitry, and +5 volts is sent to all circuitry. The primary 5 volt regulator (U24) IC is usually mounted on board on a heat sink. The custom power transformer is the only required off-board component on the TAPR TNC. The power supply consists of two subsections, the main +5 volt section, and the +/-12, -5 volt section.

The primary +5 volt section utilizes a bridge rectifier (D13-16) charging a large capacitor (C1). This raw DC is fed to regulator U24 (LM309K). The output of U24 is decoupled by numerous Ø.1-uF capacitors and distributed across the board to the digital logic.

The +/-12, -5 volt section incorporates a full-wave bridge (D9-12) fed from a center-tapped winding. The split DC obtained is filtered (C3 for +12 and C5 for -12 volts) and fed to on-board regulators U22 (7812) for +12 and U23 (7912) for -12 volts. A portion of -12 volts output is further regulated to -5 volts by U29 (79LØ5), while a portion of +12 volts is further regulated by U33 (78LØ5) to +5 for the modem section only. The resulting outputs are decoupled and distributed.

A user wire-wrap area is provided, primarily for custom interface to the user's transceiver or for level shifting the parallel interface from TTL levels if required. For convenience, the wire-wrap area has power supply busses provided for +5, +12, -12 volts and ground.

Jumper Placement

Several jumper pins are present on the board. Some of these jumper pins are used during the modem calibration procedure. Others are provided for flexibility in satisfying individual configuration requirements. Jumper placement for normal operation is discussed below.

JP1 This jumper (normally on) is removed to unlock the XR2211 VCO (U18) in order to allow measurement of its free-running center frequency during modem calibration.

JP2 This jumper (normally off) is connected to allow the on-board counter to measure the tone frequencies generated by the XR22Ø6 modulator (U19) during modem calibration.

JP3 This jumper (normally off) is connected to allow the on-board counter to measure the voltage-controlled oscillator (VCO) frequency of the XR2211 phase-locked loop (PLL) tone decoder (U18) during modem calibration.

JP4 This jumper (normally on) is connected to allow the on-board modem to accept data from the HDLC controller chip (U17). It is removed during modem calibration.

JP5 This jumper (normally off) is used during the modem calibration procedure to force a low (jumper on) or high (jumper off) control signal to the modulator during the modem calibration.

JP6 This jumper allows socket U8 (normally empty) to accept a 2k byte RAM chip (jumper to the left) or to accept an 8k byte RAM chip (jumper to the right).

JP7 This jumper selects the system clock speed. In the right position (normal) it selects a clock input to the microprocessor of 3.6864 MHz and a system clock speed of 921.6 kHz, which allows the TNC to operate with standard 1 MHz microprocessor and peripherals and slower access memories. In the left position, it selects a clock input to the processor of 7.3728 MHz and a system clock speed of 1.8432 kHz. In order to operate at this speed, the TNC requires high speed parts and will support higher baud rates.

JP8 This jumper allows socket U8 (normally empty) to accept a 2k byte to an 8k byte memory IC (jumper to the left) or a 16k byte memory IC (jumper to the right).

In addition to the numbered jumpers listed above there are eight jumpers normally installed in the modem disconnect, J5 (just to the right of U17). (The second and eighth pin pairs, counting from the top of the connector, are unoccupied.) These jumpers are necessary when using the on-board modem for packet communications and are removed to connect an off-board modem. One may also remove these jumpers to experiment with the on-board modem for other uses. If you want to use an off-board modem, remove these jumpers and use J5 to connect a cable to your external modem. A detailed description of this connector is given later in this chapter.

DIP Switches

The functions of the DIP switches are detailed below. The ON position is that closest to the right edge of the board.

Note that the function of switch 1 is defined by software and may be changed in future software releases.

All commands referred to in this section are explained in detail in Chapter IV of this manual.

Switch 1 [DEFAULT SELECT] (normally OFF)

This switch is read by the system software at the time a reset is performed.

ON instructs the software to initialize the operating parameters from the defaults stored in the system EPROM. This switch must be in the ON position when the TNC is first turned on until you have stored your operating parameters in NOVRAM. See the PERM command in Chapter IV. This switch also activates the autobaud routine (see Chapter III).

OFF instructs the software to initialize the operating parameters from the values stored in NOVRAM. Following a "soft" reset (via the RESET command), the values read will be the volatile, non-permanent ones. Following a "hard" reset (using the reset switch) the values may be the volatile, non-permanent ones, or the non-volatile values, depending on the setting of Switch 4. The non-volatile memory will be read following a power off/on cycle.

Switch 2 [NOVRAM BANK] (normally ON)

This switch is used to choose one of two sets of NOVRAM-based parameters. This allows you to select from two different sets of operating parameters allowing rapid reconfiguration for different applications such as ragchewing, mailbox systems, or two users.

Switch 3 [RESET] (normally OFF)

This is the system reset switch. To perform a "hard" reset, toggle this switch ON, then OFF. It must be in the OFF position for the TNC to operate.

Switch 4 [NOVRAM DISCONNECT] (normally ON)

This switch is used to disconnect the NOVRAM from the system reset line when performing a hardware reset via Switch 3. This feature may be used when you have a special set of parameters stored in NOVRAM that you wish to retain during a reset operation but have not saved via the PERM command. This switch is normally left in the ON position, and the TNC probably will not power up properly if it is left in the OFF position.

Terminal Interfacing

The TAPR TNC has two data ports for connection to the station terminal or computer: a serial port and a parallel port. The serial port is the primary terminal I/O port and is selected as the I/O port by the default operating parameters. In this mode, diagnostic and link information is provided on the parallel port. The parallel port may also be used to control the TAPR EPROM programmer, and may be used for terminal I/O at a future date.

Serial Port Operation

The Electronic Industries Association (EIA) RS-232C Standard defines a widely used serial data interface. A subset of the RS-232C standard was used to design the TAPR TNC serial port.

The serial port is interfaced via J2, a DB-25S style connector wired as Data Communications Equipment (DCE). (This is the same configuration a modem would have.) This allows connection of a standard terminal to the TNC through a pin-for-pin cable interface. For packet stations using a standard RS-232C compatible terminal, such a pin-for-pin cable (utilizing pins 1 through 8 and 20) is all that will be required for attaching to the TNC.

Unfortunately, not all manufacturers provide standard RS-232C interfaces for serial I/O ports. The following information may be useful if you have a non-standard device.

According to the standard, a logical "0", "spacing" or "on" state exists when a voltage between +3 volts and +25 volts is present on a line. Similarly, a logical "1", "marking" or "off" state exists when a line has a voltage between -25 volts and -3 volts. An important point to note here is that some so-called RS-232C interfaces use standard TTL logic levels (0 and +5 volts) to interface to a serial port. If your attached terminal or computer uses any voltage between +3 volts and -3 volts as an output level on its serial port, it may not work with the TAPR TNC!

RS-232C defines 20 signal lines on the 25-pin connector. The TAPR TNC, like most other devices using the RS-232C interface scheme, implements only a subset of these signal lines. Specifically, the following lines are used:

Description	Terminal (DTE) Pin	Direction	TNC/Modem (DCE) Pin
Protective Ground	1	= = =	1
Transmit Data	2	= = =>	2
Receive Data	3	<= = =	3
Request To Send	4	= = =>	4
Clear To Send	5	<= = =	5
Data Set Ready	6	<= = =	6
Signal Ground	7	= = =	7
Data Carrier Detect	8	<= = =>	8
Data Terminal Ready	20	= = =>	20

The function of these lines is explained below.

Protective Ground: This line should be electrically bonded to the machine or interface frame. It may also be tied to signal common.

Signal Ground: This line establishes the common ground reference potential for all circuits except Protective Ground.

Transmit Data (TXD): This line is the serially transmitted data from the terminal to the TNC. It is held at the marking level (less than -3 volts) by the terminal when no data is being sent.

Receive Data (RXD): This line is the serially transmitted data from the TNC to the terminal. It is held at the marking level (less than -3 volts) by the TNC when no data is being sent.

Clear To Send (CTS): This line tells the terminal whether or not it may send data to the TNC. An "on" level tells the terminal it may send data while an "off" level tells it to stop sending data.

Request To Send (RTS): This line tells the TNC that the terminal is ready to send data.

Note that when the attached terminal or computer tells the TNC to stop sending data via the RTS line, the hardware buffer in the UART (6551, U14) may still send as many as two characters.

Data Set Ready (DSR): This line from the TNC indicates that the TNC is ready or available for use. It does not mean that a connection has been established as defined by RS-232C. The current TNC software simply asserts this line during initialization and holds it in the "on" state.

Data Terminal Ready (DTR): This line tells the TNC that the attached terminal or computer is ready or available. The current TNC software ignores the state of this line.

Data Carrier Detect (DCD): This line is held high by a pull-up resistor on the TNC, and any DCD signal generated by the terminal is ignored. Thus, this line can function as an output to a terminal requiring a DCD input signal.

RTS/CTS Handshaking

The RTS/CTS and DTR/DSR lines are used for hardware "handshaking" protocol to control the flow of data between the terminal (DTE) and the TNC (DCE). The terminal indicates that it is on line by asserting DTR to the TNC, then waits for the TNC to indicate that it is on line by asserting DSR to the terminal. Note that the current TNC software always asserts DSR and ignores the state of DTR.

The TNC implements a hardware "handshaking" protocol with the attached DTE via the Request To Send/Clear To Send (RTS/CTS) pair for transferring data between the TNC and the terminal. The TNC will assert CTS when it is ready to accept data from the terminal, and negate (make false) this line when it is not ready to accept data from the attached DTE. Thus, if you find the TNC won't send data to your terminal, one of the first things to do is to verify that the RTS line at pin 4 of J2 is not being held low. If the software flow control option is disabled, the TNC will not send data to the DTE unless RTS is asserted. If the terminal does not implement the RTS/CTS protocol, the RTS/CTS lines (pins 4 and 5 on J2) should remain unconnected.

Many simple serial I/O ports do not implement RTS/CTS or DSR/DTR handshaking. If these pins are not connected at the terminal end, they will be pulled up (and thus asserted) by resistors on the TNC end. However a non-standard serial connector may use some pins for other purposes, such as supplying power to a peripheral device, so <u>be sure that your</u>

system either implements the RTS/CTS and DSR/DTR handshake or has no connections to these pins of J2 whatsoever. Note that reference to RS-232C "compatibility" or the presence of a DB-25 type connector does not guarantee that you have a full RS-232C serial port!

The TAPR TNC supports all standard baud rates from 50 through 19,200. However, it is recommended that the terminal port be configured to run at a data rate of 4800 baud or less if attached to a computer. In addition, the port supports all standard parity options (odd, even, mark, space and none) as well as 7- or 8-bit character lengths. (These terminal parameters are discussed in the operations chapter.)

If you want to interface your TNC with a device configured as DCE (such as a telephone modem), a so-called "null modem" cable may be constructed to interchange the data and handshake signals. Basically, a null modem cable is an otherwise pin-for-pin RS-232C cable that has pins 2&3, 4&5, and 6&20 interchanged on one end. See for example Byte, February, 1981 page 198.

Modem Calibration

The TAPR TNC contains a versatile, user configurable FSK modem that is suitable for both 1200 baud "Bell 202" packet use as well as lower data rate use, such as 200 Hz shift 300 baud use on hf links. These directions will be for the default 1200 baud mode.

Calibration of the modem is usually only required when the unit is initially put into operation or when you want to change the modem characteristics. You may want to check the calibration if you find you have difficulty connecting to other packet stations, but recalibration is normally not necessary.

After powering up your TNC and getting the sign-on message, type the command CALIBRA. The following display will appear on your terminal.

TAPR TNC Calibration and Checkout

```
Menu:
   1  Set low tone (1200 hz) - 1st pot
   2  Set high tone (2200 hz) - 2nd pot
   3  Set demodulator (1700 hz) - 5th pot
   4  Set audio level - 4th pot
      and tone null - 3rd pot
   5  ROM checksum
   6  End calibration and reset

cal:
```

The prompt "cal:" means the TNC is in the calibration mode and is awaiting your command. To select any menu item, enter the number of the item in response to the prompt, followed by a carriage return. The TNC will then display a message describing the jumper configuration and adjustments to be made. The trimpots are counted from left to right with the TNC in normal orientation. For example, to calibrate the 1200 Hz tone:

```
cal:1<cr>
```

The following display will appear.

```
   1 Set low tone (1200 hz) - 1st pot
Remove jumpers JP1,JP3,JP4
Place jumpers JP2,JP5
Adjust R38
Type any key to continue
```

Using a small screwdriver, adjust the indicated pot. Be patient, as these trimmers are 20-turn units and you may have to turn the adjustment screw several times.

The LEDs D2 - CWID and D1 - RESET are your frequency setting indicators. D2, when lighted, indicates the frequency is too low and the trimmer needs to be adjusted counter-clockwise, while D1, when lighted, indicates the frequency is too high and the trimmer needs to be adjusted clockwise. When both LEDs are on, or flickering about equally, the setting is just right.

When you are satisfied with the LED indications, press a key on your terminal. The calibration routine will display something like:

```
Error is 0001 counts

cal:
```

An error count no worse than 2 or 3 is satisfactory. If you are unable to get this close to the desired count you may want to verify the frequency with a standard (external) frequency counter. If you still have trouble, consult the troubleshooting section of this manual before proceeding.

The calibration for the 2200 Hz and 1700 Hz tones is similar to that for the 1200 Hz tone. Simply respond to the "cal:" prompt with "2<cr>" for 2200 Hz or "3<cr>" for the 1700 Hz tone calibration.

At this point, all the modem frequencies are calibrated. The next part of the modem calibration procedure involves setting the transmitter (microphone) drive level and no-tone null for best operation with your radio.

Respond to the "cal:" prompt with "4<cr>".

Your terminal will then display the following:

```
     4 Set audio level - 4th pot
       and tone null - 3rd pot
  Remove jumpers JP2,JP3,JP5
  Place jumpers JP1,JP4
  Adjust R31 and R33
  Type 'Q' to quit, 'K' to key/unkey transmitter,
  any other key to switch tone.
```

Set the jumpers as instructed.

In the following step you will set the mic drive level to your transmitter. The voltage at TP1, located above U31, will be about 30 times the voltage appearing at the mic output from the TNC. If you don't have a manufacturer's recommended drive level, use a value of 20 mV. This corresponds to 0.6 volts p-p or 0.2 volts RMS at TP1.

Now adjust trimpot R31 (center trimpot) 20 turns clockwise.

Monitor test point TP1 with an oscilloscope or ac voltmeter. Strike the "K" (or "k") key on your terminal until LED D4 (PTT) is on, then adjust R33 (4th trimpot from the left) clockwise to increase level, counterclockwise to reduce level, for an amplitude of about Ø.6 volts peak-to-peak or Ø.2 volts RMS at TP1 (or 3Ø times the recommended drive for your transmitter). The waveform should be a clean sine wave in appearance.

Now strike the "K" key again so that LED D4 (PTT) is off. Adjust R31 (center trimpot) (starting counter-clockwise) for a null (minimum level) in the mic drive output. The null will be fairly sharp and very deep. After you have set the null, strike the "K" key again a few times and note that the audio drive switches between the preset level and the null. Verify that the "PTT" LED (D4) glows when the tone is present and is off when the tone is nulled.

Now strike the "Q" (or "q") key.

This completes the modem calibration. The next step will verify that the EPROMs are correct before exiting the calibration software.

Type:

```
cal:5<cr>
```

The screen on your terminal will then display the following:

```
   5 ROM checksum
8ØØØ nnnn
AØØØ nnnn
CØØØ nnnn
EØØØ nnnn
```

The software determines these checksum values by adding all the bytes in each EPROM according to a certain industry standard algorithm. The software release notes (if any) accompanying your software may also contain a list of these checksums for verification that your EPROMs are valid.

The value "nnnn" above simply means a 4-digit hexadecimal number. A hexadecimal numeral can be any value of Ø-9 or A, B, C, D, E or F. The use of 'nnnn' should not be taken to mean that the checksums for all four EPROMs are the same; this is very unlikely!

After verifying that the checksums are correct, strike any key, then type:

cal:6<cr>

The TNC will respond with:

6 End calibration and reset

Remove jumpers JP2,JP3,JP5
Place jumpers JP1,JP4
Type any key to continue

Set the jumpers as instructed. Your TNC is now ready for operation! The next keystroke will return you to command mode.

Calibration for Non-Default Tones

The TNC calibration system also allows you to set tones other than the defaults in case you wish to experiment. A typical example might be to set up for 170 Hz shift for HF use at 300 baud and slower. Of course, you may want to modify the passband characteristics of the MF-10 modem filter to optimize performance at a shift different from the default. It may also be necessary to alter the values of some of the timing and filtering components in the modulator, demodulator and/or watchdog timer to use other data rates successfully. It is for these reasons that many components of the modem are mounted on DIP headers U30, U34, and U35 instead of simply being soldered to the TNC PC board.

The calibration system measures a tone's frequency by counting the number of system clock cycles which occur in two oscillations of the waveform being measured. Since the system clock runs at 921.6 kHz, it "ticks" once each 1.085 usec. This count is compared with a 16-bit counter value, and the software then commands the LEDs to indicate whether the period just measured was too long or too short. The error-count message is the residual count "overflow" on the last period measured. Thus, an error of "0001 counts" means the measured period was 1.085 usecs too long, corresponding to a frequency about 3/4 Hz low in the case of the 1200 Hz tone.

In order to calibrate the modem to a different tone pair, you must first calculate twice the period of each frequency to be measured. The period is, of course, the reciprocal of the frequency (1/f). For example, a frequency of 2975 Hz, widely used in HF AFSK work, has a period of 1/2975 = 336.1 usec. Our measuring interval is twice this, or 2 * 336.1 = 672.2 usec.

The next step is to determine how many CPU clock cycles occur in the measuring interval. With a CPU clock frequency of 921.6 kHz, each clock cycle is 1.Ø85 usec long. Again using 2975 Hz as the reference (which we will continue to use throughout this discussion), there are 672.2/1.Ø85 = 619.5 clock cycles in the measurement interval. Since the counter used can only work in terms of integers, we round this result to the nearest integer, giving a value of 62Ø.

How do you use this count information? When you enter the CALIBRAte command and then select option 2, you must enter the new count value you want the software to use. In our example, the value 62Ø (you may drop leading "Ø"s if you wish) is entered as follows:

```
cal:2/62Ø
```

The "/" tells the software to look for the new number you are supplying in place of its normal one.

In a similar manner, values for the modulator low tone and the free-running VCO frequency of the XR2211 demodulator may be calculated. Note that the 2211's VCO frequency should be the average of the high and low tones for best operation (the default 17ØØ Hz is midway between 12ØØ Hz and 22ØØ Hz).

If the TNC won't calibrate to your new tones, but calibrates to its defaults properly, it probably means that you must change a component value on the DIP header (U34 or U35) associated with the adjustment you are trying to make. To reduce the frequency limits of an adjustment, increase the value of the resistor in series with the trimpot you are adjusting, or lower the value to increase the frequency limits. Note that 1% temperature stable resistors are used along with (expensive!) zero drift capacitors in your TNC modem. Don't use typical 5% carbon resistors or other than "COG" (NPO) coefficient capacitors in the modem frequency determining components or you may experience some very annoying calibration drift.

TNC Modem Disconnect

The TAPR TNC design includes provision 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 primarily for those who wish to interface external modems to the TNC. Familiarity with modem and serial data channel terms is assumed.

2Ø-pin connector J5 is available for disconnecting the on-board modem and allows connection of an external modem at TTL interface levels. A TTL high is greater than 2.4 volts but less than 5.25 volts while a TTL low level is greater than -Ø.4 volts but less than Ø.8 volts. DO NOT connect an RS-232C level modem directly to J5!

Normally, jumpers are installed to connect pins 1-2, 5-6, 7-8, 9-1Ø, 11-12, 13-14, 17-18 and 19-2Ø. If the on-board modem is to be used, all of these jumpers <u>must</u> be installed.

The connector pinouts will now be described.

Pin 1 Carrier Detect In

This pin tells the 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 pin 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 CD1*

This pin is normally tied to ground via pull-down resistor R76, and tells the HDLC controller to interrupt the microprocessor (uP) 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 CDØ*

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 MSCOT*

This line is an output from the HDLC controller. Unless otherwise 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 pin is an input to the on-board modem and conforms to the specifications outlined above 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 1 minute, but can be changed by selection of R28 on header U35.

Pin 7 DSR*

This pin is an input to the HDLC controller and 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 DTR*

This pin is an output from the HDLC controller to the modem telling it 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 RTS*

This pin is an output from the HDLC controller and tells the attached modem that the HDLC port has data to send. It is used as a handshake with 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 CTS*

This pin is an input to the HDLC controller and 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, above.

Pin 11 TC*

This line is the transmitter clock. While various duty cycle widths are acceptable, a square-wave clock is preferred. If the DPLL is enabled (pin 15, below), the clock frequency must be at 32x the data rate (38.4 kHz for 1200 baud); otherwise the clock must be a 1x 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 32x the data rate for the HDLC port and provides a square-wave signal under software rate control.

Pin 13 RC*

This pin 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 32X*

This pin is an input to the HDLC controller and 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 TC* and RC* clocks must be at 32x 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 at 1x 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 NRZI*

This pin is an input to the HDLC controller and 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 pin 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 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 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 2Ø 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. These lines are RI*, RI1* and RIØ* and are normally disabled by pullup resistors R73, R74 and R75 respectively.

These lines 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 TAPR TNC includes a connector for attaching a tuning indicator. The attached unit may range from an oscilloscope to a specialized LED-style unit. Please refer to the Exar Application Note referenced in the Bibliography for details on functions of the XR2211 signals available on this connector.

The connector pinouts will now be described.

Pin 1 Ground

This pin is the TNC's analog ground reference. It should not be used to sink appreciable current or the modem 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, and care should be exercised to ensure that no extraneous signals or low-impedance loads are attached.

Pin 3 Demodulator Reference Voltage

The internal XR2211 data comparator reference voltage is available on this pin. By comparing this value with the signal on pin 2, correct tuning may be accomplished. As above, this pin must be carefully shielded from noise, and has a high internal impedance.

Pin 4 Data Carrier Detect

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

Pin 5 +12 Volts

This pin is a source of +12 volts dc. It should not be used to source more than a few milliamperes of current or degradation of the on-board modem's weak-signal performance may result.

CHAPTER VI

TROUBLESHOOTING

Troubleshooting Hints

General

WARNING: Never remove or insert an IC with power on!

The TAPR TNC is a complex piece of electronic equipment. Servicing must be approached in a logical manner. While it is not possible to present all possible problems, symptoms and probable cures, this section of the manual will give direction to troubleshooting based on our experience.

In most cases we have found that careful visual inspection combined with simple measurements generally reveals the problem. The most useful single instrument for trouble-shooting is a good DVM that can read AC and DC volts, and can non-destructively test resistance while the ICs are still in their sockets.

While a number of checks may be made without the aid of an oscilloscope, you will need one to check signals at various points on the board if you fail to locate the problem by visual means or with a meter. Be very careful about shorting pins on ICs when applying meter or scope probes to the board. It is a good idea to attach a secure ground lead to the meter or scope, one that won't accidentally short across components on the board.

The first thing to check in any malfunction is the power supply. First check the power supply levels on the wire wrap area at the upper left. Are they close to their nominal values? Do all the ICs in the suspected area have the proper voltage on their power pins? Is there excessive ripple in any of the DC voltage lines? If so, check the regulator and associated components, working backwards toward the transformer. If the voltage is low, in conjunction with a hot regulator, suspect a short circuit on the board. (The large +5 volt regulator normally runs quite warm, but should not be near or above the boiling point of water. The voltage from it should be within about 300 mV of 5 volts.)

Step two is to note any unusual physical symptoms. Have you installed any ICs the wrong way? This is almost guaranteed to ruin the IC and produce a high current through it, detectable by the IC's high temperature. Are any components discolored? Does something smell like a burning TNC? Do any of the parts seems excessively warm? If you have never had your fingers on operating digital integrated circuits before you may erroneously conclude they are too hot when they are actually operating normally. This is especially true of the larger integrated circuits. In general their normal temperature will be well below the boiling point of water, but you may not want to keep your finger on them very long.

The third step is to carefully inspect the pc board and component installation. Are any cold solder joints present? (See Assembly soldering instructions in Appendix F.) Is a mounting screw shorting to the board anywhere? Are all ICs firmly seated in their sockets? Are any IC leads tucked under the chip or otherwise bent in such a manner that they aren't making proper contact with the IC socket? (This is a VERY common error!)

The fourth step is to inspect the diodes and electrolytic capacitors for proper installation. Are the diode cathodes pointing the correct way? Are the negative ends of the electrolytic capacitors pointing the correct way?

The fifth step is to inspect the interconnection cabling. Does it work on another TNC? Has the radio and/or terminal been successfully used on packet with this or another TNC? Are all the connections tight? Has the cable frayed or broken?

While these steps may seem obvious, careful inspection often will point to the problem or give significant clues as to the probable area of the TNC most suspect.

After the above inspection has been completed and apparent problems dealt with, it is time to proceed to more specific analysis.

Symptom: TNC appears dead

If the TNC powers up with D1 and D6 lit, followed by D2 lighting a second or so later, the processor is working and the software is probably working correctly. You should suspect the RS-232C port at this point. Check all connections and verify the logic levels according to the RS-232C troubleshooting guide in this Chapter. If you see D1 and D2 blinking every second or so, this is a message indicating that the processor is having trouble initializing U14, the UART.

If no LEDs wink during the reset cycle the problem may be more serious. Check to see that the crystal oscillator is working and that a clock signal is coming from the uP, U5. The crystal oscillator input to the processor is pin 38, and the clock output from the processor is pin 34. Both of these should show (possibly distorted) square wave signals, with that at pin 34 running four times slower than the one at pin 38. Verify that the clock input at pin 38 of U5 is running at the correct frequency.

Remember that all the logic circuits operate at standard TTL levels, and all digital inputs and outputs switch between these two levels. Thus, if you see logic signals switching between Ø and 1 volt, say, you can be sure there is a problem (usually a short). On the other hand, do not mistake switching transients on digital logic lines for improper operation -- these show up as ringing and other distortions.

Next, verify that there is activity on the control bus READ and WRITE lines, the 8 CHIP-SELECT lines, and the IRQ line. Each of these lines should show activity, and if any is quiet this is a sign of trouble.

Logic lines that show no activity may often be traced to a short on the pc board, probably due to a solder splash or bridge.

Address and/or data line shorts may also show up as lack of activity on the control bus lines, especially the chip selects. Check each of the 16 address and 8 data lines for activity. Any lines showing a lack of activity are not operating properly.

If you suspect problems with address or data lines, try removing all the memory chips. Each address and data line will now show a distinct pattern. The address lines should be (possibly distorted) square waves whose periods increase by a factor of two on successive lines as you step line by line from AØ to A15.

If you decide to use an ohmmeter to check for shorted lines, use a low voltage/low current test instrument. If in doubt, remove any ICs connected to the lines you are measuring. If you suspect a short, check the high density areas of the pc board for the problem. In most cases it will be found there.

It is very unlikely that the pc board itself will have a short as every board shipped by TAPR has been electrically tested for shorts and opens on a commercial "bed-of-nails" board tester prior to acceptance by TAPR.

Symptom: Modem won't calibrate or key transmitter.

Double check the placement of parts on each DIP header assembly, measuring resistor values and checking for shorted capacitors.

Calibration of the demodulator's 1700 Hz tone and the modulator's 1200 Hz and 2200 Hz tones is done in software by setting the specified device to generate the frequency in question and routing the signal to pins 16 and 17 of U6 where it is counted by the 6522. The calibration routine then examines this count and sets the LEDs appropriately. This signal also goes to U17, pins 26 and 31, where it should also be observable. In the case of the 1700 Hz tone, it is first passed through U20, acting as a Schmitt trigger. The input signal to the Schmitt trigger is a sawtooth wave. In the case of the modulator, the signal presented to U6 comes directly from pin 11, and should be a reasonable square wave.

Troubleshooting improper calibration amounts to checking for proper signals at U6 and following up any improper signal. If the calibration signal is present, but you cannot reduce the error count to zero, you may have an out-of-spec frequency determining component. Check the values of the appropriate passive components. Also, check the placement of jumpers! As a last resort, check the signal frequency with a frequency counter.

If the transmitter doesn't key and the PTT LED illuminates, the problem may be in the watchdog timer, U26. Check especially for an open timing capacitor C14 or a bad solder connection associated with R28 in header U35.

Symptom: Uncopyable transmitted or received packets

If no one seems to be able to decode your packet transmissions, it is often the case that your transmitter is being overdriven. The solution is to reduce the drive level via trimpot R33.

If you manually key the transmitter and a strong tone is heard, your tone null isn't properly set and should be adjusted via R31 for a minimum background or residual tone level.

If you are having problems hearing other stations, the demodulator circuitry associated with U18 may be at fault. Check the center frequency of the VCO in U18 via the 1700 Hz calibration procedure. Working in the direction of flow of the input signal, verify that it is being passed through to pin 2 of U18, the input pin. The signal there should be above 100 mV and below 3 V for proper operation of the demodulator. It should be relatively clean, although a few millivolts of noise is normal.

RS-232C Troubleshooting

If you can't get the TNC to sign on and accept data from your terminal or computer, the problem may be in the RS-232C interface. The troubleshooting guide below is provided as an aid to help in resolving problems that may be related to the RS-232C port.

Symptom: TNC won't sign on to the terminal.

If you find the TNC won't send data to your terminal, one of the first things to do is to verify that the RTS line at pin 4 of J2 is not being held low. If the software flow control option is disabled, the TNC will not send data to the DTE unless its RTS is asserted. If the terminal does not implement the RTS/CTS protocol, the RTS/CTS lines (pins 4 and 5 on J2) should remain unconnected.

Verify that the voltages on the TNC are correct. If the TNC is in otherwise good condition, check the following pins on the UART, U14 (6551). Pin 9 should be between 0 and +0.8 volts. If this is incorrect, check U16 pin 4 and verify that the voltage is greater than +3 volts. If this is correct, U16 or the traces around it may be bad. If this is not the problem, disconnect the terminal and check it again. If this doesn't help, U16 or R1 may be at fault.

If the above checks are ok, observe pin 10 of U14 with an oscilloscope and cycle the reset switch (switch 3) on the TNC. Transitions on this pin shortly after reset indicate that the TNC is sending data. Verify that transitions are also present on U15 pin 3. If these tests fail, the fault could be with U15, C11, U14, J2, the attached cable or faulty soldering (shorts, cold joints, etc.)

Symptom: The TNC appears to be signing on but only gibberish is printed on the terminal.

This indicates that some combination of the data rate (baud), parity option, or number of start and stop bits are not set the same at the TNC and terminal ends of the RS-232C cable. If possible, set your terminal to one of 110, 300, 1200, 4800, or 9600 baud. These are the rates automatically detected by the autobaud routine on the board. Also verify that the terminal is set for seven data bits, space parity, and 1 stop bit. These are the default settings stored in EPROM. Make sure that DIP switch 1 is set ON (to the right), and perform a hard reset by toggling DIP switch 3 ON then OFF (right, then left). Wait until the gibberish stops, then enter up to 3 or 4 asterisks (*). If the TNC prints a readable message during this process, do a PERM to save the serial port parameters.

If the TNC still prints gibberish, set the terminal to 300 baud and do a power off then on cycle on both the TNC and terminal. The sign on message defaults to 300 baud when using switch 1 in the ON position, so this should fix it. If the message still fails to appear try troubleshooting with an oscilloscope, looking first at the TXD pin of U14 (6551).

Symptom: The TNC signs on OK but won't accept commands.

After the TNC signs on, try giving it a command such as MYCALL or any other command. If the default settings are in effect, it will attempt to "echo" each character you type back to the screen. If it doesn't echo, be sure that U14 pin 16 has a voltage level between 0 and +0.8 volts on it. The voltage on U16 pin 10 should be greater than +3 volts. If these voltages aren't correct, the fault could be in U16, U14, J2, R2, soldering or the interconnecting cable.

If the above checks are OK, use an oscilloscope to verify that data is present on U14 pin 12 and U16 pin 1 when you strike a key. If not, the data isn't getting from your terminal to the TNC. Check J2, the cable and U16 again. Finally, be sure that your terminal actually uses levels less than -3 volts and greater than +3 volts for signal levels. 0 and +5 volts may not work!

Diagnostic and Status Signals

During normal operation, the parallel user port (J1) is used to output diagnostic and status signals. The diagnostic signals may be of use during troubleshooting. The status signals are provided for use by bulletin board programs or other computer-based applications. This function is automatically disabled if the EPROM Programmer routine is entered.

The diagnostic signals represent interrupts to the microprocessor from its various peripheral chips. The signals are updated as the processor actually services the interrupts rather than as they are generated by the hardware. The signal in each case is toggled from low to high or high to low upon servicing the interrupt. Thus, for example, if you look at the timer interrupt signal with an oscilloscope, you should see a square wave with a period of 20 ms, since the timer interrupt is generated at 10 ms intervals.

The diagnostic signals present are:

J1 Pin Number	Designation	Interrupt Signal
19	PB0	UART (terminal) input
6	PB1	UART (terminal) output
20	PB2	6522 timer
7	PB3	1933 carrier-detect change
12	PB4	1933 XEOM with error
11	PB5	1933 XEOM without error
10	PB6	1933 REOM with error
9	PB7	1933 REOM without error

The five WD-1933 interrupt signals correspond to interrupts on pin 6 of the WD-1933. XEOM means "transmitted end of message". The only error possible for a transmitted message is an underrun condition which could occur if the data rate is too high for the processor or if many other interrupt requests are present. The XEOM-with-error condition should never occur during proper operation of the TNC. REOM means "received end of message". The errors possible for received messages are: overrun, which, like the underrun condition, should never occur; aborted or invalid frame, which is frequently present if there is random noise on the channel; and frame-check sequence error, which occurs if a genuine packet is received in error.

There are two interrupt signals from the WD-1933 which are not represented in the list above. These signals are DRQI (data request for input) and DRQO (data request for output). These signals are individually measurable at pins 18 and 19 respectively of the WD-1933

The status signals are provided for possible use by a computer attached to the TNC. The signals provided are shown below.

J1 Pin Number	Designation	Status Signal
15	PAØ	\
2	PA1	Link state
16	PA2	/
4	PA5	Operating mode
18	PA6	Active packet buffer
5	PA7	Carrier Detect

The link state is represented by a number from 1 to 5 in binary appearing on pins 15, 2, and 16. The possible link states are:

Pin 16	Pin 2	Pin 15	Link State
Ø	Ø	1	Disconnected
Ø	1	Ø	Connect attempt in progress
Ø	1	1	FRMR condition detected
1	Ø	Ø	Disconnect in progress
1	Ø	1	Connected

The significance of these link states is described in Chapter VII and appendices B and C. Note that you may regard pin 16 as a connect indication, similar to the modem carrier-detect that a telephone bulletin-board program might use.

Pin 4 indicates that the TNC is in Command Mode (1) or data mode (Ø). Pin 18 indicates that the radio transmit buffer is active, meaning that not all packets which have been generated have been transmitted. Pin 5 is a copy of the state of the audio carrier detect, which is also indicated by LED D5.

CHAPTER VII

PROTOCOLS

Explanation of Protocol

The material in this chapter is intended to supply an overview of the protocols used to transmit data by the TAPR 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. Persons already well versed in networking will want to skip this chapter and refer to Appendix B, AX.25 Protocol Specifications and Appendix C, VADCG Protocol Specifications.

The TAPR 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 TAPR 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, i.e., voltages on a hardwire line or modem tones on phone or RF links.

The physical layer of the TAPR TNC is described in Chapter V. It is compatible with the VADCG TNC. The actual modem interface is compatible with the Bell 202 standard which is similar to the CCITT V.23 standard. Any other hardware device which is compatible with the Bell 202 standard will be compatible with the TAPR 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 TAPR 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 TAPR TNC is a Western Digital 1933 running in NRZI mode. 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 sending bit stream data asynchronously as is done by the TAPR TNC. Other chips which are compatible with the 1933 are the Intel 8273 (used on the VADCG TNC) and the Zilog 8530.

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. These are 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 unacknowledgeable but can be transmitted by layer two at any time without fear of disturbing higher layers. It is used by the TAPR TNC for beacon frames and for sending information frames when the TNC is not connected to another TNC, i.e., CQ and QST activities.

The exact specifications for AX.25 are supplied in Appendix B, AX.25 Protocol Specification. The TAPR implementation makes three deviations from that specification. These deviations are detailed below.

The DM Frame

The DM frame is sent whenever a non-SABM frame is received when the TNC is in the disconnected state. TAPR has expanded this definition. A TAPR 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

The Address Field

The address field has a single digipeater call sign slot specified. TAPR 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 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 Ø. If that subfield is the current TNC, the frame is repeated, first setting the H bit in that 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 sending packets in the other direction.

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

The Poll/Final Bit

The handling of the Poll/Final bit is an area of controversy. X.25, from which AX.25 was taken defines the uses 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 TAPR TNC code does not use the P/F bit to perform its error recovery; all packet retries are based on the timers, RR, and REJ frames already defined by AX.25. The TAPR 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 TAPR TNC will generate an RR frame with the final bit set in response to a frame with the poll bit set.

All of the above items are invisible to the TAPR TNC user and are mentioned only for the benefit of those who may be writing software for other TNCs.

The following paragraphs will list the frame types used by AX.25 and describe their purpose. The material is intended for persons desiring a general idea of what transpires on an AX.25 link. Implementers are again referred to Appendix B. The control field contents are given as they appear in memory after data is received, i.e., the high order 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 specification document reproduced in Appendix B 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 hex with the "x" character used to signify four bits which may be any value, depending on what ACK 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 7: 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
Ø3	UI - Unnumbered Information
ØF	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 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 by sending an I frame 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 can't connect to the requesting user, e.g., 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 occurs, i.e., the control 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 protocol description appears in Appendix C.

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 Use and Timing Functions

The following discussions mention timing parameters which are set by various commands. The time values selected are discussed in Chapter III.

An important part of any packet radio protocol 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 Carrier-Sensed Multiple Access (CSMA) with collision detection and collision avoidance.

CSMA means simply that (as every Amateur knows) 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. (Notice 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 and a packet is not being sent, as during the transmitter keyup 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 of simultaneous keyup by two or more stations. This is where collision detection and collision avoidance come in. The TNC detects a collision by the absence of an ACK 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 non-receipt 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 is received, the packet must be re-sent. If the unACKed 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 keyup 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. We have chosen the random time to be a multiple (Ø-15) of the transmitting station's keyup delay (TXDELAY). This is reasonable if one's own keyup 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 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.

APPENDIX A

ILLUSTRATIONS

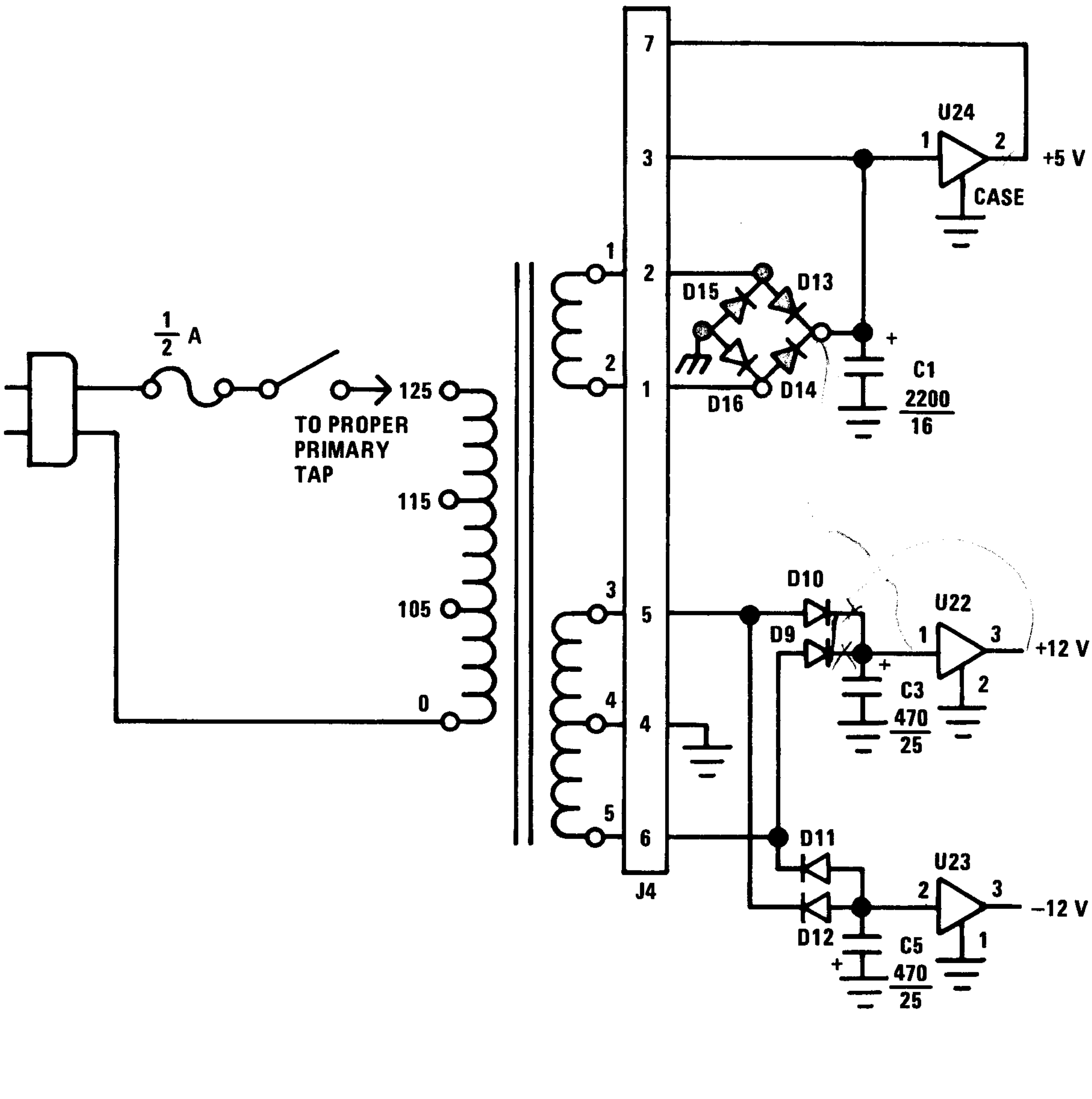


Fig. A.1   Transformer Wiring

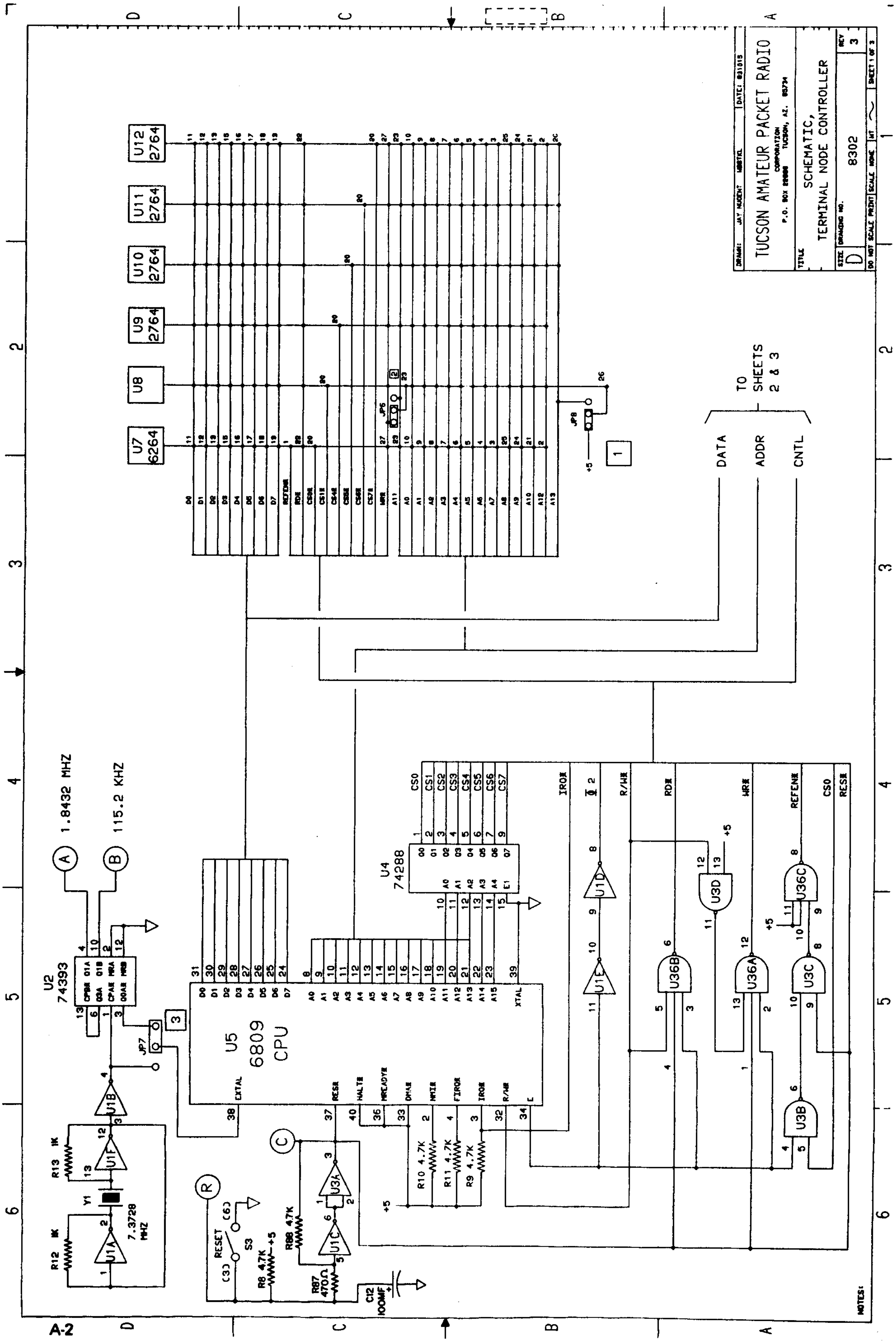


Fig. A.2 TNC Schematic — part 1 of 3

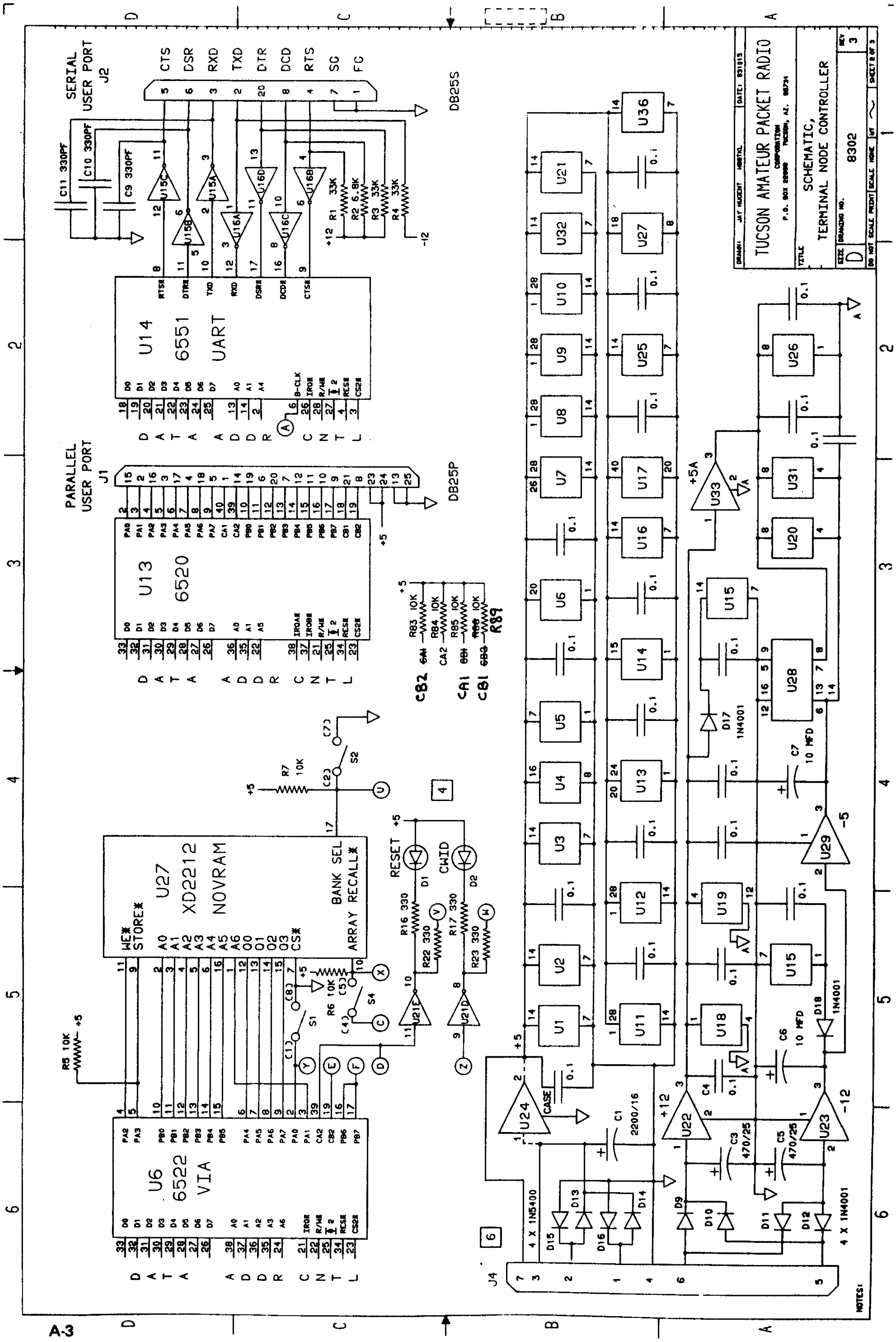


Fig. A.3 TNC Schematic – part 2 of 3

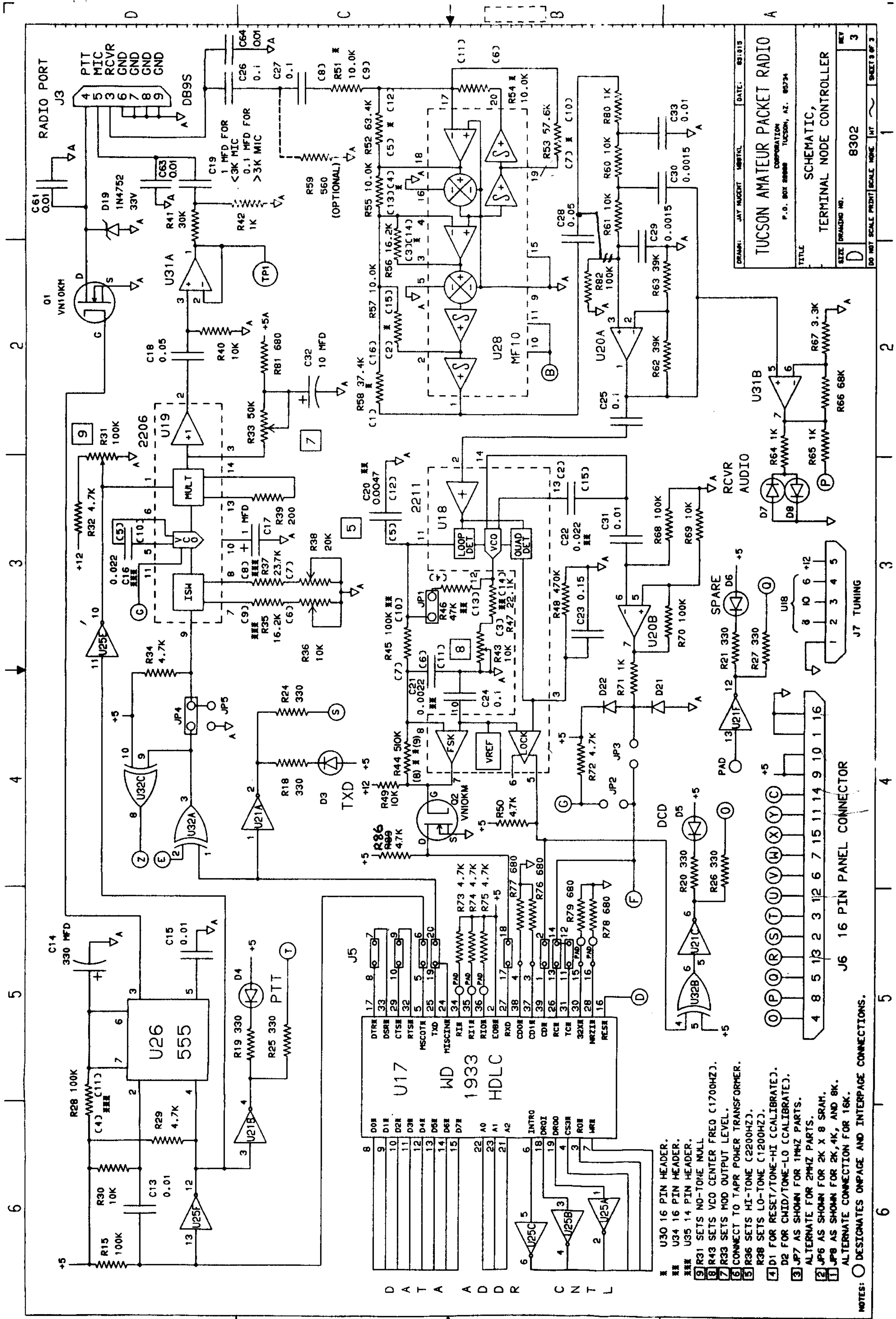


Fig. A.4 TNC Schematic – part 3 of 3

U1	74LSØ4	Hex Inverter
U2	74LS393	Dual 4-bit Counter
U3	74LSØØ	Quad NAND Gate
U4	74S288	32 x 8 Shottky PROM
U5	68Ø9	Microprocessor
U6	6522	Parallel Port w/Timers
U7	6264	8k x 8 Static RAM
U8	----	(spare memory site)
U9	2764	"8" 8k x 8 EPROM
U1Ø	2764	"A" 8k x 8 EPROM
U11	2764	"C" 8k x 8 EPROM
U12	2764	"E" 8k x 8 EPROM
U13	652Ø	(or 6821) Parallel Port
U14	6551	Serial Port (UART)
U15	1488	TTL --> RS232 Driver
U16	1489	RS232 --> TTL Receiver
U17	1933	HDLC Controller
U18	2211	FSK Demodulator
U19	22Ø6	FSK Modulator
U2Ø	1458	Dual Op Amp
U21	74Ø6	Hex O.C. Inverter
U22	7812	+12 Volt Regulator
U23	7912	-12 Volt Regulator
U24	3Ø9	+5 Volt Regulator
U25	74Ø6	Hex O.C. Inverter
U26	555	Timer
U27	2212	1k bit NOVRAM
U28	MF-1Ø	Four-Pole Filter
U29	79LØ5	-5 Volt Regulator
U3Ø	----	16-pin DIP Header
U31	1458	Dual Op Amp
U32	74LS86	Quad Exclusive-OR Gate
U33	78LØ5	+5 Volt Regulator
U34	----	16-pin DIP Header
U35	----	14-pin DIP Header
U36	74LS1Ø	Triple NAND Gate

Fig. A.5 IC List

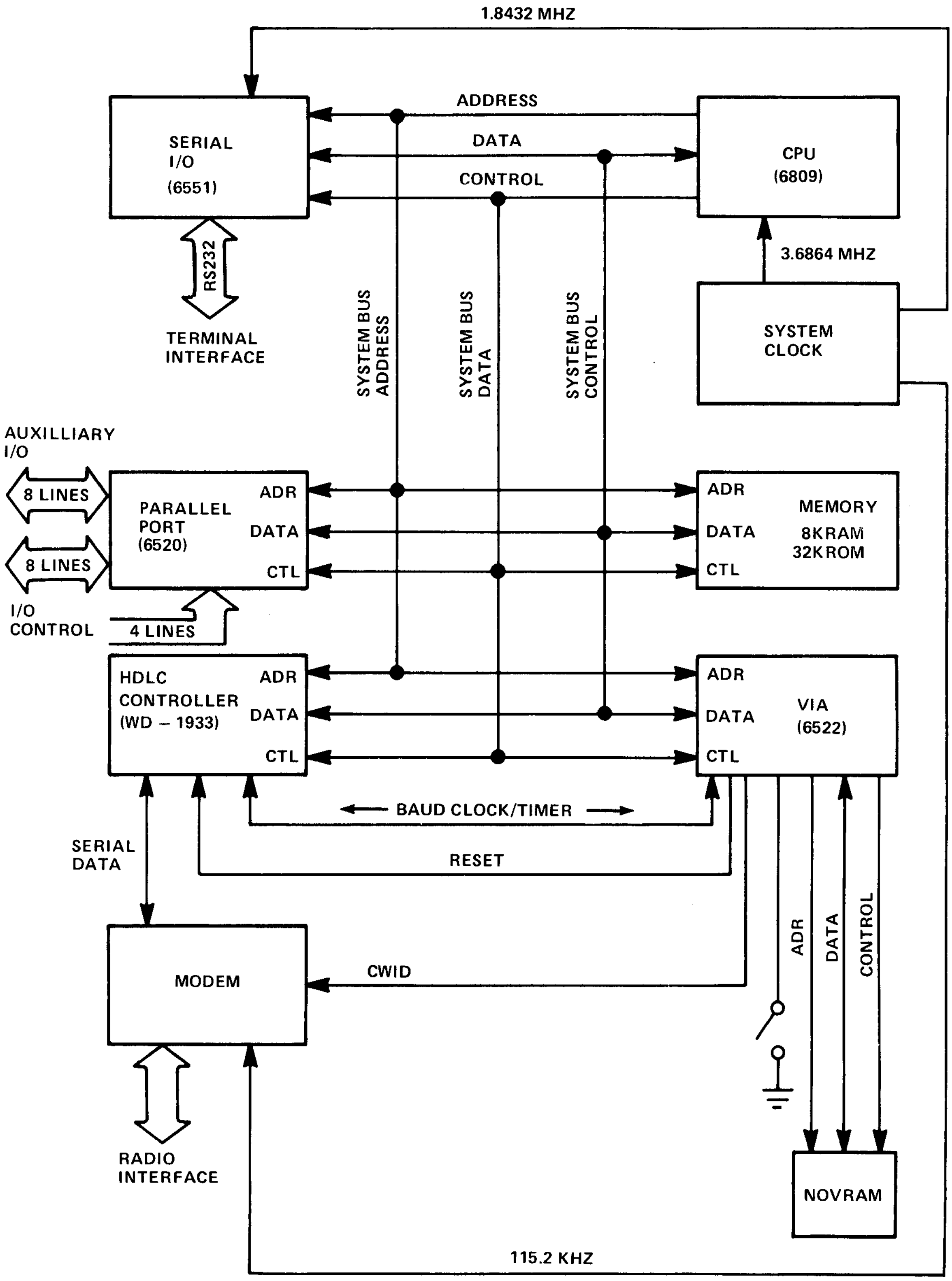


Fig. A.6 System Block Diagram

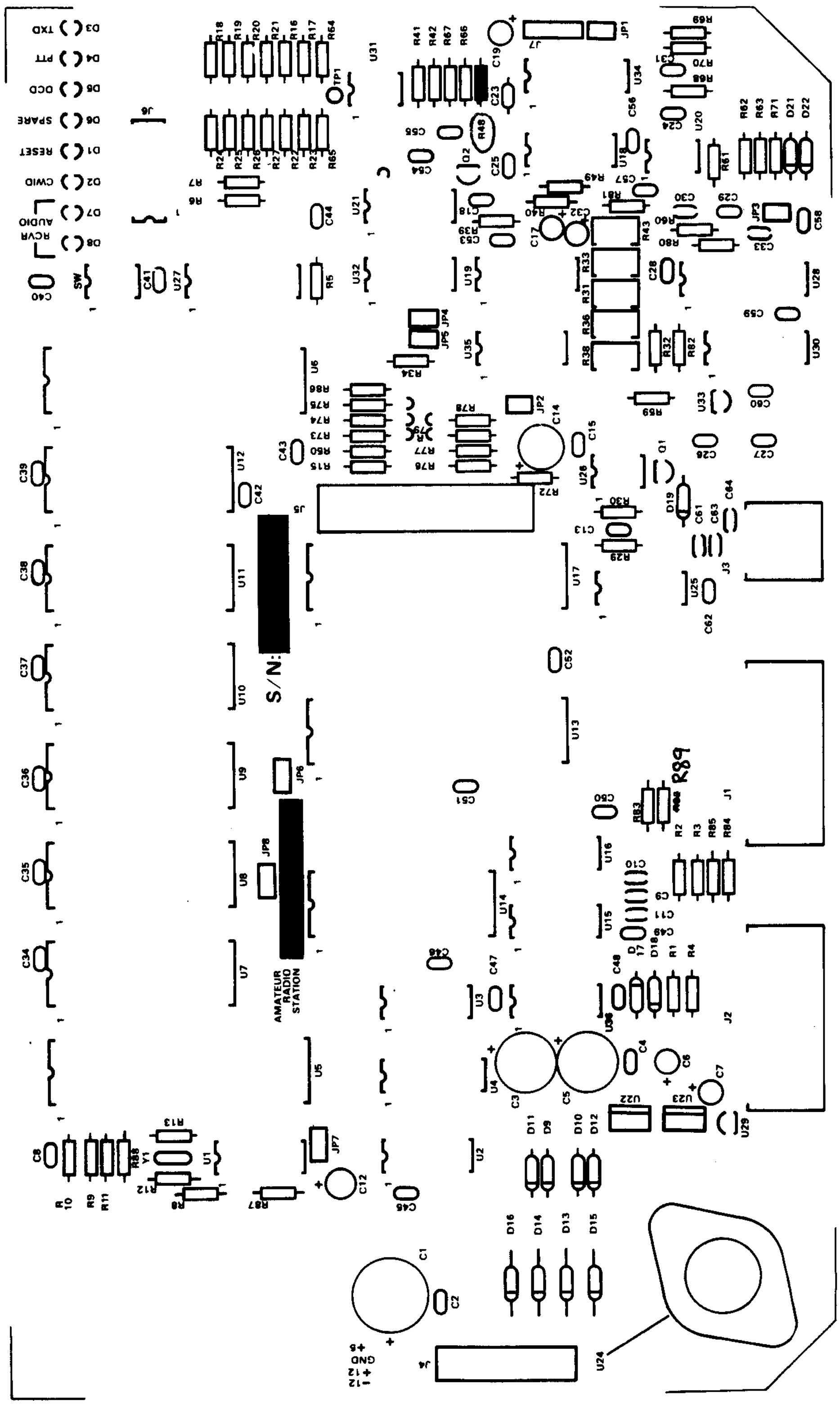


Fig. A.7 Board Layout

2Ø			3Ø	Ø	-----	4Ø	@	-...- (BT)	5Ø	P	.--.
21	!	---.	31	1	.----	41	A	.-	51	Q	--.-
22	"	.-..-.	32	2	..---	42	B	-...	52	R	.-.
23	#	.-.-. (AR)	33	3	...--	43	C	-.-.	53	S	...
24	$	...-.- (SK)	34	4	....-	44	D	-..	54	T	-
25	%	invalid	35	5	.....	45	E	.	55	U	..-
26	&	invalid	36	6	-....	46	F	..-.	56	V	...-
27	'	.----.	37	7	--...	47	G	--.	57	W	.--
28	(	-.--.	38	8	---..	48	H	....	58	X	-..-
29	)	-.--.-	39	9	----.	49	I	..	59	Y	-.--
2A	*	-.-.-- (KW)	3A	:	---...	4A	J	.---	5A	Z	--..
2B	+	-...- (BT)	3B	;	-.-.-.	4B	K	-.-			
2C	,	--..--	3C	<	invalid	4C	L	.-..			
2D	-	-....-	3D	=	invalid	4D	M	--			
2E	.	.-.-.-	3E	>	invalid	4E	N	-.			
2F	/	-..-.	3F	?	..--..	4F	O	---			

Note: All lower case characters are sent as upper case.

Fig. A.8 Morse Code Characters Available

APPENDIX B

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 TAPR/AMSAT AX.25 level 2 protocol has followed this set of specifications closely.

TAPR 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 Standard 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

There are two basic types of devices used in packet networking. One is a device called a data circuit-terminating equipment (DCE), which is usually a larger device, such as a Metropolitan Network Controller (MNC), or some other device that is smart enough to handle the link connection. The other device 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 Fig. B.1. Figure B.1 shows how 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 the 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 is discussed in a following 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.

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 5 contiguous 1 bits exist. Any time that 5 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 doesn't accidentally appear anywhere but at the beginning and end of 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 fifteen contiguous ones must be sent, with no bit-stuffing zeros 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.

Non-Repeater (Normal) Address Field Generation

The address field is encoded as shown in Fig. B.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 sub-field 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.

Destination Address							Source Address						
A1	A2	A3	A4	A5	A6	A7	A8	A9	A1Ø	A11	A12	A13	A14

Fig. B.3 Address Field Encoding

A1 through A14 are the fourteen octets that make up the two address sub-fields 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 sub-fields are the same format, so just the destination sub-field encoding will be shown here.

Destination Sub-Field Encoding

Fig. B.4 shows how an Amateur Radio call sign is placed into the destination address sub-field in octets 1 through 7 of the address field.

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	RRØ
Ø111Ø1Ø1	ØØ1ØØØØ1	ØØØ1Ø11Ø	Ø11Ø1ØØ1	ØØ11ØØØ1	Ø1ØØ1ØØ1	Ø	SSID	11Ø

LSB >>Order of Bit Transmission>> MSB

Fig. B.4 Callsign Encoded into 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 Ø 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 aren't 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 (upper case 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 (2Ø hex).

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:

```
ØØØØ-Ø111       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.

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 Fig. B.3a.

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

Fig. B.3a 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, and 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 Fig. B.4a.

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

Fig. B.4a Repeater Sub-Address Encoding

Where:

1. "E" is the HDLC extender bit as mentioned earlier.

2. "H" is the has-been-repeated bit. If H=0, then the frame has not been repeated, while if H=1, then the frame has been repeated.

Note: Advantages of the WB4JFI Addressing Scheme

Some of the advantages to using this addressing system are:

1. Every packet station will have a unique fixed address that doesn't change every time a new network is logged into.

2. Relocating to a new area won't 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.

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 multi-point operation.

There are three basic formats of the control fields. They are the Information format (I frames), the numbered Supervisory format (S frames), and Unnumbered control frames (U frames). Fig. B.5 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	Ø	N(S)		P/F	N(R)
S Frame	1	Ø	S S	P/F	N(R)
U Frame	1	1	M M	P/F	M M M

Fig. B.5 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.

Control Field Definitions:

Information Frame Control Field

I frames have bit 1 of the control field set to Ø. 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

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

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 (non-extended) control field, every I frame shall be assigned a sequential number varying from Ø 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 Fig. B.6. 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
Ø	N (S)			P/F	N (R)		

Fig. B.6 I Frame Control Field

Supervisory Frames

The supervisory frame control fields are encoded as shown in Fig. B.7. 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	Ø	Ø	Ø	P/F	N(R)		
Receive Not Ready	RNR	1	Ø	1	Ø	P/F	N(R)		
Reject	REJ	1	Ø	Ø	1	P/F	N(R)		

Fig. B.7 S Frame Control Fields

Receive Ready (RR) Response

Receive Ready is used to do the following:

1. To indicate that the sender of the RR is now able to receive more I frames.

2. To acknowledge properly received I frames up to, and including N(R)-1.

3. To clear a previously set busy condition created by an RNR command having been sent.

It should be noted that the status of the other side of the link can be requested by setting the poll bit.

Receive Not Ready (RNR) Response

Receive not ready is used to indicate to the sender of I frames that receiver is temporarily busy and cannot accept any more I frames. Frames up to N(R)-1 are acknowledged. Any I frames numbered N(R) and higher that might have been caught in between and not acknowledged when the RNR command was sent are NOT acknowledged.

The RNR condition can be cleared by the sending of a UA, RR, REJ, SARM, or SABM frame. The P/F bit can be used within the RNR frame to interrogate the status of the other side of the link.

Reject (REJ) Response

The reject frame is used to request retransmission of I frames starting with N(R). Any frames that were sent with a sequence number of N(R)-1 or less are acknowledged. Additional I frames may be appended to the retransmission of the N(R) frame if there are any.

Only one reject frame condition is allowed in each direction at a time. The reject condition is cleared by the proper reception of I frames up to the I frame that caused the reject condition to be initiated.

As with the other supervisory responses, the P/F bit may be used in the REJ frame.

Unnumbered Type Frames

Unnumbered frame control fields are either commands or responses. This standard follows X.25 as much as possible. The only deviation from X.25 is in the addition of the Unnumbered Information (UI) frame from ADCCP. X.25 is designed to work in full duplex systems with only one main device (DCE) and potentially many users (DTEs).

Amateur radio packet systems differ greatly on both of these respects. Not only is amateur radio packet networking done in a half duplex RF environment, but many DCE/DTE links many be sharing the same channel. Many amateurs have rejected the use of X.25 as a result of these problems. X.25 can easily be enhanced so that it will perform properly over amateur radio.

Fig. B.8 shows the layout of U frames implemented within this standard.

Control Field	Type	Control Field bits 1 2	3 4	5	6 7 8
Set Asynchronous (SABM) Balanced Mode	Command	1 1	1 1	P	1 Ø Ø
Disconnect (DISC)	Command	1 1	Ø Ø	P	Ø 1 Ø
Disconnected Mode (DM)	Response	1 1	1 1	P/F	Ø Ø Ø
Unnumbered Acknowledge (UA)	Response	1 1	Ø Ø	F	1 1 Ø
Frame Reject (FRMR)	Response	1 1	1 Ø	F	Ø Ø 1
Unnumbered Information (UI)	Either	1 1	Ø Ø	P/F	Ø Ø Ø

Fig. B.8 U Frame Control Fields

Set Asynchronous Balanced Mode (SABM) Command

The SABM command is used to place 2 stations in the asynchronous balanced mode. This is a balanced mode of operation known as LAPB where DCEs and DTEs are treated as equals.

Information fields aren't allowed in SABM commands. Any outstanding I frames left when the SABM command is issued will remain unacknowledged.

Disconnect (DISC) Command

The DISC command is used to terminate a link session between two stations. No information field is permitted in a DISC command frame. Any outstanding I frames will remain outstanding.

Disconnected Mode (DM) Response

The disconnected mode response is sent whenever the DTE or DCE receives a frame other than a SABM while in a disconnected mode. It is sent to request a set mode command, or to indicate it cannot accept a connection at the moment. The DM response cannot have an information field.

A DCE or DTE in the disconnected mode will respond to any command other than a SABM with a DM response with the P/F bit set to 1.

Unnumbered Acknowledge (UA) Response

The UA response frame is sent to acknowledge the reception and acceptance of a U frame command. A received command is not actually processed until the UA response frame is sent. An information field is not permitted in a UA frame.

Frame Reject (FRMR) Response

The FRMR response frame is sent to report that for some reason the receiver of a command or information frame cannot successfully process that frame and that the error condition is not correctable by sending the offending frame again. Typically this condition will appear when a frame without an FCS error has been received with one of the following conditions:

1. The reception of an invalid or not implemented command or response frame.

2. The reception of an I frame whose information field exceeds the agreed upon length.

3. The reception of an improper N(R). This usually happens when the N(R) frame has already been sent and acknowledged, or when N(R) is out of sequence with what was expected.

4. The reception of a frame with an information field where one is not allowed, or the reception of an U or S frame whose length is incorrect.

When a CMDR or FRMR frame is sent, an information field is added to the frame that helps to explain where the problem occurred. This information field is three octets long and its content is shown if Fig. B.9 below.

Information Field Bits: 1 2 3 4 5 6 7 8	9	1Ø 11 12	13	14 15 16	17	18	19	2Ø	21 22 23 24
Rejected Frame Control Field	Ø	V(S)	C	V(R)	W	X	Y	Z	Ø Ø Ø Ø

Fig. B.9 FRMR Frame Information Field

Where:

1. The rejected frame control field carries the control field of the frame that caused the reject condition. It is in bits 1-8 of the information field.

2. V(S) is the current send state variable of the device reporting the rejection (bit 1Ø is the low bit).

3. V(R) is the current receive state variable of the device reporting rejection (bit 14 is the low bit).

4. If W is set to 1, the control field received was invalid or not implemented.

5. If X is set to 1, the frame that caused the reject condition was considered invalid because it was a U or S frame that had an information field that is not allowed. Bit W must be set to 1 in addition to the X bit.

6. If Y is set to 1, the information field of a received frame exceeded the maximum capacity of the device reporting the condition.

7. If Z is set to 1, the control field received and returned in bits 1 to 8 contained an invalid N(R).

8. Bits 9, and 21 to 24 are set to Ø in both CMDR and FRMR frames. Bit 13 of a FRMR is set to Ø if the rejected frame was a command, or 1 if it was a response.

Unnumbered Information (UI) Frame

The unnumbered information frame is used to pass information along the link outside the normal information controls. This allows information fields to go back and forth on the link bypassing flow control. Since these frames are NOT acknowledgeable, if one gets wiped out, there is no way to recover it.

The UI frame is not defined in X.25. It has been taken from ADCCP to allow uncontrolled information to flow through the link without interfering with a next higher layer.

Link Error Recovery

There are several link level errors that are recoverable without tearing down the connection. These error situations may occur as a result of malfunctions within the DTE or DCE, or if transmission errors occur.

Invalid Frame or FCS Error

If an invalid frame is received, or a frame is received with an FCS error, that frame will be discarded with no action taken.

Device Busy Condition

When a DTE or DCE becomes temporarily busy, such as when receive buffers are full, it will send a receive not ready (RNR) frame. This tells the other side of the link that the device cannot handle any more I frames at the moment. This condition is usually cleared by the sending of a UA, RR, REJ, or SABM command frame.

Send Sequence Number Error

If the send sequence number, N(S), of an otherwise error-free received I frame does not match the receive state variable, V(R), a send sequence error has occurred, and the information field will be discarded. The receiver will not acknowledge this frame, or any other I frames until N(S) matches V(R).

The control field of the erroneous I frame(s) will be accepted so that link supervisory functions can still be performed, such as checking the P/F bit. Because of this updating, the retransmitted I frame may have an updated P bit and N(R).

Reject (REJ) Error Recovery

REJ is used to request a retransmission of I frames following the detection of a sequence error.

Only one outstanding reject condition is allowed at a time. This condition is cleared when the requested I frame has been received.

A device receiving the REJ command will clear the error by sending over the I frame indicated in N(R) of of the REJ command frame.

Time-out Error Recovery

When a transmission abnormality wipes out a single I frame, or the last I frame of a group, there is no way of telling this immediately, since the receiver does not necessarily know something was sent until another frame is sent resulting in an out-of-sequence error. To cope with this situation better, some form of time-out delay will be incorporated by the sender after it sends out a frame. This time-out timer is started at the time a frame is sent, and stopped by the reception of an acknowledgment for the sent frame. If the timer times out before an acknowledgment is received, any unacknowledged frames are retransmitted. The delay is an agreed upon amount that will vary with the type of RF medium and signalling speed used.

Rejection Error

A rejection error condition occurs when an error-free received frame has one of the following problems:

1. An invalid command or response control field.

2. An invalid frame format.

3. An invalid N(R).

4. An information field that exceeds the maximum the device can accept.

Once a rejection error occurs, no more I frames are accepted (with the exception of the P/F bit still usable) until the error is resolved. The error condition is reported to the other side of the link by sending either a CMDR of FRMR response frame.

Primary/Secondary versus Balanced Operation

There are two basic classes of link level connections. The first, known as Link Access Procedure (or LAP) is often called an unbalanced service where the DCE is considered the primary (or master) devices and the DTEs are considered secondary (or slave) devices. The second class of service is known as LAPB, Link Access Procedure Balanced. In this service both devices are treated as equals as far as connection requests and other types of commands. There is still only one DCE and potentially many DTEs, but both ends can command the link equally.

Primary/Secondary (LAP) Operation

LAP is the older style of link control, where most of the intelligence was assumed to be in a large mainframe (the DCE) and the end users were just using smart terminals (the DTEs). Since network software can have a lot of overhead, it made sense at the time to put most of the overhead in the big computer, and just enough smarts to make the link work in the terminals.

Balanced (LAPB) Operation

LAPB is a slightly modified version of LAP. It has been changed to allow the two sides of a link to operate in a more balanced manner. In the official version of X.25 there is still only one DCE to potentially many DTEs, but the two can operate more as equals than master and slave.

LAPB is what this document describes for use over Amateur Radio packet networks. Even when there is a network controller overseeing the network operation, the balanced link procedure will enhance operation.

Connection Operation

In Amateur network operations, it would be very helpful if one level 2 protocol would work with the various RF systems in use. An example of this is the difference in operation between a simple two-station link, and multiple stations operating through a network controller. Obviously, when a network controller exists, it should be considered the DCE, while the other stations connecting to it would be the DTEs. A simple two-station connection is another matter. To this type of connection the station requesting a connection should always be considered the DTE, while the device that is receiving the connection request should operate as the DCE. This simple rule should eliminate any ambiguity that might otherwise occur under these conditions.

NOTE:
There are a couple minor changes from the official X.25 standard in the protocol recommended here. These changes are done only as absolutely necessary to work over the shared RF media. Since X.25 was written to work so that one DCE talked with many DTEs over a closed network, it cannot properly cope with a channel where there may be many DCEs linked to many DTEs. Some amateurs have thrown X.25 out because of this problem. It seems to take just a couple minor changes in the initial link set-up procedure to make X.25 work properly over amateur radio. Where these changes are made, both the original X.25 procedure and the recommended amateur procedure will be noted.

LAPB Procedures

The following describes the procedures used to set-up, use, and disconnect a balanced link between a DTE and DCE. These procedures have been taken from X.25 and conform very closely to that standard, except where it was necessary to change due to the radio environment.

Address Field Operation

All transmitted frames shall have address fields conforming to above mentioned rules. Except for all-call frames (those with a single octet of all ones), all frames will have both the destination device and the source device addresses in the address field, with the destination address coming first. This will allow many links to share the same rf channel. The destination address is always the address of the station(s) to receive the frame, while the source address contains the address of the device that sent the frame. The destination address can be a group name or club call however, if point to multi-point operation is allowed. This will be discussed further under link operations.

LAPB Connection Establishment

When a device (either a DCE or DTE) wishes to connect to another device, it will send a SABM command frame to that device and start a time-out timer (T1). If the other device is there and able to connect, it will answer with a UA response frame and at the same time reset both of it's internal state variables (V(S) and V(R)). The reception of the UA response frame at the other end will cause the device requesting the connection to abort the T1 timer and set it's internal state variables to Ø also.

If the other device doesn't respond before T1 times out, the device requesting the connection will re-send the SABM frame, and start T1 running again. This trying to establish a connection will continue until the requesting device has tried unsuccessfully a number of times. That number (N1) is variable, depending on the frequency of operation, type of transmission (e.g. terrestial vs. satellite), and the signalling speed in use. N1 will be discussed in another section.

Information Transfer

Once a connection has been established as outlined above, both devices are able to accept I, S, and U frames.

Sending of I Frames

Whenever a station has an I frame to transmit, it will send the I frame with N(S) of the control field equal to it's current send state variable V(S). Once the I frame is sent, the send state variable is incremented by one.

The station should not transmit any more I frames if it's send state variable equals the last received N(R) from the other side of the link plus seven. If it were to send more I frames, the flow control window would be exceeded and errors could result.

If a device is in a busy condition, it may still send I frames as long as the other device is not also busy.

If a device is in the frame-rejection mode, it will stop sending I frames.

Receiving I Frames

If a device receives a valid I frame (one with a correct FCS and whose send sequence number equals the receiver's receive state variable) and is not in the busy condition, it will accept the received I frame, increment it's receive state variable, and act in one of the following manners:

1. If it has an I frame to send, that I frame may be sent with the transmitted N(R) equal to it's receive state variable V(R) (thus acknowledging the received frame. Alternately, the device may send an RR frame with N(R) equal to V(R), and then send the I frame.

2. If there are no outstanding I frames, the receiving device will send an RR frame with N(R) equal to V(R).

If the device is in a busy condition, it may ignore any received I frames without reporting this condition other than repeating the indication of the busy condition.

The reception of I frames that contain zero length information fields shall be reported to the packet level but no information field will be transferred.

APPENDIX C

VADCG PROTOCOL SPECIFICATION

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

The protocol used by the Vancouver Digital Communications Group 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 zeroes
FLAG	Start of frame, bit pattern Ø111111Ø
ADDR	Address byte, hex ØØ to FF
CNTL	Control byte, indicates type of frame, 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 - Zeroes 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 & 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
Ø7H = connect acknowledge
53H = disconnect request
43H = disconnect acknowledge

The poll/final (P/F) bit, 1ØH, is used to force a response from the receiving station. Used here and in other frame types for this function.

FMCALL Call of station originating the frame (6 characters)

TOCALL Call of station receiving the frame (6 characters)

The callsign is left-justified in the field and padded with trailing blanks if the call is shorter than 6 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

Addresses:

As used in SF Bay Area, subject to change.

ØØ	No operation
Ø1-7F	Station direct connect addresses
8Ø	Reserved
81-9F	Repeater input channel addresses
AØ	Reserved
A1-BF	Repeater output channel addresses
CØ-FE	Reserved
FF	All parties addressed

APPENDIX D

HOOKUP FOR SPECIFIC RADIOS

This appendix contains interfacing information for the IC-2AT and the FT-2Ø8R, along with a table listing hookup information for several commonly used Amateur transceivers. All the data in the table is derived from information received from participants in our Beta test. While we hope there are no errors in this information, we urge caution in making use of it, in view of the potentially grave results some errors might produce. Please DO NOT use this information without independently checking its accuracy!

Each entry includes the source being quoted, the connector type, as described by that source, and the pin-out for the various functions. We also list the PTT open-circuit voltage as well as the current measured while keyed. In some cases more than one source described the same transceiver and we've tried to include all variations reported. In the case of reported variations we have simply listed them as described. They may be due to the use of alternative connectors where more than one exists on the unit.

Simple electrical tests with a VOM should be enough to determine the accuracy of the information listed. The primary situations to watch out for are accidently connecting the radio positive supply to the TNC's PTT line or audio input line (the latter is bad only if a polarized capacitor is used or the capacitor rating is exceeded.)

Interfacing the IC-2AT

The IC-2AT keys the transmitter by completing the ground connection on the microphone. The cleanest way to interface to this combination is to use an audio transformer, connecting the primary side to the TNC audio out and return lines (pins 5 and 6-9, respectively), and connecting one side of the secondary to the microphone input of the IC-2AT and the other to the PTT line on the TNC (pin 4). Almost any transformer should work, provided the turns ratio is not too great.

Interfacing the Yaesu FT-2Ø8R

The FT-2Ø8R must be interfaced carefully (presumably these peculiarities exist in other Yaesu equipment) in order to avoid damaging either the TNC or the radio. Using Yaesu's numbering of pins on the external connector, attach radio pin 1 to TNC pin 5, radio pin 2 to TNC pin 3, radio pin 3 to TNC pin 4 and radio pin 4 to TNC pins 6-9. On no account connect radio pin 6 to any pins on the TNC radio connector! The radio pins mentioned are the four closest to the FRONT panel of the HT, with pin 1 to the left as you hold it normally.

Note: Unless described to the contrary, all model numbers beginning with "TR" or "TS" in the table below refer to transceivers made by Kenwood.

Appendix D, HOOKUP FOR SPECIFIC RADIOS

SOURCE	RADIO	CONNECTOR TYPE	SP AUD PIN	PTT PIN	MIC AUD PIN	COMMON PIN	PTT OCV/ CURR.
KF4LG	TR-76ØØ	5-pin		2	1	4,5	4.39V/3ØmA
		1/8" phone	tip			sleeve	
N6MA	GENAVE	1/4" phone	tip			sleeve	12V/22mA
	GTX-1ØØ	1/4" phone		tip	ring	sleeve	
W4CQI	KDK FM-2Ø25A/E	4-pin		2	4	3	13V/1.ØmA
	Mark II	mini-phone	tip			sleeve	
WB9HBH	KDK FM-2Ø25	5-pin	1	5	2	case	12V/4Ø mA
	(NOTE: this arrangement via rear connector, J2)						
W1BEL	TS-7ØØA	9-pin	1	6	3	2,4	17.3V/Ø.4mA
	(Note: W1BEL used rear aux. conn., tied pins 8 and 9 together)						
W1BEL	AZDEN PCS-3ØØØ	EC-8Ø		9	12	2,11	9V/16mA
		mini phone	tip			sleeve	
WBØIBZ	TR-913Ø	6-pin	?	2	1	6	8.6V/49mA
W6SZX	ICOM 22-S	4-pin mike		3	1	4	13.8V/6ØmA
		small audio	tip			sleeve	
	(NOTE: W5FD used pin 2 as PTT)						
WB6HHV	Drake UV-3	Molex, under chassis. Added speaker audio to an unused pin, others per Drake. Changed R35 to 2.2k to help with p.s. "noise."					11.6V/3ØmA
WD6FPY	FT-227R	5-pin	1	3	2	case	?
WA6CFM	FT-23ØR	7-pin	5	3	2	1	4V/4ØmA
N6TE	ICOM 211	24-pin Molex	4	3	5	8	?
		Waldom 1625-24P					
WB6UUT	ICOM 22U	Amphenol CBC-8		5	1	7	9.5V/25mA
		mini phone	tip			sleeve	
	(Note: WB9SVM used pin 6 instead of 7 for common)						
WD9DBJ	WILSON MARK II	WILSON P1Ø16	J3-2	J3-3	J3-1	J3-4	12V/?
W7KB	TR 795Ø	6-pin		2	1	6	9.5V/16mA
		mini phone	tip			sleeve	
N8ANJ	HW-2Ø36	RS 274-2Ø4/2Ø5		1	2	4	Ø.8V/2.2mA
		phono	tip			sleeve	
NØDVS	TR 74ØØ A	4-pin		2	1	3,4	?
		mini phone	tip			sleeve	
	(NOTE: KB7XP measured 12V/7ØmA)						
W8KOX	TR 22ØØ	4-pin	3*	2	1	4	?V/#<1ØmA
	(*NOTE: pin 3 was rewired as speaker out rather than ground)						
WB9HBH	Drake TR-7	4-pin	*	2	4	3	5V/1.ØmA
	(*Note: audio out from speaker line)						
NØENN	WILSON WE8ØØ	?	4	3	5	1	12.7V/5ØmA
AI6C	TR 9ØØØ	6-pin		2	1	6	8V/26mA
		mini phone	tip			sleeve	
	(NOTE: TouchTone jack was used for mike/PTT)						
KØHOA	TR 78ØØ	6-pin		2	1	6	Ø.1V/8.2mA
		mini phone	tip			sleeve	
WDØFHG	ICOM 29ØA	9-pin	7	2	9	4	13.4V/13mA
	(NOTE: pin 7 was connected to ext. speaker jack, pin 9 to tone input pin 1)						

TAPR wishes to thank all those participants in Beta test who submitted radio interface data and returned software questionnaires. Along with Amateurs mentioned above, we thank W3VS, W6LLE, N9ATM, WB9BWN, WB9CNE, W9TD, WA9WXC, WA9YKK, NØCCZ, KØNN, and WØQKA for their radio interface information.

APPENDIX E

SPECIAL FUNCTIONS

Trace Function

The trace function is a protocol debugging tool. It allows the user 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. Trace options are individually enabled with the TRACE command.

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. Be aware 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 the section on Editing Commands in Chapter III for fast relief from this condition.

The frame dump output contains three sections and will be properly formatted if the screen width is at least 8Ø. Unless the option to dump the entire frame is enabled, the trace dump will show the header only.

The first section (leftmost column) contains the hex 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-1933 chip. Convert Øs 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. People wishing to modify TNC software or examine hardware I/O addresses 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 (we don't really expect any, but it's always wise to be prepared...).

The debug program is entered by typing the character specified by the DEBUG command, by default <ctl-E>. The same character is used to exit the debug program. Following the exit command the message "Bye" will be displayed.

When debug mode is entered, 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 hex and the hex prefix ($) is optional. This is different from the convention employed in 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 aren't executed. Parsing of numeric input stops when an invalid (non-numeric) character is encountered.

Following is a summary of debug mode commands. All commands are started with a special command character and terminated by a <cr>. The command characters are @, #, >, T, L, and Q. User-supplied variables are identified here in lower case and explained in the following text. Material enclosed in square brackets is explanatory.

Display Storage

@address

where address is the address whose contents you want displayed.

The program will respond with

address content [no <cr>]

Content is the value read by the processor when the specified location is addressed. (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. This mode is terminated by typing "Q".

Example:

```
:@ACØØ
ACØØ 56
ACØ1 41
ACØ2 4C Q
:
```

Following the prompt symbol, the user typed the characters "@ACØØ" followed by three <cr>s. This requested the contents of three bytes starting at hex ACØØ. The storage asked for contained the ASCII characters "VAL" which was sent to the terminal as 6 ASCII characters showing the hex representation of the storage content. The display function was terminated by typing "Q" and a final <cr>.

There is no restriction as to what address you specify. You can specify RAM, ROM, some hardware device, or a non-existent address. The result of asking for the contents of a non-existent address is not predictable and probably not very useful. You should be aware 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 hex representation of one byte of data. The program will respond by prompting you with the next address and an "=" sign:

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

Subsequent addresses can be loaded by typing only the data to be stored. Typing a <cr> only will enter Ø into the location. This mode is terminated by typing "Q".

If you try to write to a non-existent 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:

```
:@1FØØ=48
1FØ1=49
1FØ2=Q
:
```

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

Registers

Contents of registers as they appeared just prior to entering debug mode can be examined and changed. To examine the register contents:

```
#register
```

Here, register is a register code as follows: C (condition code), A, B, D (A|B), 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

```
#register content
:
```

To alter register contents:

#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

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 a 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.

APPENDIX F

ASSEMBLY

TAPR TNC - Parts List

Please check the enclosed parts with this list. Check off each item in the space ( ) provided. The number following the check-off space is the quantity required.

The TAPR part numbers referenced below are not necessarily marked on the parts. Rather, they are provided for reference when ordering replacement parts from TAPR.

Parts Sorting

As you sort the parts in the parts list, you may find it convenient to place them in a muffin tin, egg carton or other compartmentalized container for ready access.

Resistors

1/4 watt, 5%

Check	Qty	Value	Color code	TAPR P/N
(1)	1	200 ohm	(red-black-brown-gold)	CFR1/4-201
(12)	12	330 ohm	(orange-orange-brown-gold)	CFR1/4-331
(1)	1	470 ohm	(yellow-violet-brown-gold)	CFR1/4-471
(1)	1	560 ohm	(green-blue-brown-gold)	CFR1/4-561
(5)	5	680 ohm	(blue-grey-brown-gold)	CFR1/4-681
(7)	7	1k ohm	(brown-black-red-gold)	CFR1/4-102
(1)	1	3.3k ohm	(orange-orange-red-gold)	CFR1/4-332
(14)	14	4.7k ohm	(yellow-violet-red-gold)	CFR1/4-472
(1)	1	6.8k ohm	(blue-grey-red-gold)	CFR1/4-682
(13)	13	10k ohm	(brown-black-orange-gold)	CFR1/4-103
(1)	1	30k ohm	(orange-black-orange-gold)	CFR1/4-303
(3)	3	33k ohm	(orange-orange-orange-gold)	CFR1/4-333
(2)	2	39k ohm	(orange-white-orange-gold)	CFR1/4-393
(1)	1	68k ohm	(blue-grey-orange-gold)	CFR1/4-683
(5)	5	100k ohm	(brown-black-yellow-gold)	CFR1/4-104
(1)	1	470k ohm	(yellow-violet-yellow-gold)	CFR1/4-474
(1)	1	510k ohm	(green-brown-yellow-gold)	CFR1/4-514

1/8 watt, 1% — TAPR P/N

(4)	4	10.0k ohm	(brown-black-black-red-brown)	CFR1/8-1002
(2)	2	16.2k ohm	(brown-blue-red-red-brown)	CFR1/8-1622
(1)	1	22.1k ohm	(red-red-brown-red-brown)	CFR1/8-2212
(1)	1	23.7k ohm	(red-orange-violet-red-brown)	CFR1/8-2372
(1)	1	37.4k ohm	(orange-violet-yellow-red-brown)	CFR1/8-3742
(1)	1	47.5k ohm	(yellow-violet-green-red-brown)	CFR1/8-4752
(1)	1	57.6k ohm	(green-violet-blue-red-brown)	CFR1/8-5762
(1)	1	63.4k ohm	(blue-orange-yellow-red-brown)	CFR1/8-6342
(1)	1	100.k ohm	(brown-black-black-orange-brown)	CFR1/8-1003

Trimpots

(2)	2	10k ohm	(68WR-10K)	68WR-103
(1)	1	20k ohm	(68WR-20K)	68WR-203
(1)	1	50k ohm	(68WR-50K)	68WR-503
(1)	1	100k ohm	(68WR-100K)	68WR-104

Capacitors

Ceramic Disc

Capacitors may be marked in various ways. The typical markings are given but may vary. Find all that match the typical markings given and the remaining ones, if any, should become apparent by elimination.

TAPR P/N

(3)	3	330 pF	(330 or 331)	DISC-331
(2)	2	1500 pF	(.0015 or 152)	DISC-152
(7)	7	0.01 uF	(.01 or 103)	DISC-103
(3)	2	0.05 uF	(.05 or 503)	DISC-503
(36)	36	0.1 uF	(.1 or 104)	DISC-104
(1)	1	0.15 uF	(.15 or 154)	DISC-154

Mylar or Monolithic

(1)	1	2200 pF	(.0022 or 222)	MONO-222
(1)	1	0.0047 uF	(.0047 or 472)	MONO-472
(2)	2	0.022 uF	(.022 or 223)	MONO-223

Electrolytic

The electrolytic capacitors may have voltage ratings equal to or greater than those listed below, but not less.

			TAPR P/N
(2)	2	1 uF, 16 volt	RAD16V-1Ø5
(3)	3	1Ø uF, 16 volt	RAD16V-1Ø6
(1)	1	1ØØ uF, 6 volt	RADØ6V-1Ø7
(1)	1	33Ø uF, 6 volt	RADØ6V-337
(2)	2	47Ø uF, 25 volt	RAD25V-477
(1)	1	22ØØ uF, 16 volt	RAD16V-228

Diodes

Diodes are marked with the value directly on the body of the part. Some diodes are very small, so a magnifying glass may be helpful!

Note that the banded or tapered end is the cathode on standard diodes and the flat side is the cathode on the LEDs. The LEDs may not have a part number on them.

			TAPR P/N
(6)	6	1N4ØØ1	1N4ØØ1
(2)	2	1N4148 (these are the smallest diodes)	1N4148
(1)	1	1N4752 Zener	1N4752
(4)	4	1N54ØØ (these are the largest diodes)	1N54ØØ
(8)	8	LED, red	SR-5Ø3D

Transistor and Voltage Regulators

(2)	2	VN1ØKM or VN1ØKMA Power FET TX101	VN1ØKM
(1)	1	3Ø9 or 34Ø-5 or 78Ø5 +5 volt regulator	LM3Ø9K
(1)	1	78LØ5 +5 volt regulator	UA78LØ5
(1)	1	7812 +12 volt regulator	UA7812
(1)	1	79LØ5 or LM32Ø -5 volt regulator	UA79LØ5
(1)	1	7912 -12 volt regulator	UA7912

Integrated Circuits

Integrated circuits may come from various manufacturers and may have differing prefixes and/or suffixes. For example, if the part is listed as a 74LS00, it may be marked SN74LS00N or MC74LS00P or DM74LS00N or F74LS00P or a similar variation. The key is that the sequence 74LS00 appears in the part number. A four-digit number, such as 8304, indicates the year and week of manufacture and should not be confused with being a part number.

WARNING: Do not handle the ICs at this time! Carefully remove the black foam carrier with ICs from the anti-static bag and verify the ICs against this list. Then return the foam with ICs to the anti-static bag. Do not touch any of the ICs. This precaution cannot be overemphasized!

			TAPR P/N
( )	1	74LS00 TTL NAND gate	74LS00
( )	1	74LS04 TTL Hex inverter	74LS04
( )	2	7406 TTL OC inverter	7406
( )	1	74LS10 TTL NAND gate	74LS10
( )	1	74LS86 EX-OR gate	74LS86
( )	1	74LS393 TTL divider	74LS393
( )	1	1488 TTL to RS232 driver	MC1488
( )	1	1489 or 1489A RS232 to TTL receiver	MC1489
( )	1	82S123, 74S288 or 7603 Bipolar PROM	74S288
( )	1	555 Timer	NE555V
( )	2	1458 dual Op Amp	LM1458
( )	1	2206 FSK Modulator	XR2206
( )	1	2211 FSK Demodulator	XR2211
( )	1	2212 NOVRAM	XD2212
( )	1	MF-10 CMOS filter	MF-10
( )	1	6809 NMOS microprocessor	MC6809
( )	1	6520 or 6821 NMOS parallel port	MC6821
( )	1	6522 NMOS parallel port/timer	SY6522
( )	1	6551 NMOS Uart	SY6551
( )	1	1933 or 1935 NMOS HDLC controller	WD1933
( )	1	6264 (or equiv.) MOS 8k by 8 RAM	8KRAM
( )	4	2764 (labelled 8, A, C, E)	2764-( )

Sockets			TAPR P/N
(3)	3	8-pin DIP	DIPS-Ø8
(11)	11	14-pin DIP	DIPS-14
(5)	5	16-pin DIP	DIPS-16
(1)	1	18-pin DIP	DIPS-18
(1)	1	2Ø-pin DIP	DIPS-2Ø
(7)	7	28-pin DIP	DIPS-28
(4)	4	4Ø-pin DIP	DIPS-4Ø

Connectors

(5)	5	2-pin male header	HM-Ø2
(3)	3	3-pin male header	HM-Ø3
(1)	1	5-pin male header with "wall"	HMW-Ø5
(13)	13	Jumpers, "push-on"	JMP-Ø2
(1)	1	2Ø-pin male header with ejector/latch	HMSEL-2Ø
(1)	1	DE9S 9-pin female PC right angle	DEØ9PCR-S
(1)	1	DB25P 25-pin male PC right angle	DB25PCR-P
(1)	1	DB25S 25-pin female PC right angle	DB25PCR-S
(1)	1	7-pin male "MOLEX" power	MOLPWR-Ø7P
(1)	1	14-pin male DIP header	DIPH-14
(2)	2	16-pin male DIP header	DIPH-16

Miscellaneous

(1)	1	Crystal, 7.3728 MHz (NDK, M-tron or equal)	XTAL-Ø73
(1)	1	4-position DIP switch (CTS 2Ø6-4)	DIPSW-Ø2
(2)	2	6-32 x 3/8 screw	MS632-3/8
(2)	2	6-32 x 5/8 screw	MS632-5/8
(4)	4	6-32 hex nut	MN632
(4)	4	#6 flat washer	MW632-FLAT
(2)	2	#6 lock washer	MW632-LOCK
(1)	1	MOLEX Power connector housing	MOLPWR-Ø7S
(7)	7	MOLEX Power connector pins	MOLPWR-PINS
(1)	1	DE9P 9-pin male connector	DEØ9-P
(1)	1	3 foot length of stranded hookup wire	W2ØS-36

The following components are packed separately

(1)	1	Heat sink	HSTO3-1
(1)	1	PC board, TNC rev 3	TNCPC-R3
( )	1	Power Transformer - 115 v 6Ø Hz (4111)	XFMR-TNC-2
		or	
		Power Transformer - 23Ø v 5Ø Hz (4425)	XFMR-TNC-3
(1)	1	Backshell for DE9P	DEØ9-BACK

TAPR TNC - Assembly Directions

This section will proceed in a step-by-step fashion. Please mount and solder the components to the board in the order given. It has been tested on numerous units and found to be a very efficient order for assembly.

When mounting axial lead discrete components, such as resistors and certain diodes, grasp the body of the part in one hand and bend the leads of the device with the other, such that the leads will pass through the holes provided on the board. Note that all axial lead components except 1N5400 diodes are on 0.5 inch lead centers. Place the part so the body is parallel to, and flat against, the circuit board unless instructed otherwise. Next, bend the leads on the bottom of the board slightly to secure the part. Then proceed to mount the next component. Do not solder the part until directed. (See Fig. F.1.)

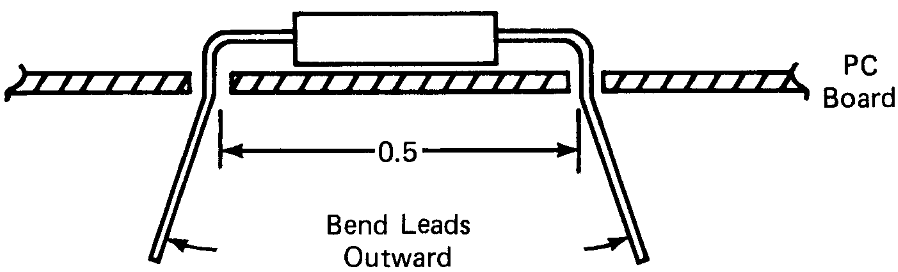


Fig. F.1 Axial Lead Component Mounting

Soldering

When soldering to the printed circuit board, use only a <u>very small</u> tipped controlled temperature (700 degrees F. maximum) soldering iron and fine (.050 inch) rosin-core solder. 63/37 (eutectic) solder is preferred, but 60/40 is acceptable.

To ensure the joint is properly soldered, the iron should be placed so that it contacts the pad on the board and the lead to be soldered. Solder should then be applied to the pad and the opposite side of the lead from the iron. Thus, the iron must heat both the pad and the lead to cause the solder to melt. This helps prevent cold solder joints. (See Fig. F.2.) Also, keep a wet sponge handy and wipe the soldering iron tip on it frequently to keep it (and your connections) clean.

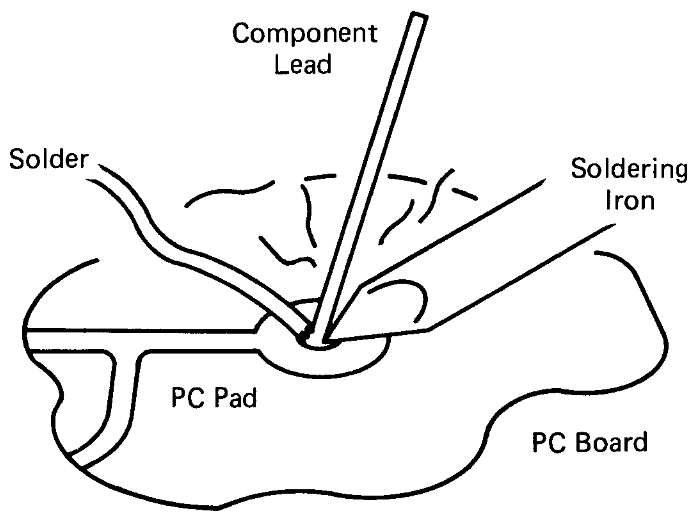


Fig. F.2 Soldering

Care must be taken to ensure that the pad is not overheated, or it may lift from the PC board! Apply heat only when ready to solder. Do not apply heat longer than necessary to complete a good joint, but be sure to heat adequately for the solder to flow completely.

NOTE: All modern components are designed for soldering and you should not worry about 3-5 seconds of heating.

Board Mapping

Place the circuit board on a clean working surface. Orient the board so the lettering is in normal reading position. For purposes of description, the board consists of four sectors. The top of the board is the edge nearest the memories (U7-U12). The bottom, left and right follow in natural order. The upper left is quadrant one, the upper right is quadrant two, the lower left is quadrant three and the lower right is quadrant four. The board assembly will proceed by quadrants. (See Fig. F.3.)

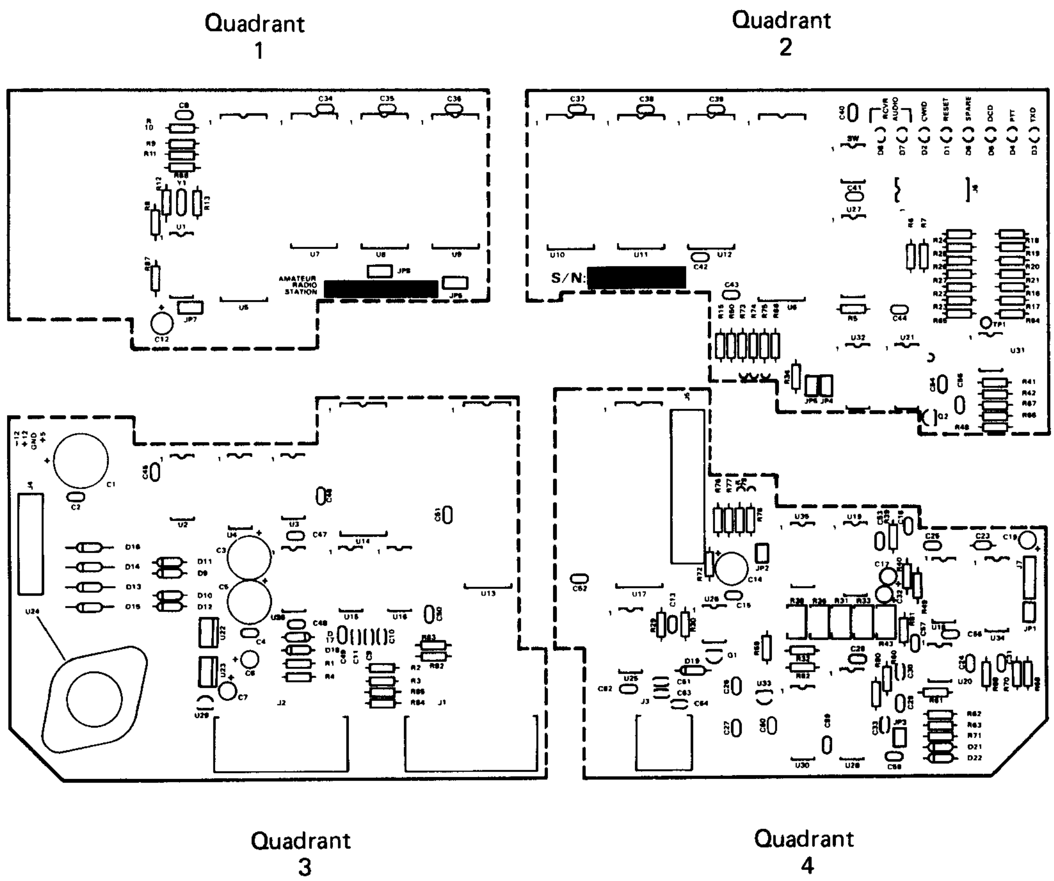


Fig. F.3 Board Mapping

IC Sockets

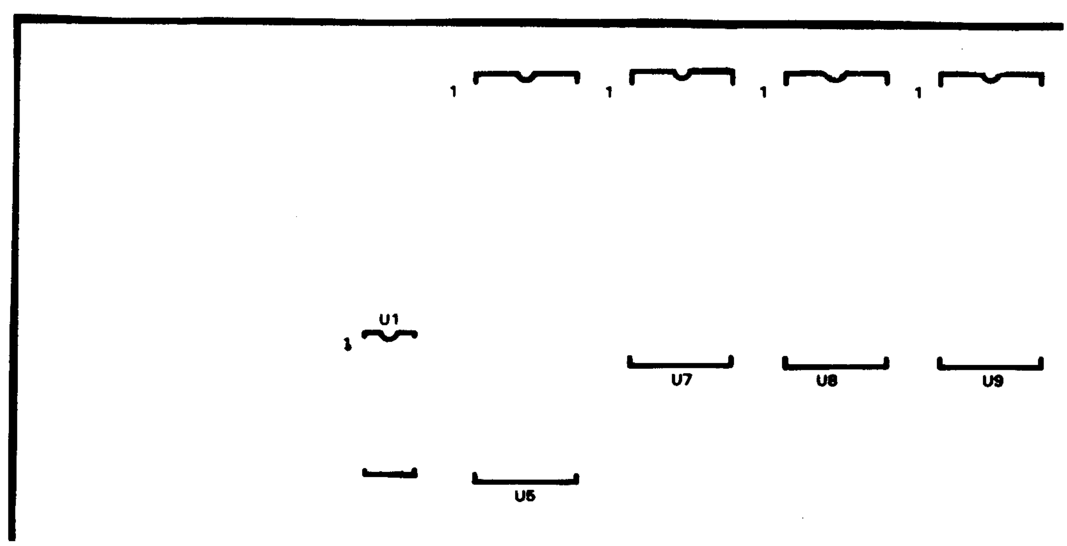
NOTE: If any socket pins are bent, <u>carefully</u> straighten them with a pair of long-nose pliers.

When installing IC sockets, tack-solder two diagonally opposite corners first (such as pin 1 and pin 8 on a 14-pin socket). Double check to ensure that the socket is seated properly against the board with the notch, bevelled corner or "1" nearest pin 1 (pin 1 is the upper left corner when viewed from the top of the board on all socketed IC positions) and that <u>all</u> IC socket pins are showing on the solder side of the board. Then solder the remaining pins of that socket before proceeding to the next one.

NOTE: Take care <u>now</u> to avoid solder bridges!

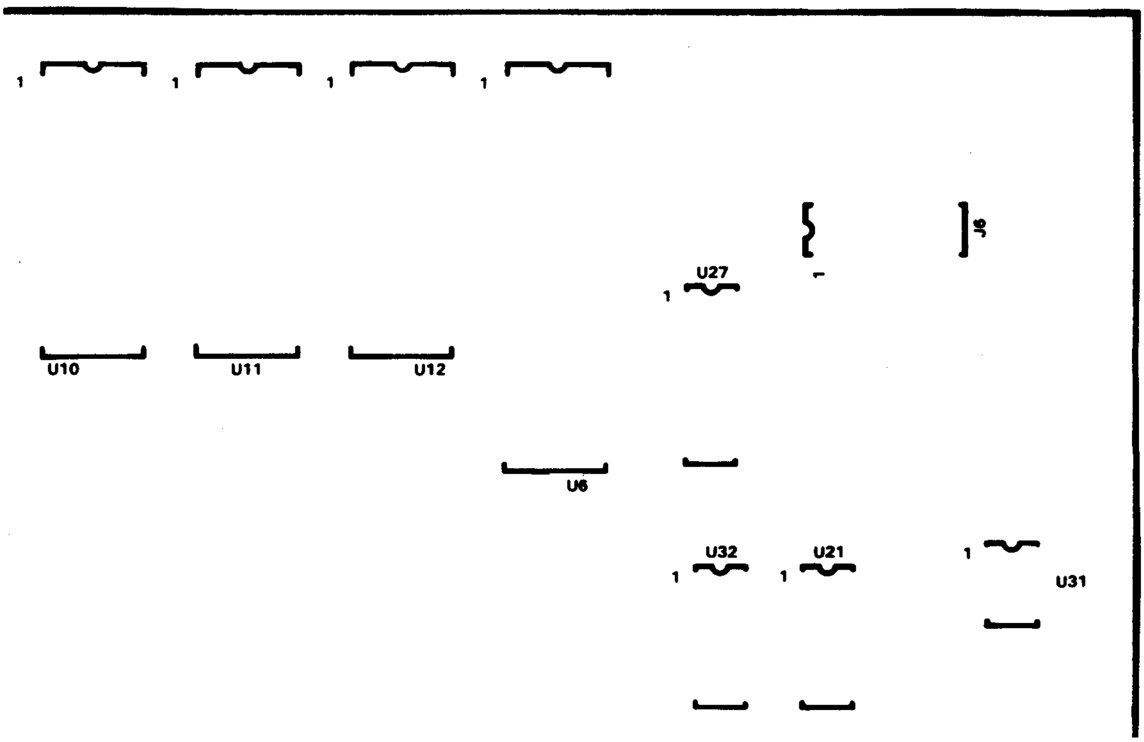
(quadrant 1)

- ( ) U1 14-pin
- ( ) U5 4Ø-pin
- ( ) U7 28-pin
- ( ) U8 28-pin
- ( ) U9 28-pin



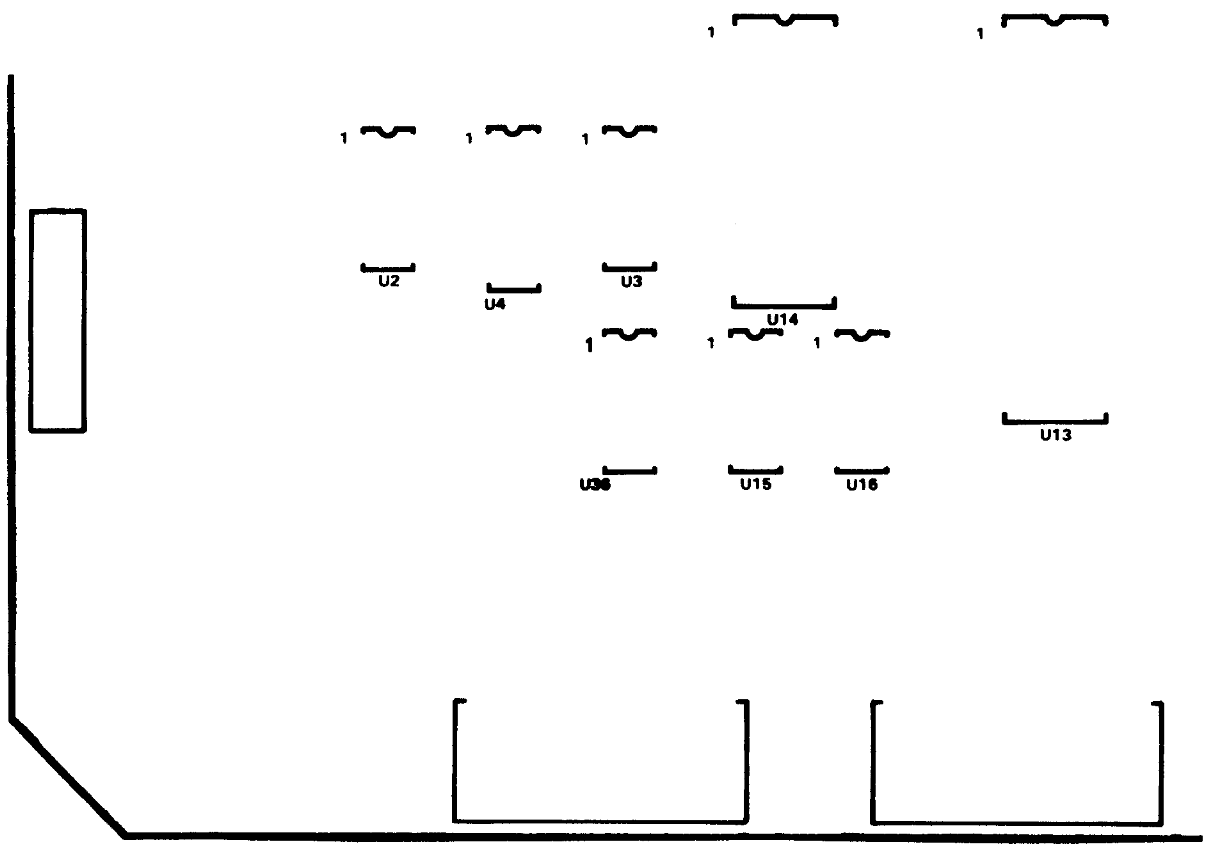
(quadrant 2)

( ) U10 28-pin
( ) U11 28-pin
( ) U12 28-pin
( ) U6 40-pin
( ) U27 18-pin
( ) J6 16-pin
( ) U32 14-pin
( ) U21 14-pin
( ) U31 8-pin



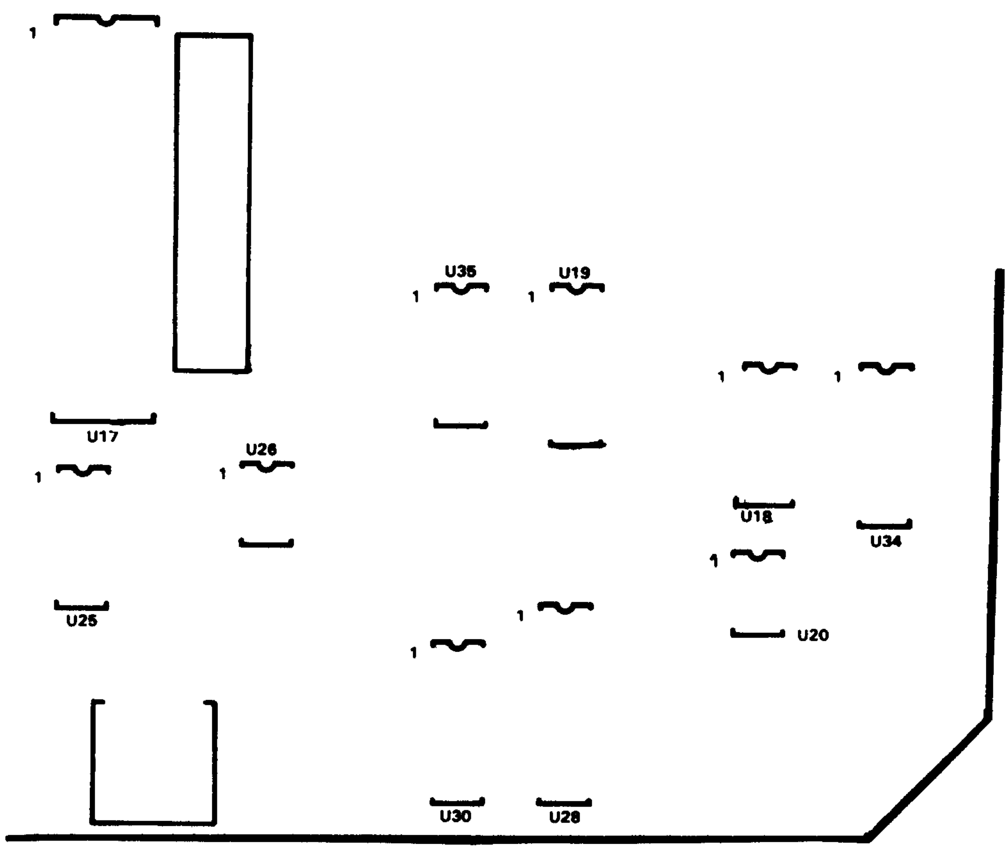
(quadrant 3)

( ) U2 14-pin
( ) U4 16-pin
( ) U3 14-pin
( ) U14 28-pin
( ) U13 40-pin
( ) U36 14-pin
( ) U15 14-pin
( ) U16 14-pin



(quadrant 4)

- ( ) U17 4Ø-pin
- ( ) U35 14-pin
- ( ) U19 16-pin
- ( ) U18 14-pin
- ( ) U34 16-pin
- ( ) U25 14-pin
- ( ) U26 8-pin
- ( ) U3Ø 16-pin
- ( ) U28 2Ø-pin
- ( ) U2Ø 8-pin



Now check your work. All leads should be soldered. There should be no solder bridges (a blob of solder that shorts two adjacent soldered connections) or cold (grey and/or grainy-looking) solder connections.

( ) OK so far.

Resistors

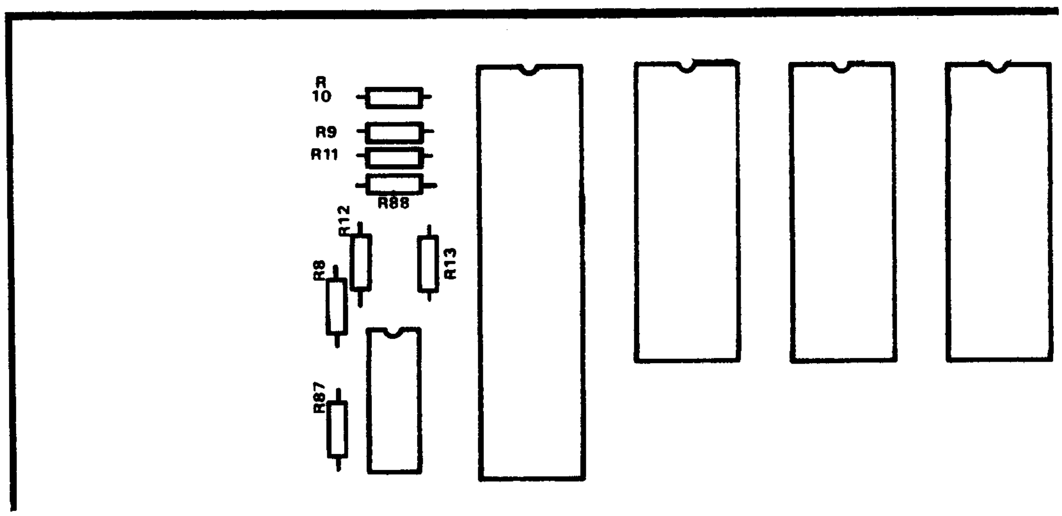
Install the following 5% resistors on the board:

(quadrant 1)

( ) R1Ø 4.7k (yellow-violet-red-gold)
( ) R9 4.7k (yellow-violet-red-gold)
( ) R11 4.7k (yellow-violet-red-gold)
( ) R88 4.7k (yellow-violet-red-gold)
( ) R12 1.Øk (brown-black-red-gold)
( ) R13 1.Øk (brown-black-red-gold)
( ) R8 4.7k (yellow-violet-red-gold)
( ) R87 47Ø (yellow-violet-brown-gold)

WARNING: Be careful when clipping leads, as they may have a tendency to fly towards your eyes! Take appropriate precautions (grasp leads and provide eye protection).

( ) Solder and clip the leads (16 total).



(quadrant 2)

( ) R15 1ØØk (brown-black-yellow-gold)
( ) R5Ø 4.7k (yellow-violet-red-gold)
( ) R73 4.7k (yellow-violet-red-gold)
( ) R74 4.7k (yellow-violet-red-gold)
( ) R75 4.7k (yellow-violet-red-gold)
( ) R86 4.7k (yellow-violet-red-gold)

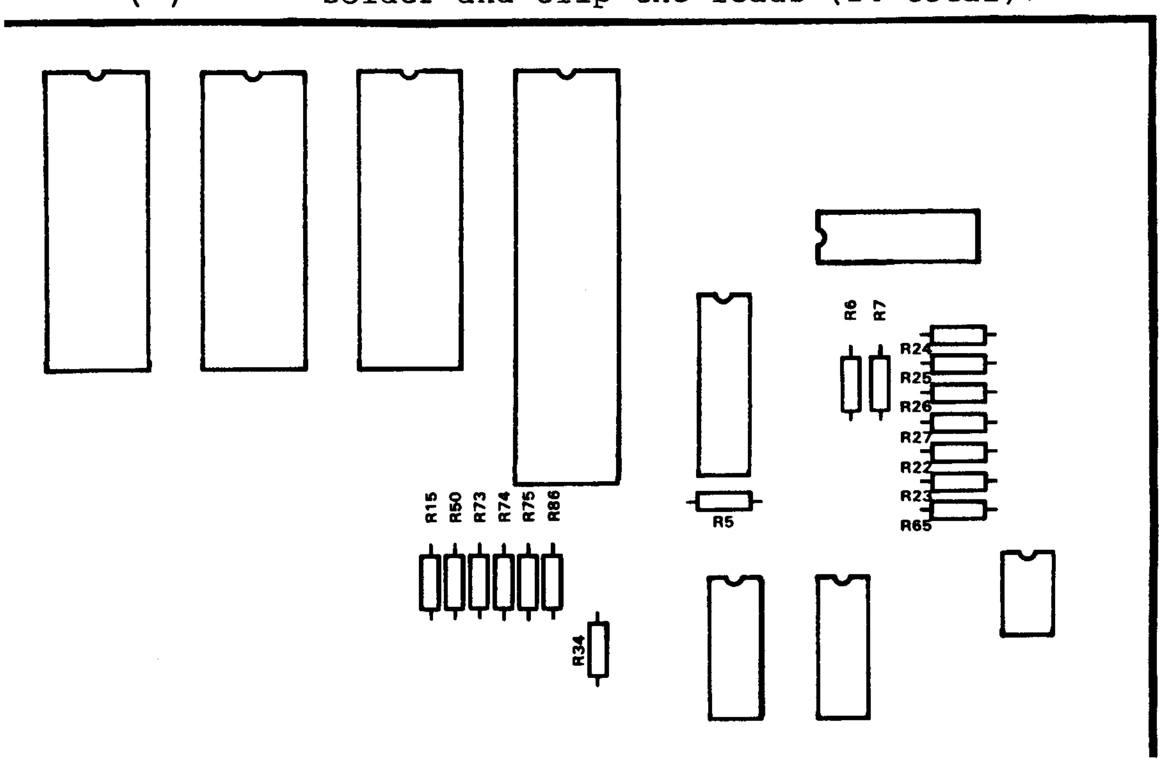
( ) Solder and clip the leads (12 total).

( ) R34 4.7k (yellow-violet-red-gold)
( ) R5 1Øk (brown-black-orange-gold)
( ) R6 1Øk (brown-black-orange-gold)
( ) R7 1Øk (brown-black-orange-gold)

( ) Solder and clip the leads (8 total).

( ) R24 33Ø (orange-orange-brown-gold)
( ) R25 33Ø (orange-orange-brown-gold)
( ) R26 33Ø (orange-orange-brown-gold)
( ) R27 33Ø (orange-orange-brown-gold)
( ) R22 33Ø (orange-orange-brown-gold)
( ) R23 33Ø (orange-orange-brown-gold)
( ) R65 1k (brown-black-red-gold)

( ) Solder and clip the leads (14 total).



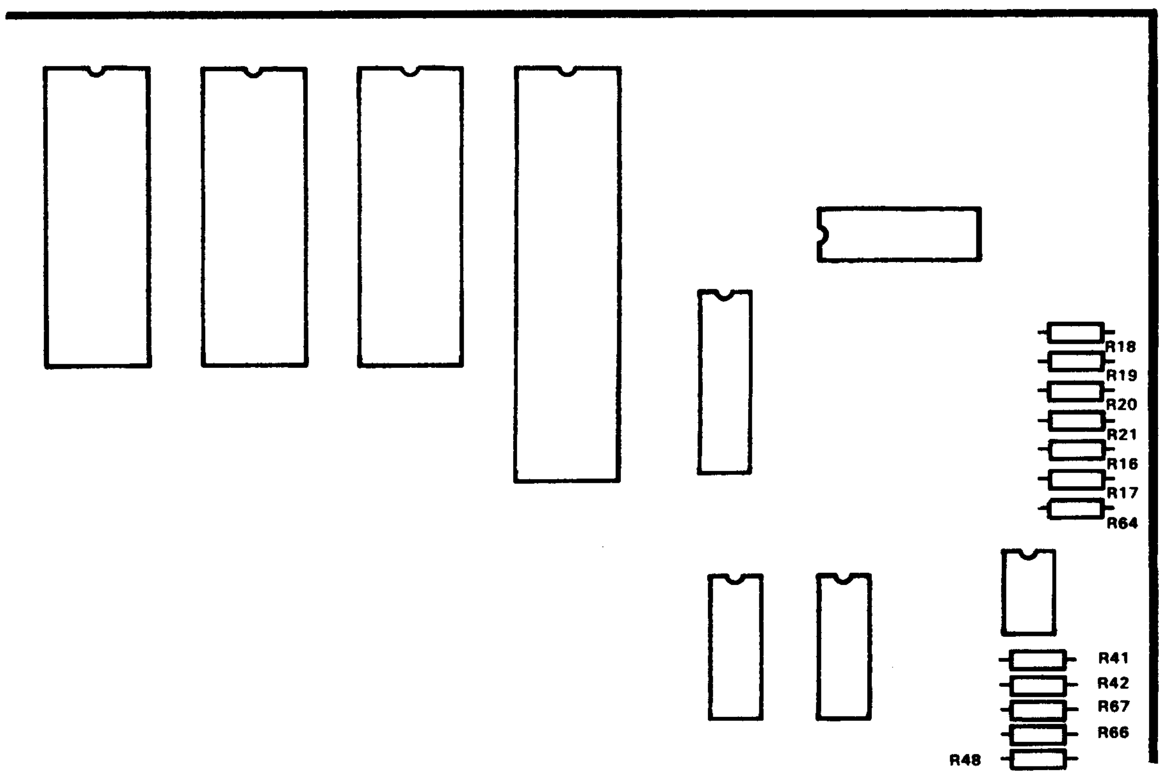
(quadrant 2 - continued)

( ) R18 33Ø (orange-orange-brown-gold)
( ) R19 33Ø (orange-orange-brown-gold)
( ) R2Ø 33Ø (orange-orange-brown-gold)
( ) R21 33Ø (orange-orange-brown-gold)
( ) R16 33Ø (orange-orange-brown-gold)
( ) R17 33Ø (orange-orange-brown-gold)
( ) R64 1k (brown-black-red-gold)

( ) Solder and clip the leads (14 total).

( ) R41 3Øk (orange-black-orange-gold)
( ) R42 1k (brown-black-red-gold)
( ) R67 3.3k (orange-orange-red-gold)
( ) R66 68k (blue-grey-orange-gold)
( ) R48 47Øk (yellow-violet-yellow-gold)

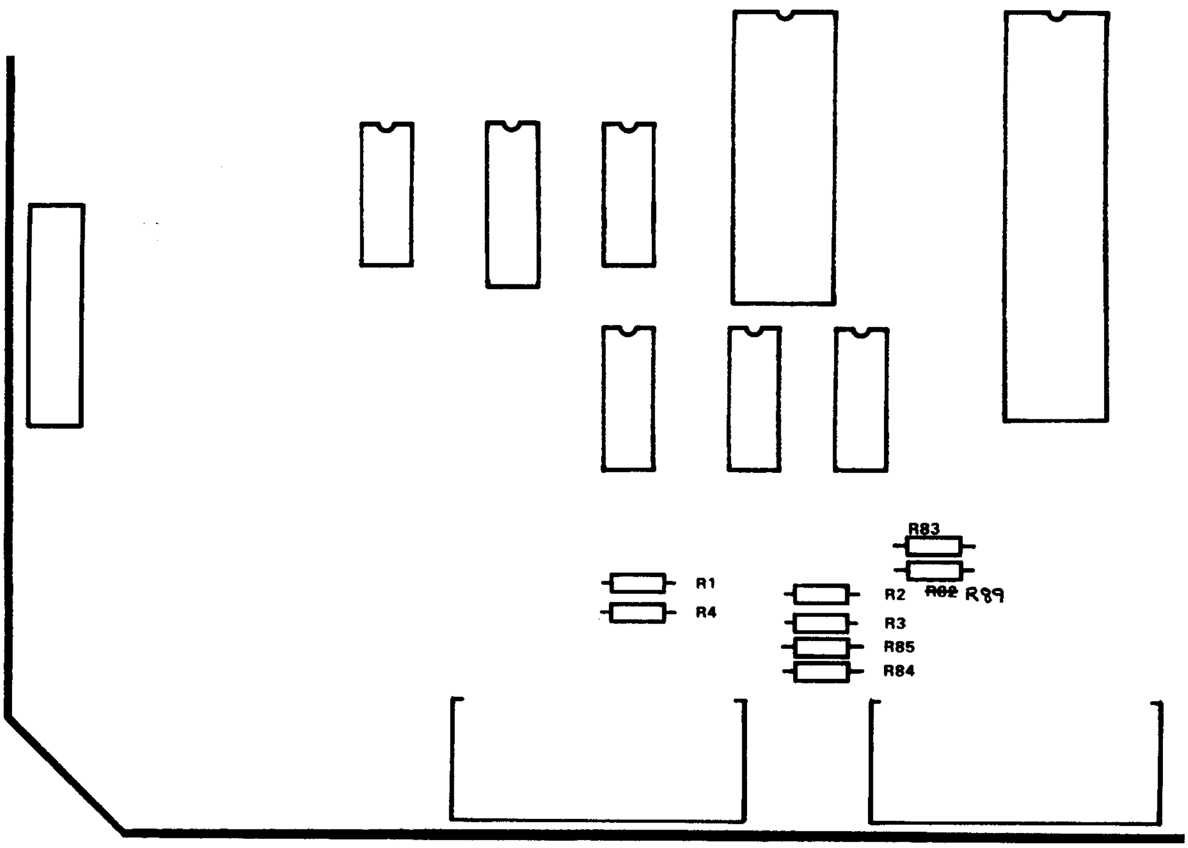
( ) Solder and clip the leads (1Ø total).



(quadrant 3)

( ) R1 33 k (orange-orange-orange-gold)
( ) R4 33 k (orange-orange-orange-gold)
( ) R2 6.8k (blue-grey-red-gold)
( ) R3 33 k (orange-orange-orange-gold)
( ) R85 10 k (brown-black-orange-gold)
( ) R84 10 k (brown-black-orange-gold)
( ) R83 10 k (brown-black-orange-gold)
( ) R82 10 k (brown-black-orange-gold)

( ) Solder and clip the leads (16 total).



(quadrant 4)

( ) R76 680 (blue-grey-brown-gold)
( ) R77 680 (blue-grey-brown-gold)
( ) R79 680 (blue-grey-brown-gold)
( ) R78 680 (blue-grey-brown-gold)

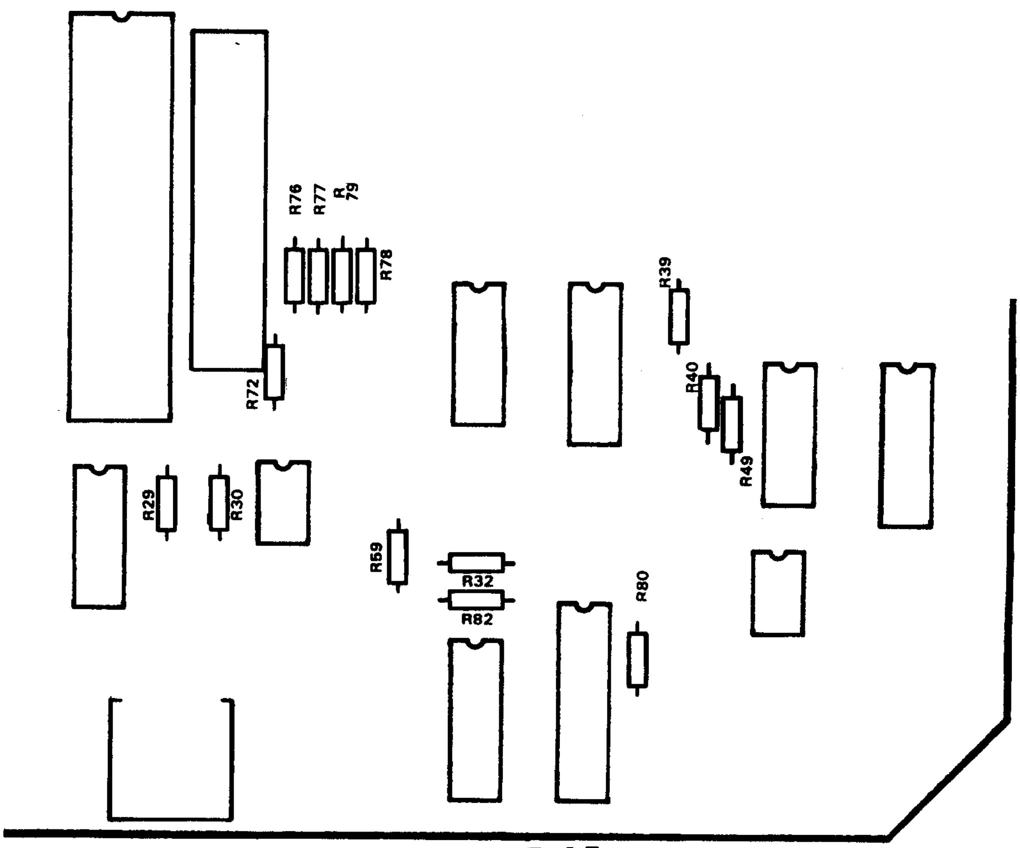
( ) Solder and clip the leads (8 total).

( ) R72 4.7k (yellow-violet-red-gold)
( ) R39 200 (red-black-brown-gold)
( ) R40 10k (brown-black-orange-gold)
( ) R49 10k (brown-black-orange-gold)

( ) Solder and clip the leads (8 total).

( ) R29 4.7k (yellow-violet-red-gold)
( ) R30 10k (brown-black-orange-gold)
( ) R59 560 -- not installed at this time.
(normally not needed, see page 2-3)
( ) R32 4.7k (yellow-violet-red-gold)
( ) R82 100k (brown-black-yellow-gold)
( ) R80 1k (brown-black-red-gold)

( ) Solder and clip the leads (10 or 12 total).



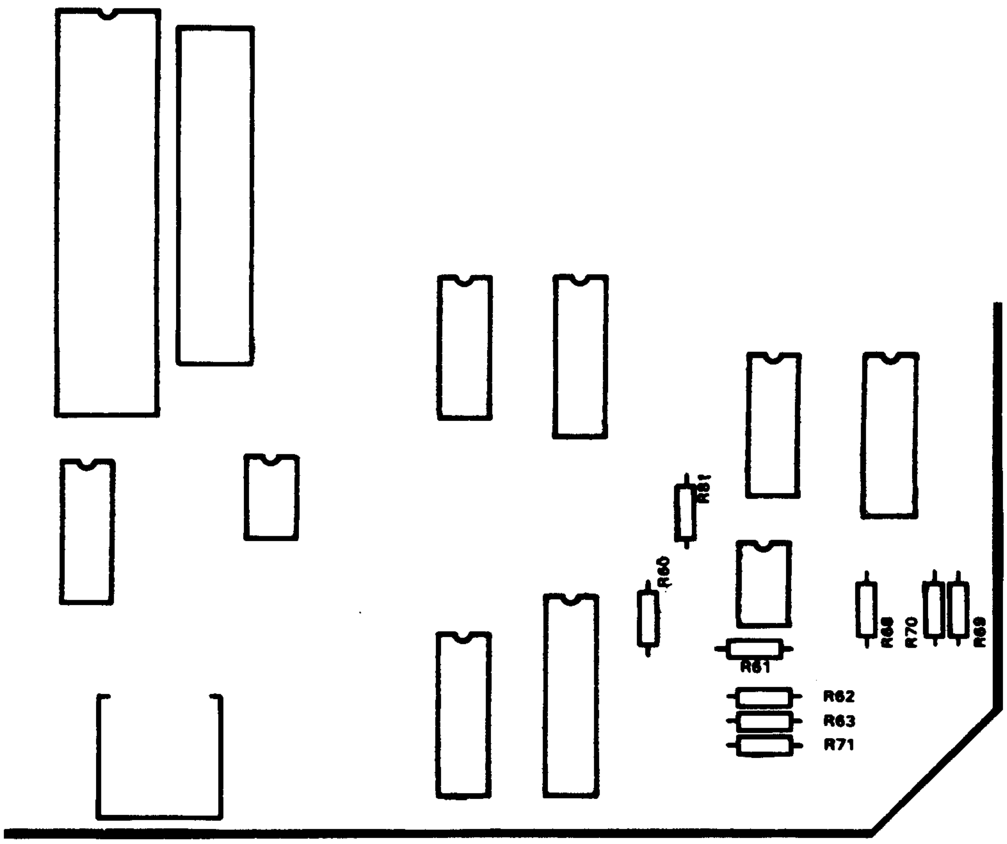
(quadrant 4 - continued)

( ) R60 10k (brown-black-orange-gold)
( ) R81 680 (blue-grey-brown-gold)
( ) R61 10k (brown-black-orange-gold)
( ) R68 100k (brown-black-yellow-gold)
( ) R70 100k (brown-black-yellow-gold)
( ) R69 10k (brown-black-orange-gold)

( ) Solder and clip the leads (12 total).

( ) R62 39k (orange-white-orange-gold)
( ) R63 39k (orange-white-orange-gold)
( ) R71 1k (brown-black-red-gold)

( ) Solder and clip the leads (6 total).



There should be three 5% resistors remaining, one 510k (green-brown-yellow-gold), one 100k (brown-black-yellow-gold), and one 560 ohm (green-blue-brown-gold).

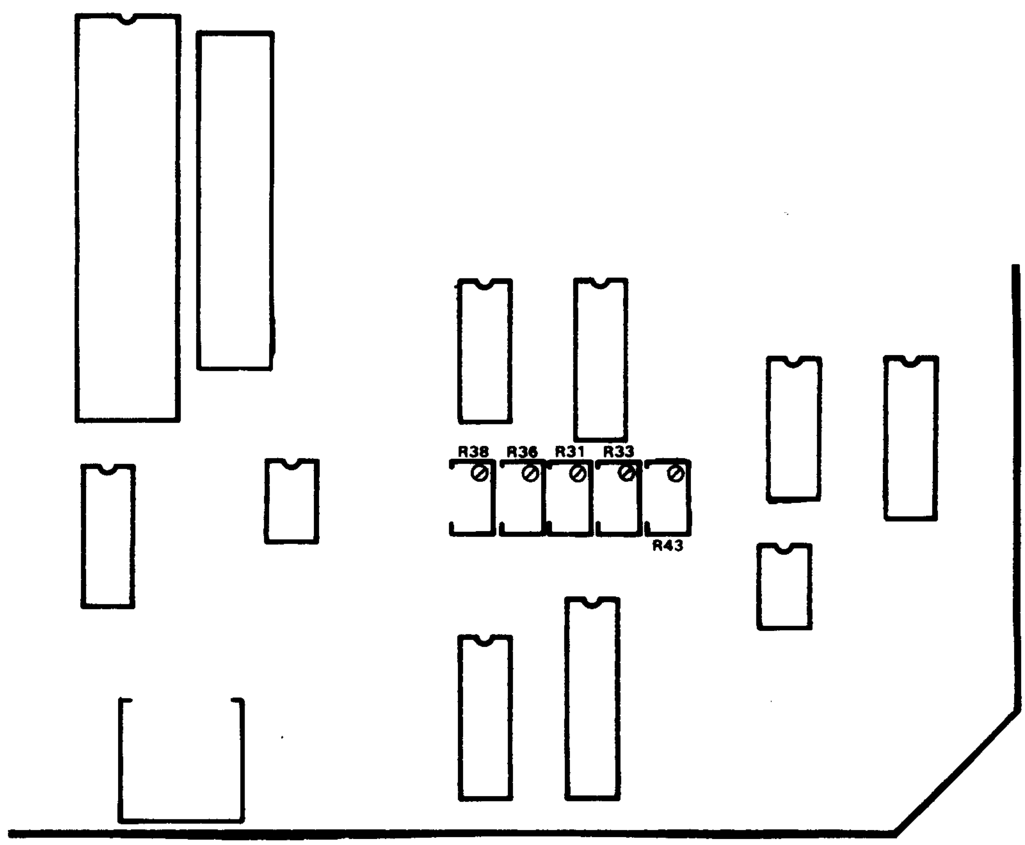
( ) Three 5% resistors remaining.

The following devices are trimpots. When you install one, tack solder the center lead to the PC board. Do not solder the other leads, nor clip any leads, until directed. This enables you to line up all the trimpots neatly before you commit any of them to a "final" position.

NOTE: Orient each trimpot with the adjustment screw towards the top of the board.

( ) R38 2Øk
( ) R36 1Øk
( ) R31 1ØØk
( ) R33 5Øk
( ) R43 1Øk

( ) Solder and clip the leads (15 total).



Now check your work. All leads should be soldered and clipped close to the bottom of the PC board. You should have all thirteen 1% and three 5% resistors remaining. There should be no solder bridges nor cold solder connections.

( ) OK so far.

Diodes

NOTE: LEDs are polarity sensitive! The flat side of the LEDs (cathode) should match the flat side of the silk screen outline.

NOTE: The LED cathode lead is the shorter of the two.

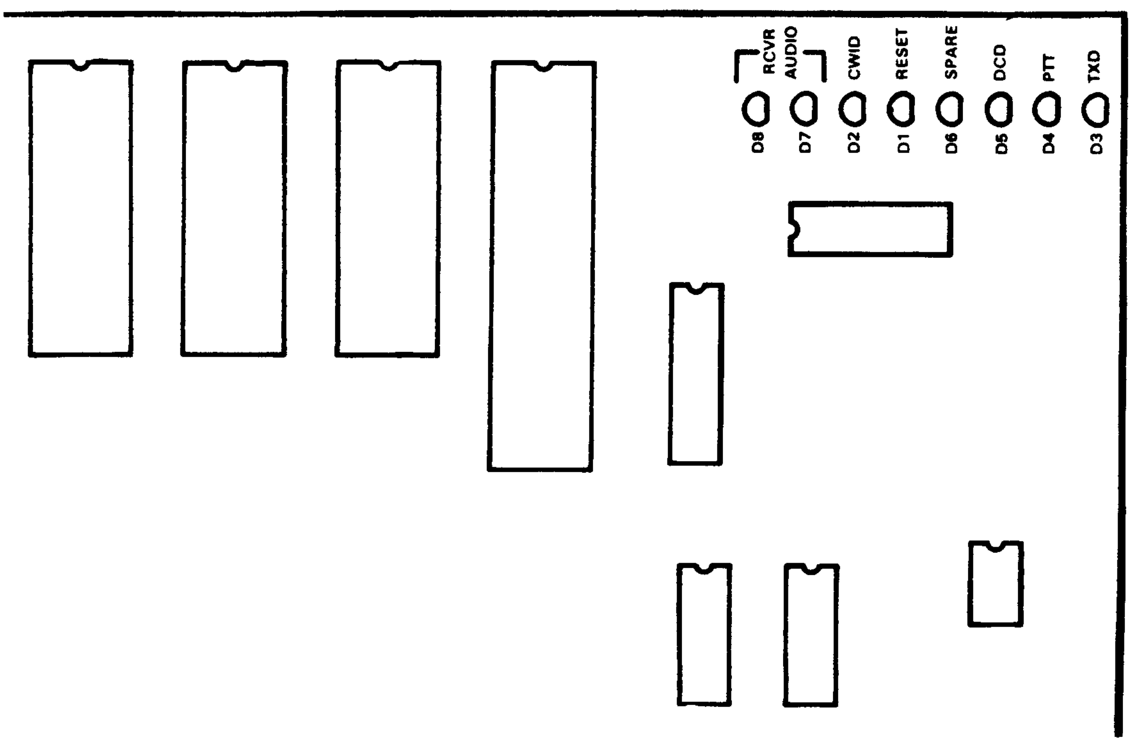
WARNING: When installing LEDS, do not bend the leads to secure the LED to the board. Bending the leads can fracture the plastic casing and cause the device to fail. Instead, hold the LED while soldering it.

(quadrant 1) -- (none)

(quadrant 2)

NOTE: All LEDs are oriented with the flat (cathode) to the right.

( ) D8 LED, red
( ) Solder and clip the leads (2 total).
( ) D7 LED, red
( ) Solder and clip the leads (2 total).
( ) D2 LED, red
( ) Solder and clip the leads (2 total).
( ) D1 LED, red
( ) Solder and clip the leads (2 total).
( ) D6 LED, red
( ) Solder and clip the leads (2 total).
( ) D5 LED, red
( ) Solder and clip the leads (2 total).
( ) D4 LED, red
( ) Solder and clip the leads (2 total).
( ) D3 LED, red
( ) Solder and clip the leads (2 total).



WARNING: Diodes are polarity sensitive! The marked (banded or tapered) end of the diodes (cathode) should match the banded end of the silk screen.

(quadrant 3)

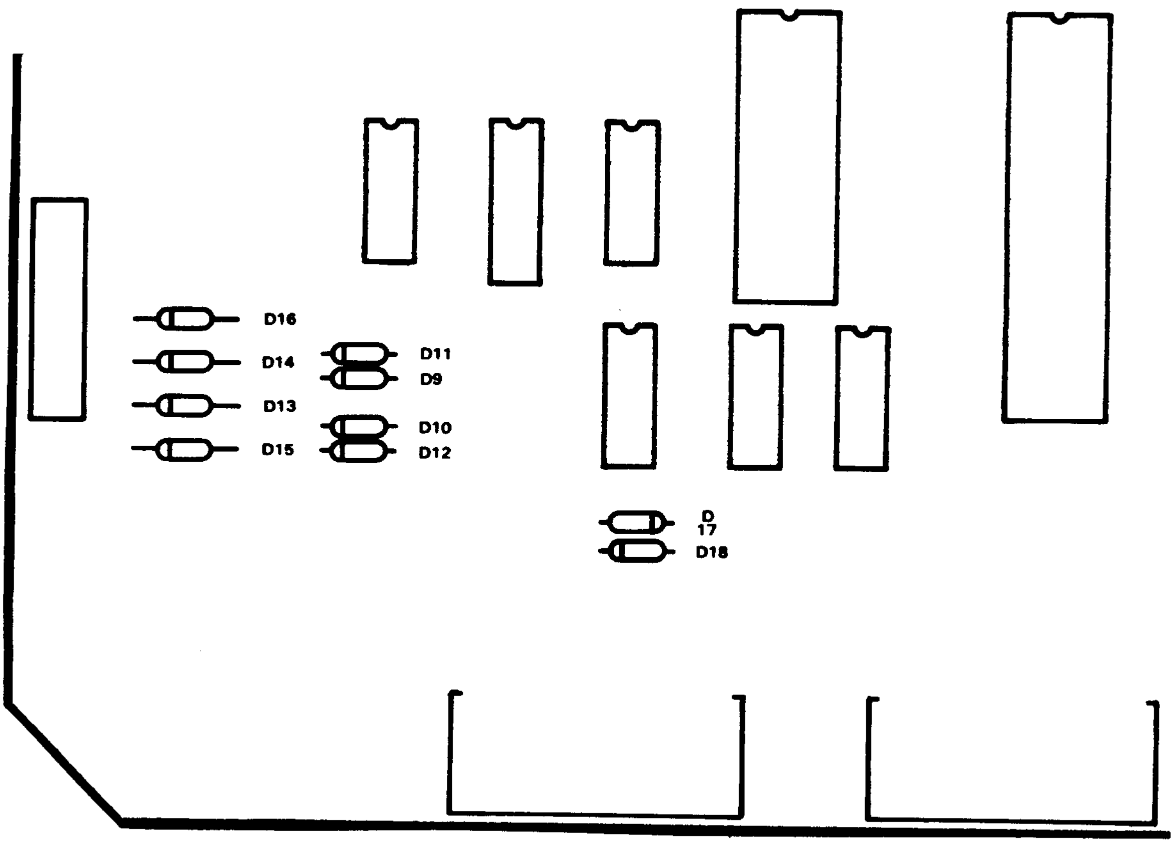
NOTE: The next 4 diodes should be placed so they are about 1/8 inch above the PC board surface.

( ) D16 1N54ØØ
( ) D14 1N54ØØ
( ) D13 1N54ØØ
( ) D15 1N54ØØ

( ) Solder and clip the leads (8 total).

( ) D11 1N4ØØ1
( ) D9 1N4ØØ1
( ) D1Ø 1N4ØØ1
( ) D12 1N4ØØ1
( ) D17 1N4ØØ1
( ) D18 1N4ØØ1

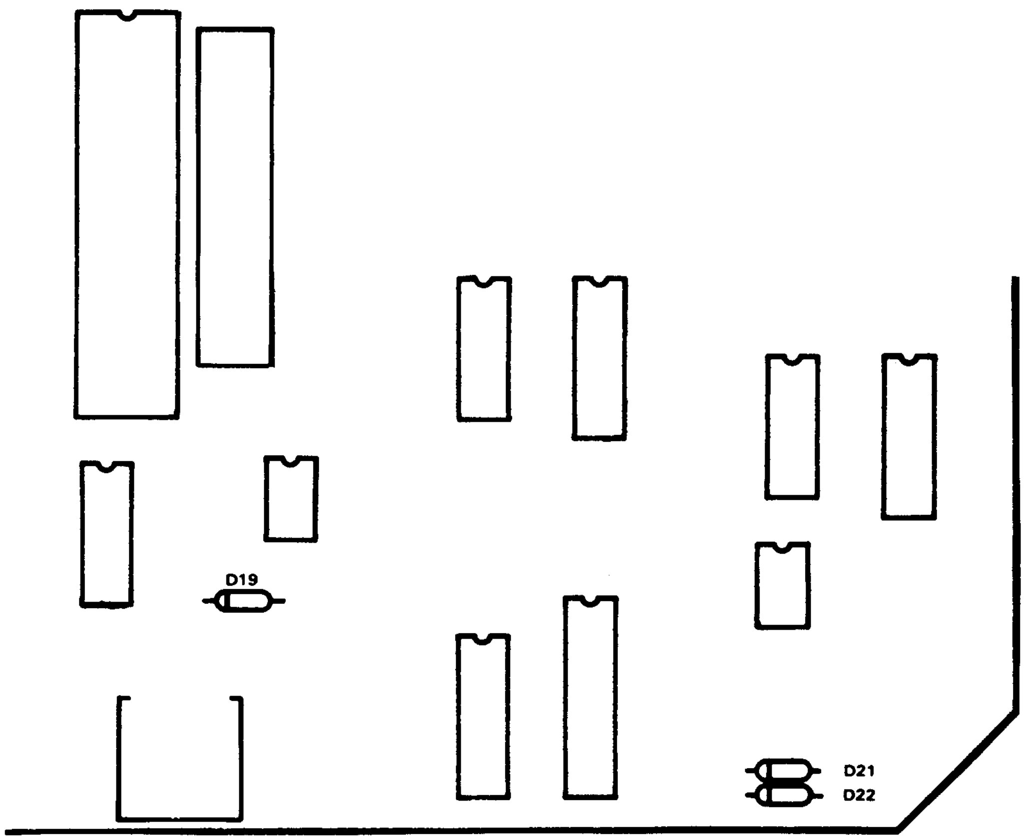
( ) Solder and clip the leads (12 total).



(quadrant 4)

( ) D19 1N4752
( ) D21 1N4148
( ) D22 1N4148

( ) Solder and clip the leads (6 total).



Now check your work. You should have no diodes nor LEDs remaining. All leads should be soldered and clipped close to the bottom of the PC board. There should be no solder bridges nor cold solder connections.

( ) OK so far.

Capacitors

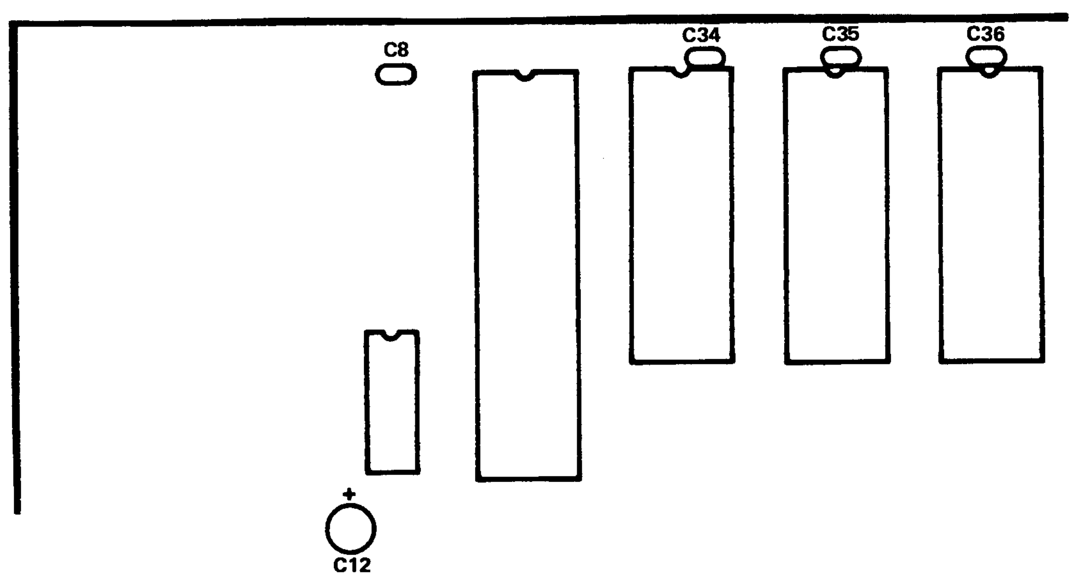
WARNING: The electrolytic capacitors are polarity sensitive. The silk screen indicates the positive lead, while most electrolytic capacitors have the negative lead marked. Be careful to ensure the parts are correctly installed or damage may occur at power up!

NOTE: Inspect your Ø.1 uF disc capacitors and choose one that has a thin profile (there may be several). Save it until specifically called for in the assembly instructions! It will be installed in a tight place on the board.

(quadrant 1)

( ) C8 Ø.1 uF
( ) C34 Ø.1 uF
( ) C35 Ø.1 uF
( ) C36 Ø.1 uF
( ) C12 1ØØ uF electrolytic

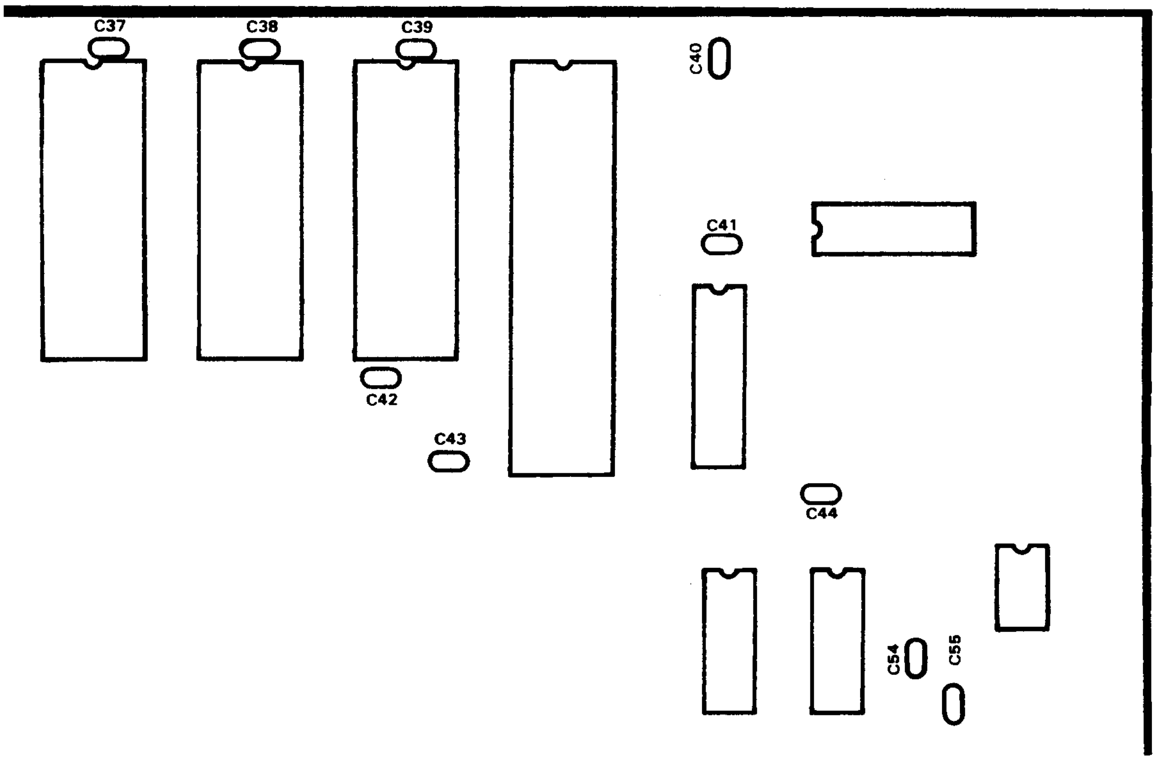
( ) Solder and clip the leads (1Ø total).



(quadrant 2)

( ) C37 Ø.1 uF
( ) C38 Ø.1 uF
( ) C39 Ø.1 uF
( ) C4Ø Ø.1 uF
( ) C41 Ø.1 uF
( ) C42 Ø.1 uF
( ) C43 Ø.1 uF
( ) C44 Ø.1 uF
( ) C54 Ø.1 uF
( ) C55 Ø.1 uF

( ) Solder and clip the leads (2Ø total).



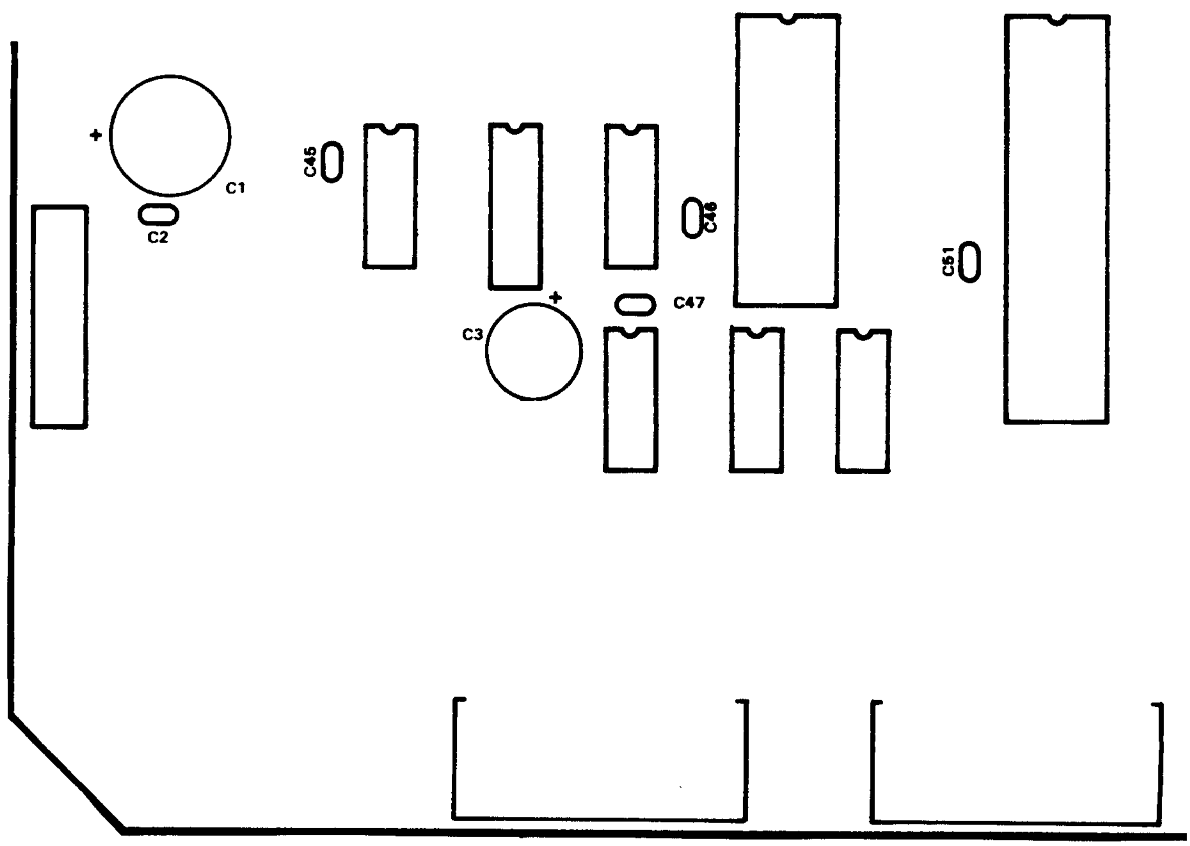
(quadrant 3)

( ) C1 2200 uF electrolytic
( ) C2 0.1 uF
( ) C45 0.1 uF

NOTE: In the following step, the + side of C3 goes towards the top of the board.

( ) C3 470 uF electrolytic
( ) C47 0.1 uF
( ) C46 0.1 uF
( ) C51 0.1 uF

( ) Solder and clip the leads (14 total).



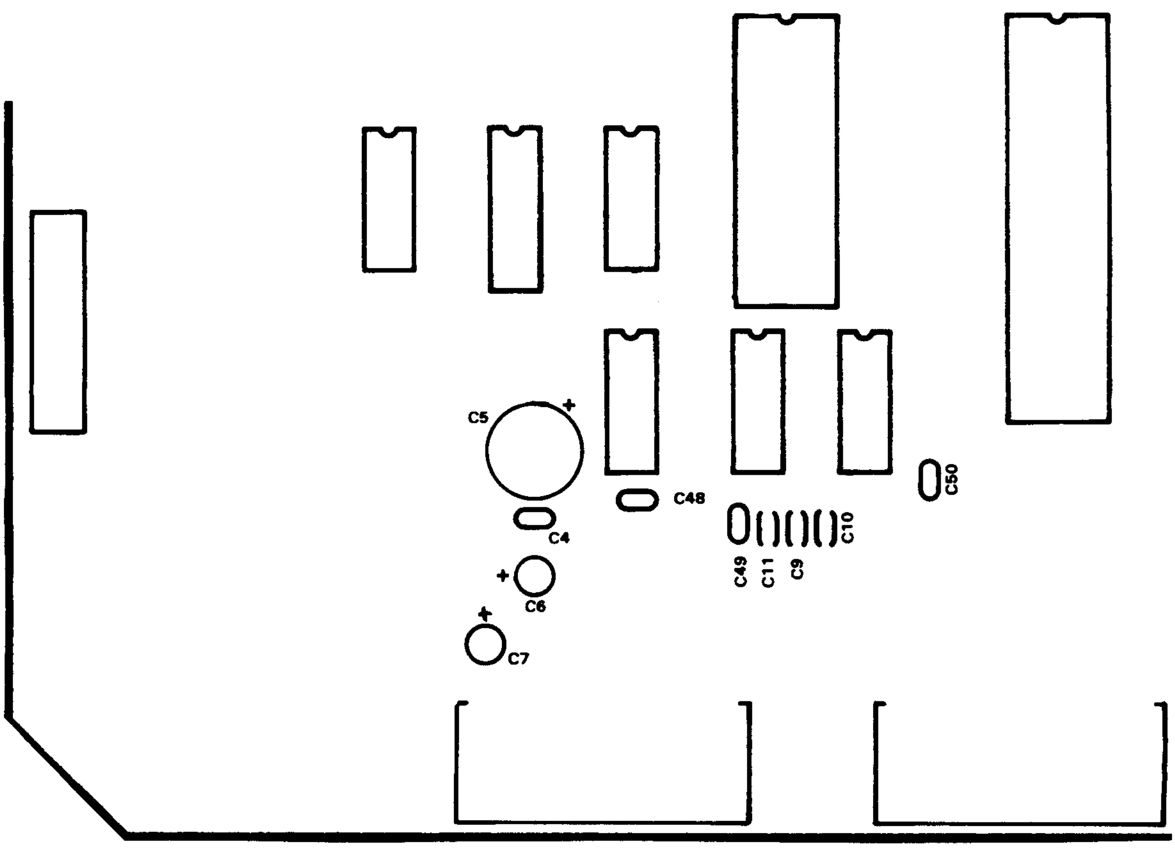
(quadrant 3 - continued)

( ) C5 470 uF electrolytic
( ) C4 0.1 uF
( ) C6 10 uF electrolytic
( ) C7 10 uF electrolytic
( ) C48 0.1 uF

( ) Solder and clip the leads (10 total).

( ) C49 0.1 uF
( ) C11 330 pF
( ) C9 330 pF
( ) C10 330 pF
( ) C50 0.1 uF

( ) Solder and clip the leads (10 total).



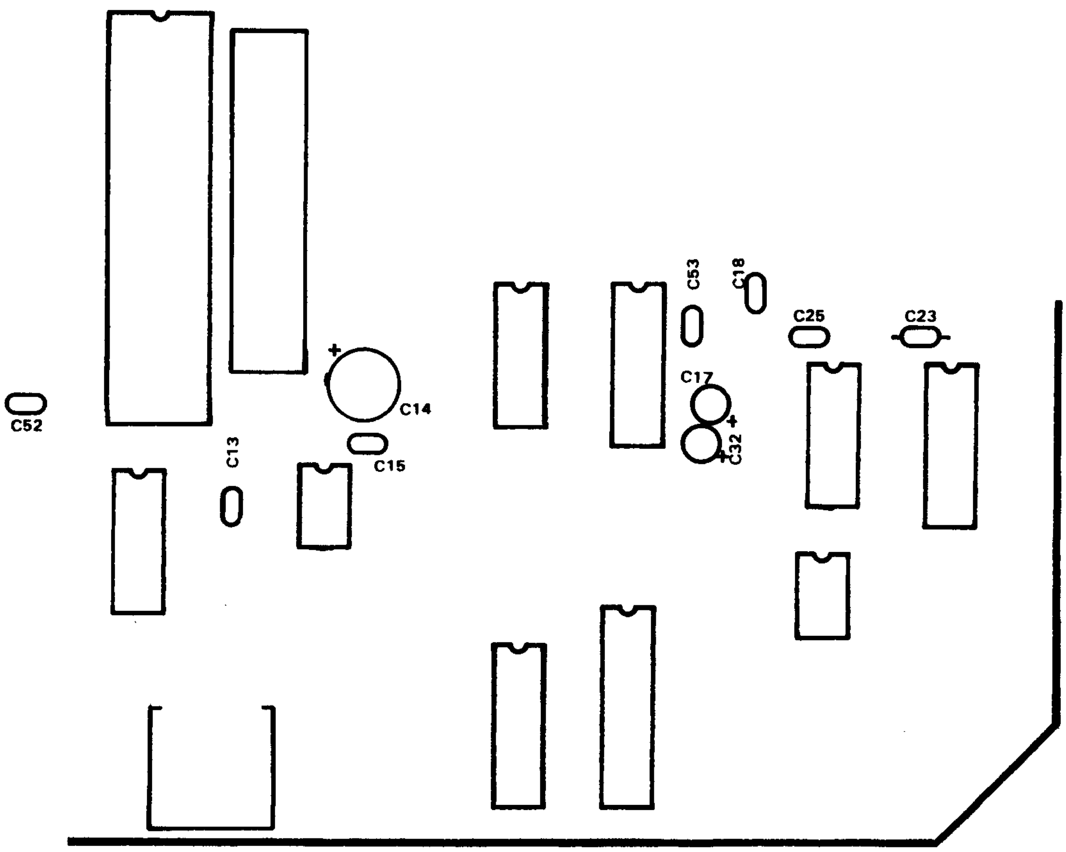
(quadrant 4)

( ) C52 Ø.1 uF
( ) C13 Ø.Ø1 uF
( ) C14 33Ø uF electrolytic
( ) C15 Ø.Ø1 uF

( ) Solder and clip the leads (8 total).

( ) C53 Ø.1 uF
( ) C32 1Ø uF electrolytic
( ) C17 1 uF electrolytic
( ) C18 Ø.Ø5 uF
( ) C25 Ø.1 uF
( ) C23 Ø.15 uF

( ) Solder and clip the leades (12 total).



(quadrant 4 - continued)

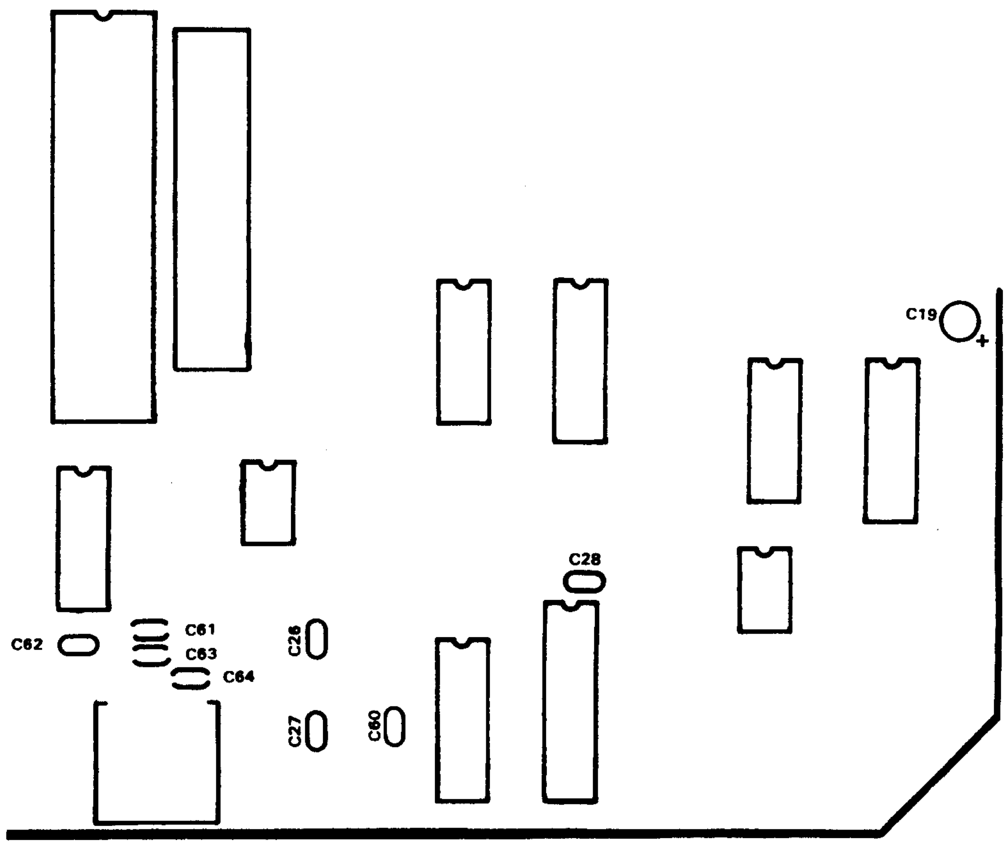
The next capacitor, C19, is the mic audio coupling capacitor. If you will use the TNC with a low impedance microphone input on your radio (such as 600 ohms), use the 1 uF part. If you will be using a radio with a high impedance microphone input, use the 0.1 uF capacitor. If you are not sure, or plan on using both, use the 1 uF device. Holes are provided for both capacitor styles.

( ) C19 1 uF electrolytic
or 0.1 uF

( ) Solder and clip the leads (2 total).

( ) C62 0.1 uF
( ) C61 0.01 uF
( ) C63 0.01 uF
( ) C64 0.01 uF
( ) C26 0.1 uF
( ) C27 0.1 uF
( ) C60 0.1 uF
( ) C28 0.05 uF

( ) Solder and clip the leads (16 total).



(quadrant 4 - continued)

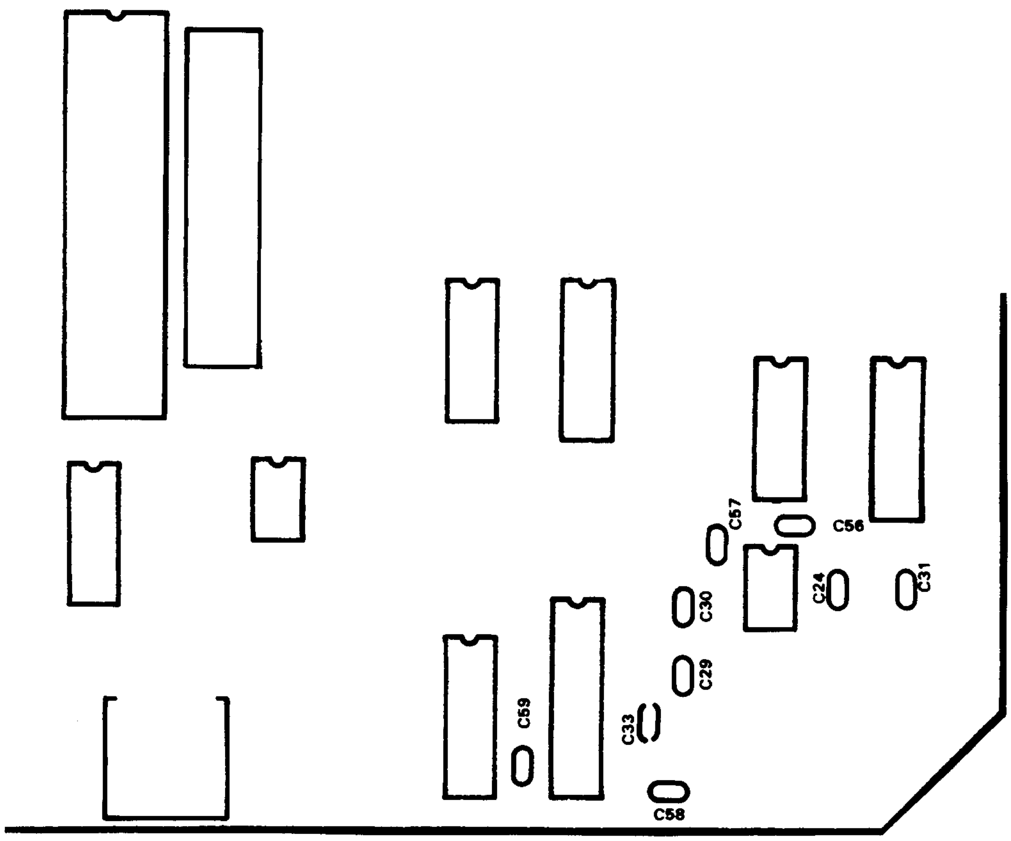
( ) C30 0.0015 uF
( ) C57 0.1 uF
( ) C56 0.1 uF
( ) C24 0.1 uF
( ) C31 0.01 uF
( ) C29 0.0015 uF
( ) C33 0.01 uF
( ) C58 0.1 uF

( ) Solder and clip the leads (16 total).

The selected "thin" capacitor should be installed in the next step.

( ) C59 0.1 uF

( ) Solder and clip the leads (2 total).



You should have 5 capacitors remaining (two Ø.Ø22 uF monolithic, one Ø.ØØ47 uF monolithic, one Ø.ØØ22 uF monolithic and one of either Ø.1 uF or 1 uF left over from step C19).

( ) 5 capacitors remaining.

Now check your work. All leads should be soldered and clipped close to the bottom of the PC board. There should be no solder bridges nor cold solder connections.

( ) OK so far.

Miscellaneous Components

NOTE: In the following steps you will complete soldering parts on the TAPR TNC printed circuit board. Solder each component as you install it in this assembly phase. Be sure to proceed carefully and continue to exercise good workmanship!

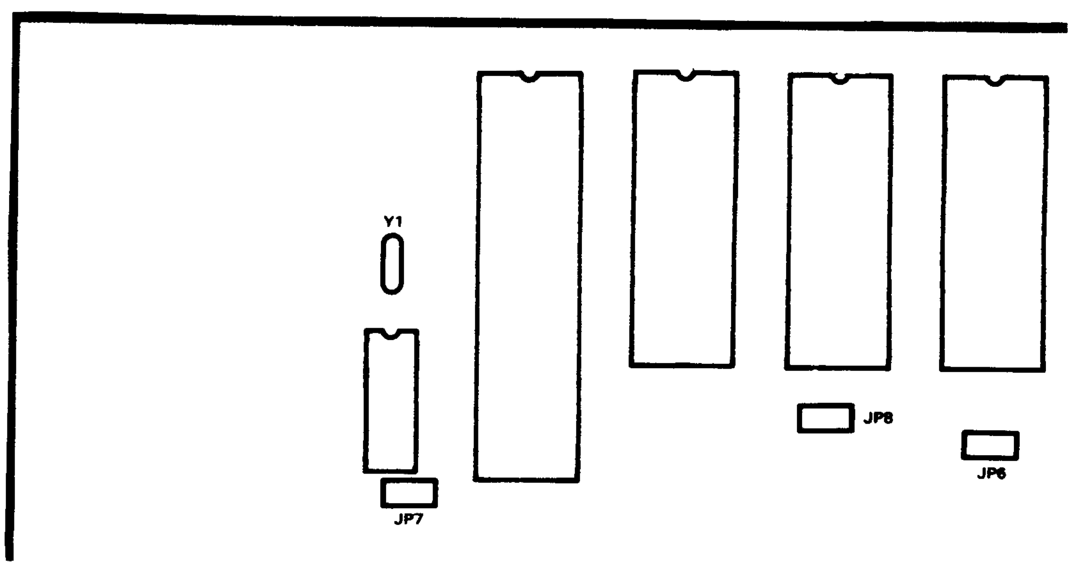
(quadrant 1)

( ) Y1 7.3728 MHz crystal (Do not overheat!)

NOTE: When installing jumper strips, be sure that the shorter pins are soldered to the PC board and the longer pins stick up.

WARNING: Don't grip jumpers with your fingers while soldering. The pins quickly get very hot!

( ) JP7 3-pin jumper strip
( ) JP8 3-pin jumper strip
( ) JP6 3-pin jumper strip

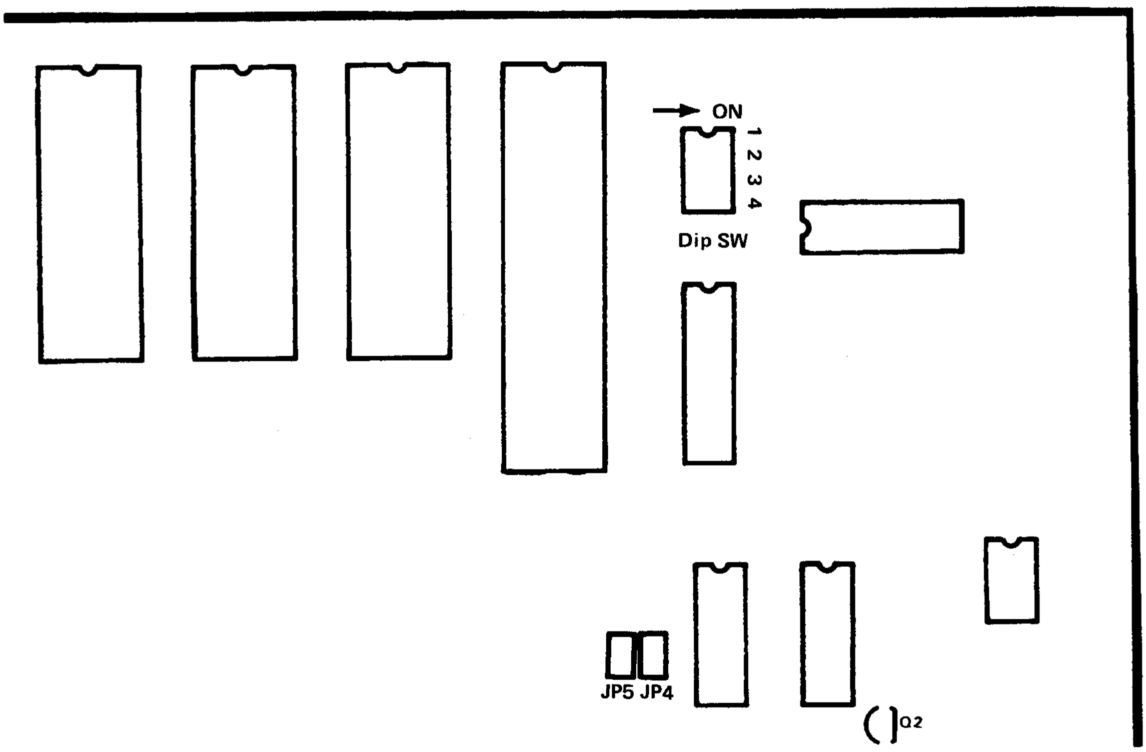


(quadrant 2)

( ) SW DIP switch. Install this part with the "on" positions to the right.
( ) JP5 2-pin jumper strip
( ) JP4 2-pin jumper strip

WARNING: Q2 is static sensitive. Exercise proper handling procedures.

( ) Q2 TX101 VN10KMA. Install this transistor with the flat side towards the right.



(quadrant 3)

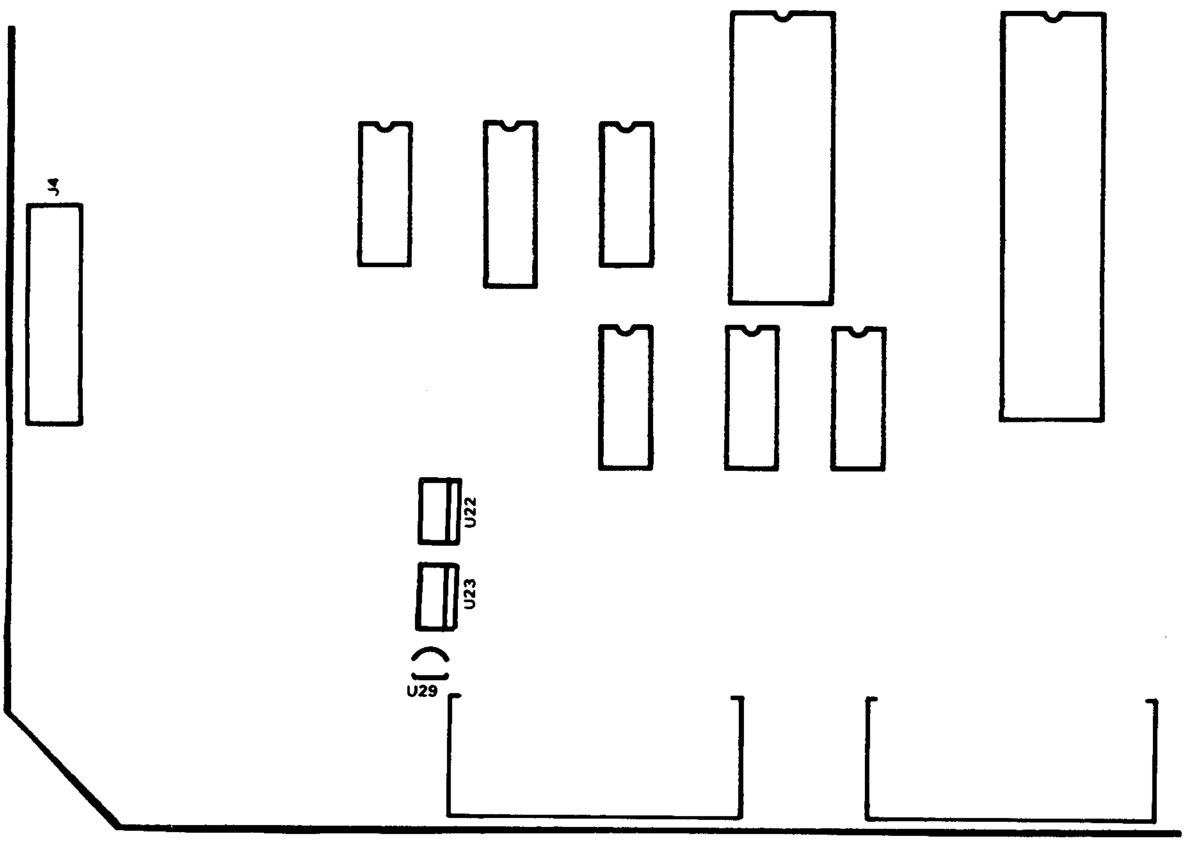
NOTE: The "shoulders" on the leads of the regulators in the next two steps may be of different lengths. The result is that, if you seat the parts against the top board surface, they will stand at different heights. For best appearance, they should stand at the same height.

( ) U22 7812. Mount with the plastic body of the part facing the left edge of the board.

( ) U23 7912. Mount with the plastic body of the part facing the left edge of the board.

( ) U29 79LØ5. Mount with the flat of the body (if the part is "round") or the tapered portion (if the part is rectangular) facing the lower edge of the board.

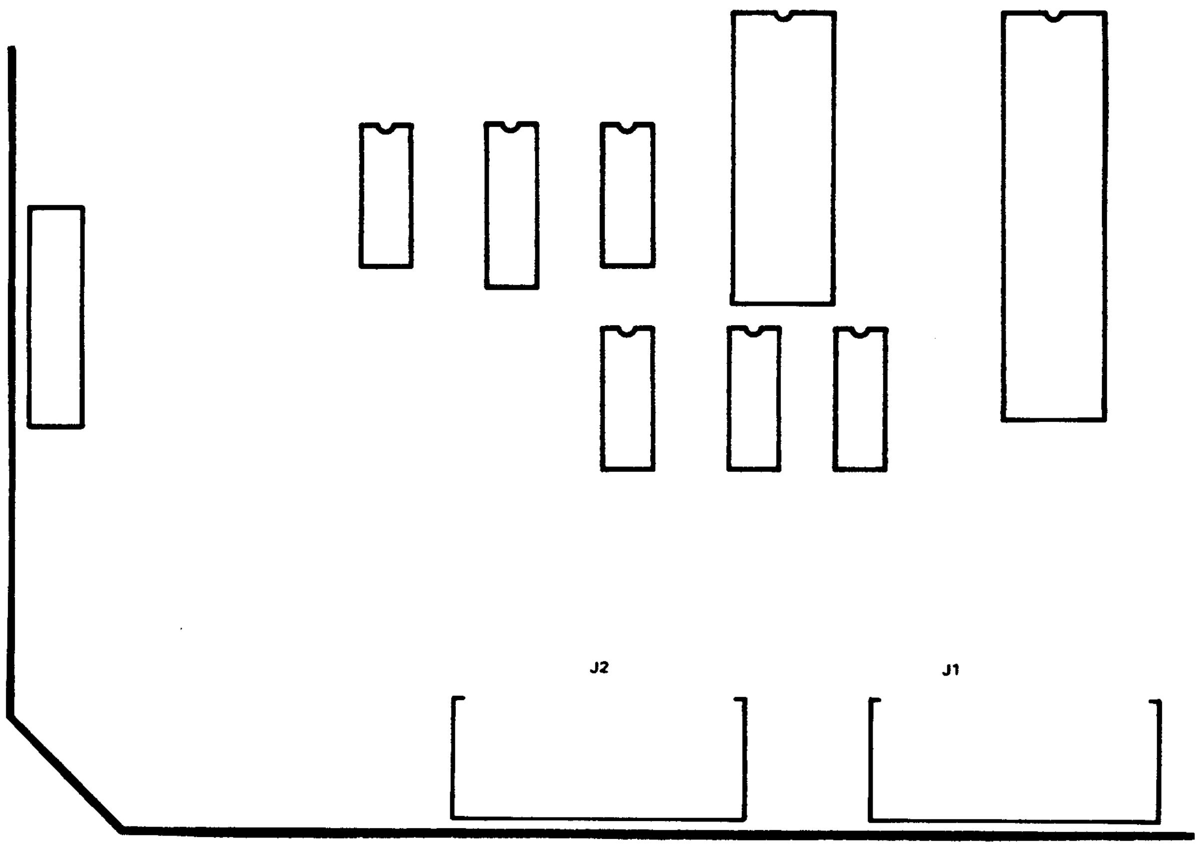
( ) J4 7-pin Power Connector. Install with the "wall" to the right (away from the edge of the board).



(quadrant 3 - continued)

NOTE: The two pc mount right angle connectors will be installed in the next two steps. One is a socket (female) and the other is a plug (male). The "sex" of the connector refers to the contacts, not the plastic shell. Be certain you install them in their correct locations!

( ) J2 25-pin FEMALE DB-25 connector.
( ) J1 25-pin MALE DB-25 connector.



NOTE: If you intend to use the TAPR recommended case for your TNC, skip the following 5 steps! If you elect to mount your TNC in a case, be alert to the dangers of overheating U24 if it is mounted on the board without adequate ventilation or other means of removing the heat.

(quadrant 3 - continued)

( ) U24 LM3Ø9K. (See Fig. F.4.) If you are going to mount the main 5 volt regulator on the PC board, place it on the heat sink and insert the two 6-32 x 5/8 machine screws through the mounting holes in the case of U24 and the corresponding holes in the heat sink.

( ) Be sure the two pins of the IC are centered in the holes in the heat sink. The hole locations are not symmetrical and severe damage to the TNC may result if U24 is improperly oriented in this and the next three steps!

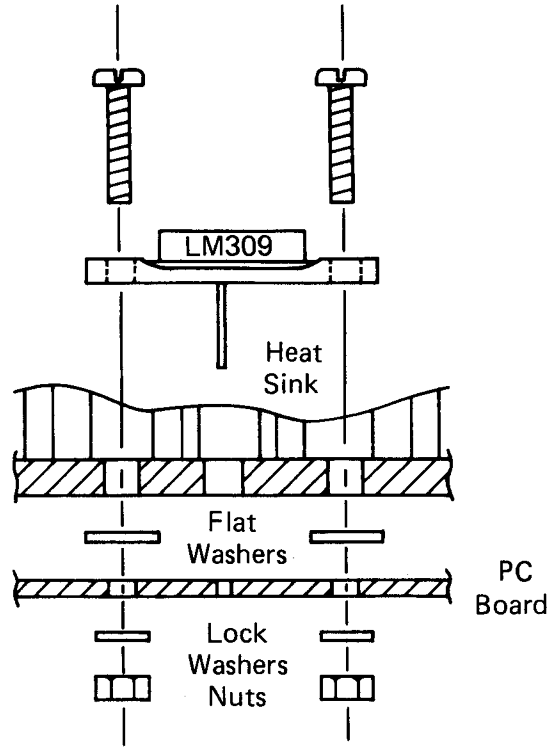


Fig. F.4 U24 Assembly

( ) Now, place the screws through the flat washers.

( ) Seat the heat sink/U24 assembly over the provided holes in the lower left corner of the PC board. The flat washers will be between the heat sink and the board. Attach the assembly to the PC board with 6-32 lock washers and 6-32 nuts. Tighten snugly, as the screws are a power connection in addition to being mechanical fasteners.

( ) Solder the two pins from U24 to the PC board and clip the leads.

(quadrant 4)

( ) J3 9-pin female DE-9 connector. Fasten this connector in place with two 6-32 x 3/8 screws, #6 lockwashers and 6-32 nuts.

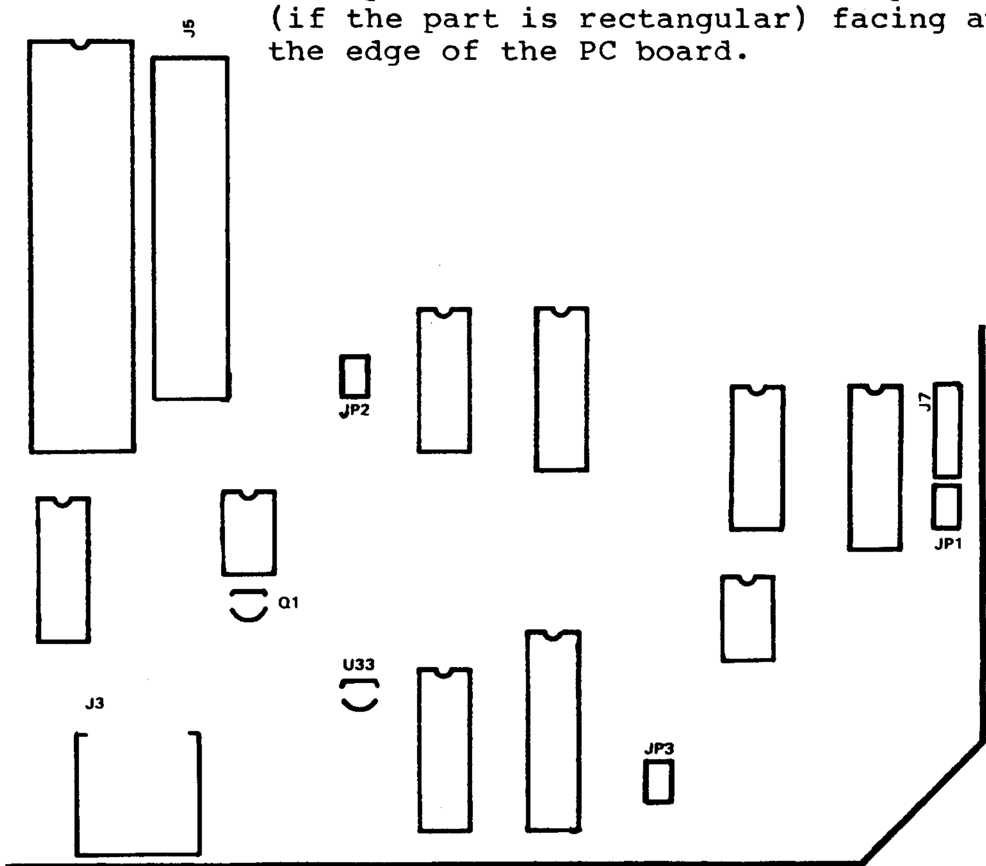
( ) JP2 2-pin jumper strip
( ) JP3 2-pin jumper strip
( ) JP1 2-pin jumper strip
( ) J7 5-pin male strip connector. Orient this connector with the wall nearest U34.

( ) J5 2Ø-pin male header with ejector/latches. Orient this part so the notched wall is nearest U17.

WARNING: Q1 is static sensitive. Exercise proper handling procedures.

( ) Q1 TX101 VN1ØKMA. Install this transistor with the flat side nearest U26.

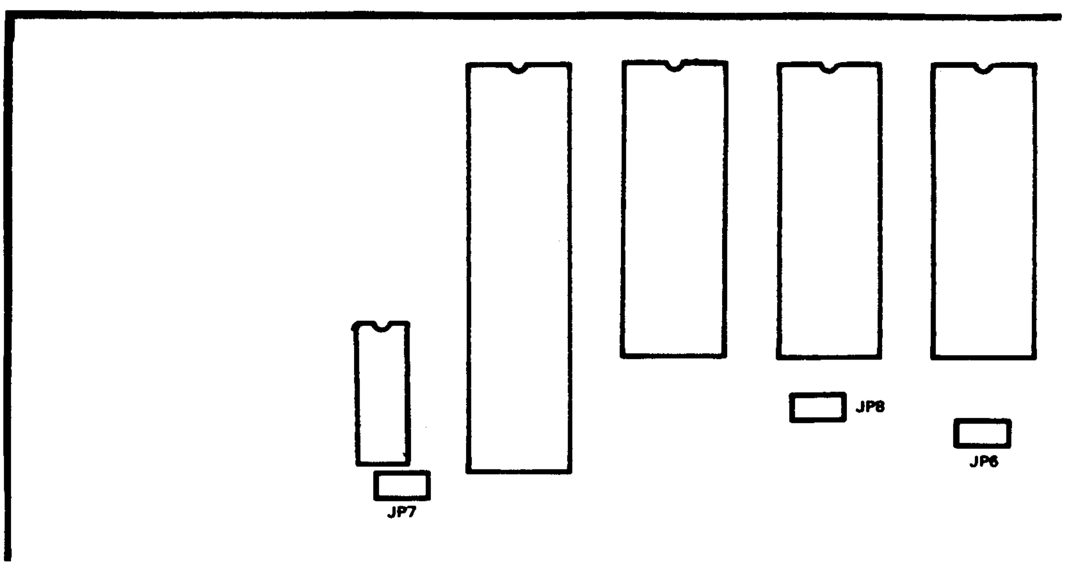
( ) U33 78LØ5. Mount with the flat of the body (if the part is "round") or the tapered portion (if the part is rectangular) facing away from the edge of the PC board.



In the following steps you will install the push-on jumpers in their default positions. They are installed by simply pressing down over the indicated pins. DO NOT solder them in place!

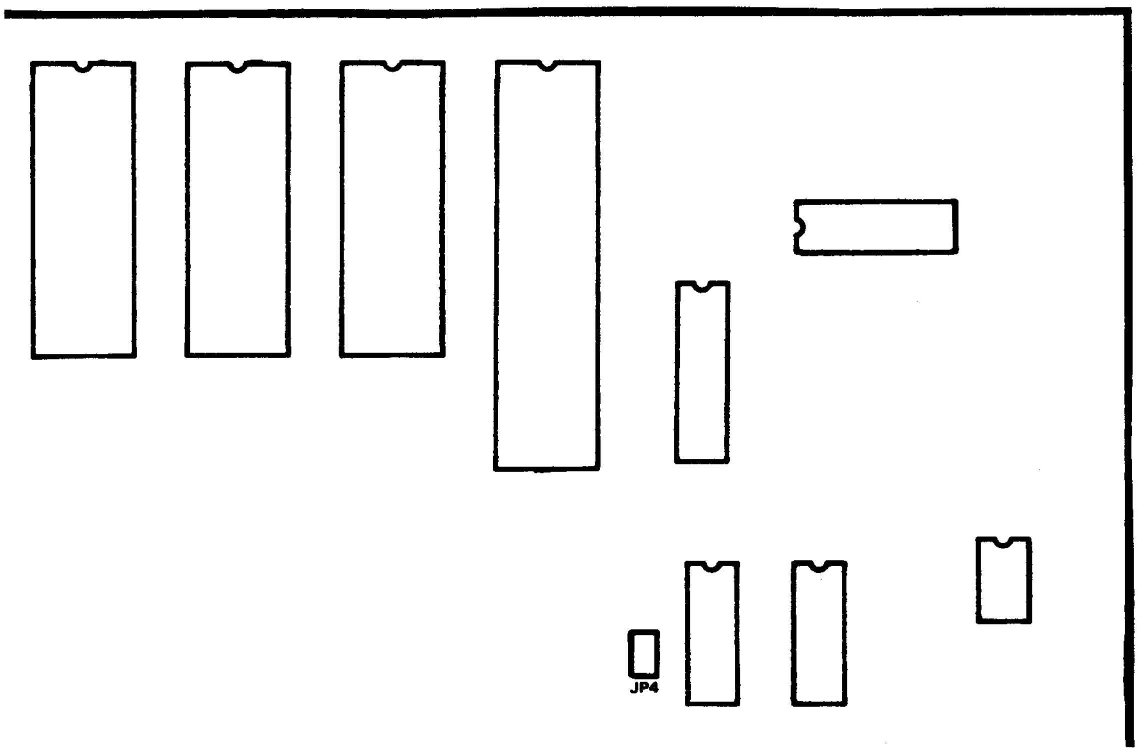
(quadrant 1)

( ) JP7 place over the center and right pins.
( ) JP8 place over the center and left pins.
( ) JP6 place over the center and left pins.



(quadrant 2)

( ) JP4



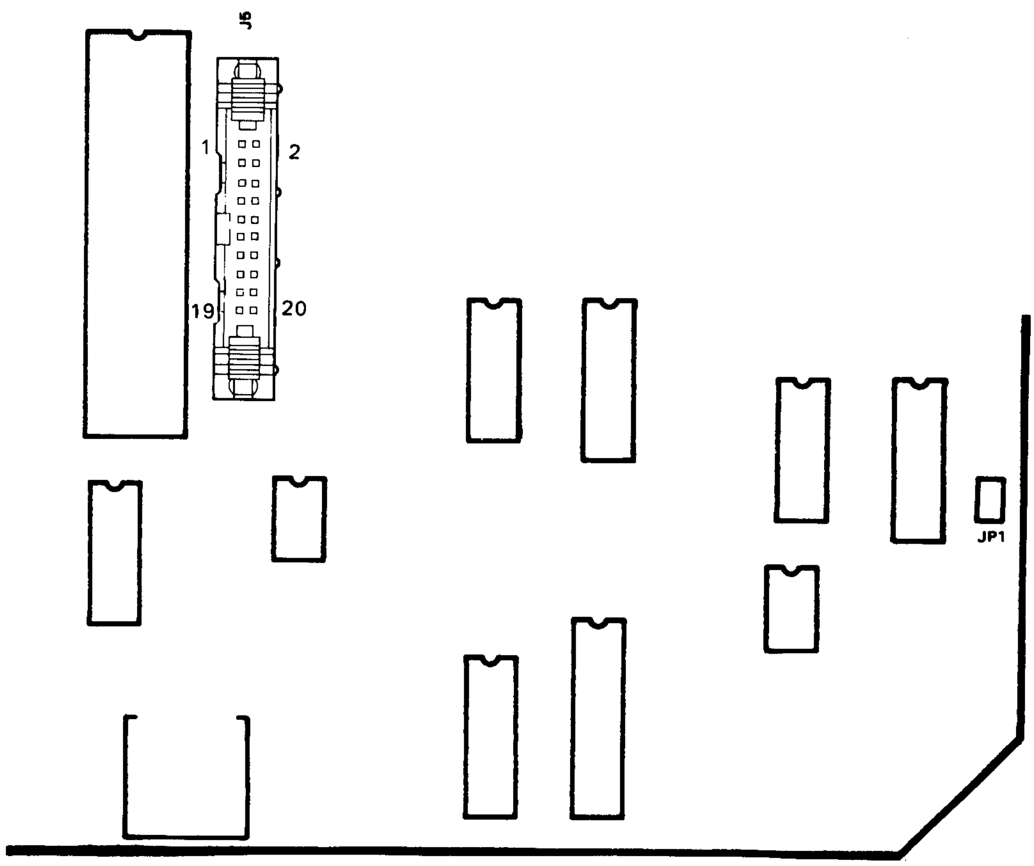
(quadrant 3) -- (none)

(quadrant 4)

( ) JP1

A total of 8 jumpers will be installed in header J5. This is the large 20-pin unit with the ejector/latches. The jumpers will be installed parallel to the long dimension of the board (left to right). The connector pins are numbered with pin 1 in the upper left hand corner, pin 2 in the upper right, pin 3 below pin 1, etc. Thus, the jumpers will short pins 1 and 2, or 9 and 10, etc.

( ) J5 pins 1 and 2
( ) J5 pins 5 and 6
( ) J5 pins 7 and 8
( ) J5 pins 9 and 10
( ) J5 pins 11 and 12
( ) J5 pins 13 and 14
( ) J5 pins 17 and 18
( ) J5 pins 19 and 20



Now lay the PC board aside and prepare to wire the DIP headers. These are the devices that plug into an IC socket but have notched pins above the plug to allow soldering discrete components to the header.

NOTE: DO NOT overheat the headers when soldering or they may deform!

U35

The following devices mount on a 14-pin DIP header, U35. Form the leads on the parts so the component can lay in the slots provided on the top of the header. Solder the two leads, then clip them flush with the sides of the post. Note the bevelled corner marking pin 1 of the header.

WARNING: Capacitors are easily confused due to similarity in markings. Be very sure you install the proper one.

NOTE: The resistor in the next step is 5%.

- ( ) R28 1ØØk (brown-black-yellow-gold) (pins 4 and 11)
- ( ) C16 Ø.Ø22 uF monolithic (223) (pins 5 and 1Ø)
- ( ) R35 16.2k 1% (brown-blue-red-red-brown) (pins 6 and 9)
- ( ) R37 23.7k 1% (red-orange-violet-red-brown) (pins 7 and 8)

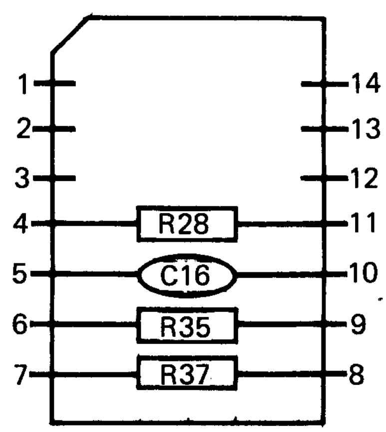


Fig. F.5 U35 Assembly

U34

The following devices mount on a 16-pin DIP header, U34. Note the bevelled corner marking pin 1 of the header.

( ) C22 Ø.Ø22 uF monolithic (223)
(pins 2 and 15)

( ) R47 22.1k 1% (red-red-brown-red-brown)
(pins 3 and 14)

47·5k ( ) R46 ~~47.Øk~~ 1% (yellow-violet-black-red-brown)
(pins 4 and 13)

( ) C2Ø Ø.ØØ47 uF monolithic (472)
(pins 5 and 12)

( ) C21 Ø.ØØ22 uF monolithic (222)
(pins 6 and 11)

( ) R45 1ØØk 1% (brown-black-black-orange-brown)
(pins 7 and 1Ø)

NOTE: The resistor in the next step is 5%.

( ) R44 51Øk (green-brown-yellow-gold)
(pins 8 and 9)

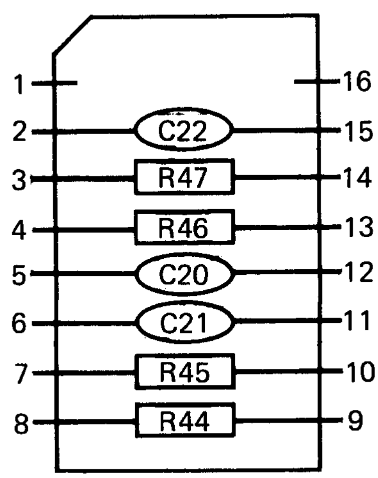


Fig. F.6 U34 Assembly

U30

The following 1% resistors mount on a 16-pin header, U30. Note the bevelled corner marking pin 1 of the header.

- ( ) R58 37.4k 1% (orange-violet-yellow-red-brown) (pins 1 and 16)
- ( ) R57 10.0k 1% (brown-black-black-red-brown) (pins 2 and 15)
- ( ) R56 16.2k 1% (brown-blue-red-red-brown) (pins 3 and 14)
- ( ) R55 10.0k 1% (brown-black-black-red-brown) (pins 4 and 13)
- ( ) R52 63.4k 1% (blue-orange-yellow-red-brown) (pins 5 and 12)
- ( ) R54 10.0k 1% (brown-black-black-red-brown) (pins 6 and 11)
- ( ) R53 57.6k 1% (green-violet-blue-red-brown) (pins 7 and 10)
- ( ) R51 10.0k 1% (brown-black-black-red-brown) (pins 8 and 9)

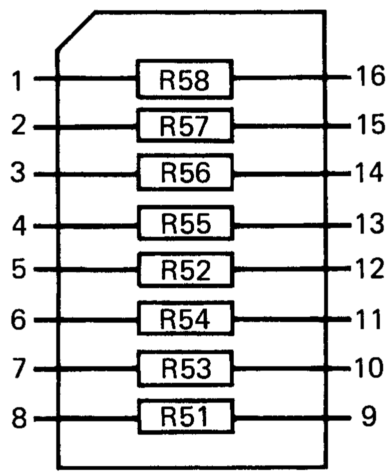


Fig. F.7 U30 Assembly

Now lay aside the three DIP headers you have just completed and prepare to wire the power transformer and connector.

Power Transformer Wiring

The TAPR TNC utilizes a custom wound power transformer for maximum efficiency and minimum bulk. Inspect the transformer carefully and note the three sets of terminal strips on it. On the side with two sets of terminals, the lower set has a black dot near one end. This is the primary terminal strip.

Of the remaining two terminal strips, one has two terminals and the other three terminals. These are the secondary terminals. The two-terminal set (the ac output to drive the main +5 volt regulator) consists of terminals 1 and 2, while the three-terminal set (for the + and - 12 volt, -5 volt and modem +5 volt regulators) are called terminals 3, 4 and 5. These will be connected to the 7-pin power connector in the following steps. (See Fig. F.8.)

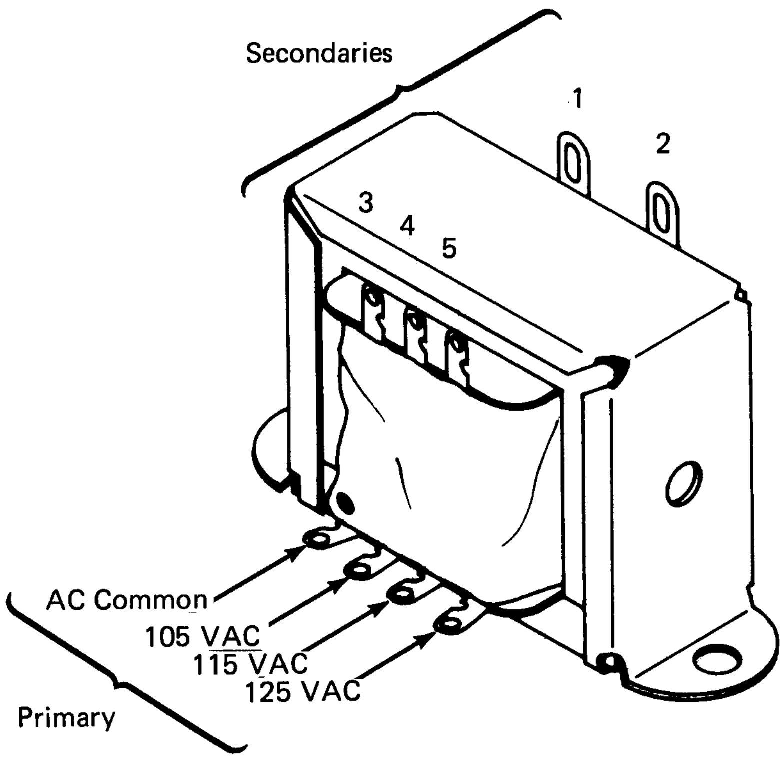


Fig. F.8 Power Transformer Terminals

Transformer Secondary Wiring

( ) Cut five 7" lengths of the supplied hookup wire.

( ) Strip 1/4" of insulation from each end of the five wires.

( ) Attach a MOLEX pin to one end of one of the wires by crimping it with long nose pliers and neatly soldering it. (See Fig. F.9.) Do not allow any solder to run up into the flexing portion of the pin or the locking tab!

( ) Similarly, attach a MOLEX pin to one end of each of the remaining four wires.

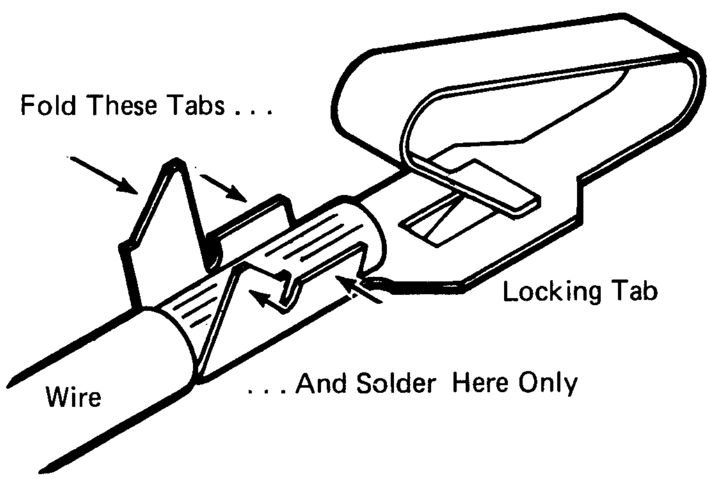


Fig. F.9 MOLEX Terminal Attachment

( ) Insert one of the wired MOLEX pins into the connector housing at pin 6 and push until the tab on the bottom of the pin clicks in place. (See Fig. F.1Ø.)

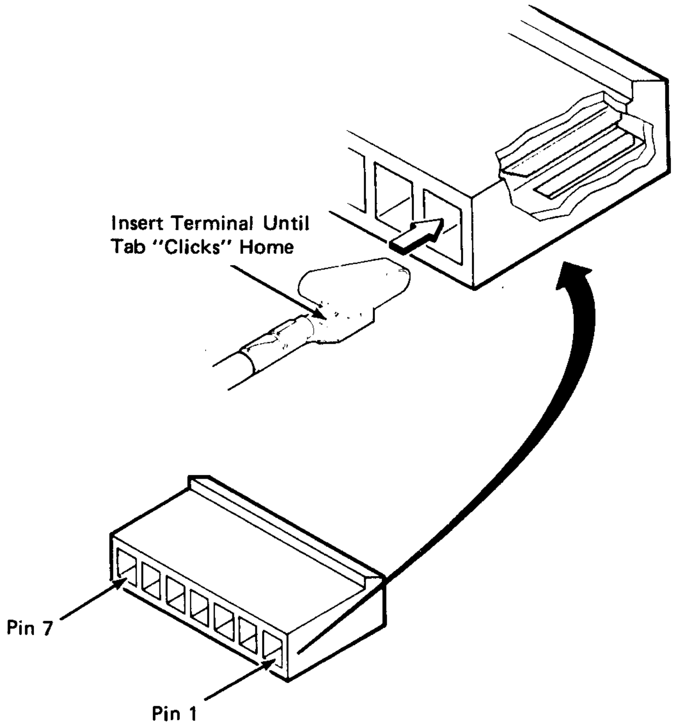


Fig. F.1Ø Power Connector Construction

( ) Solder the free end of this wire to transformer terminal 5.

( ) Insert one of the wired MOLEX pins into the connector housing at pin 5 and push until the tab locks.

( ) Solder the free end of this wire to transformer terminal 3.

( ) Insert one of the wired MOLEX pins into the connector housing at pin 4 and push until the tab locks.

( ) Solder the free end of this wire to transformer terminal 4.

( ) Insert one of the wired MOLEX pins into the connector housing at pin 2 and push until the tab locks.

( ) Solder the free end of this wire to transformer terminal 1.

( ) Insert the remaining wired MOLEX pin into the connector housing at pin 1 and push until the tab locks.

( ) Solder the free end of this wire to transformer terminal 2.

You are now done with wiring the transformer secondary. If you have elected to mount the +5 volt regulator U24 off board, and if you are not using the TAPR recommended cabinet kit, complete the following steps.

If you have purchased the TAPR TNC cabinet kit, proceed to the instructions supplied with it.

NOTE: The +5 volt regulator U24 must be mounted on a suitable heat sink. The following wiring steps assume you have made provision for this. The lead length from U24 to the power connector should not exceed 4 inches or external bypass capacitors of Ø.1 to Ø.5 uF must be added from the regulator's input and output pins to ground with the capacitors mounted at the regulator.

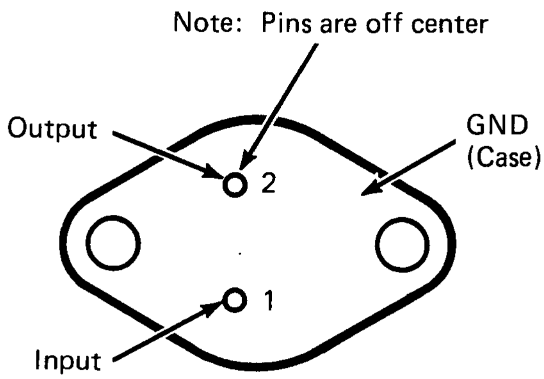


Fig. F.11 U24 Bottom View

( ) Prepare three lengths of hookup wire. Use number 22 AWG or larger. The length should be sufficient to run from your proposed mounting location of U24 to the power connector.

( ) Strip 1/4" of insulation from both ends of each wire.

( ) Attach and solder a MOLEX pin to one end of one of the wires.

( ) Solder the free end of this wire to pin 1 of U24.

( ) Insert the end of this wire with the MOLEX pin into the power connector housing, pin 3.

( ) Similarly, attach a MOLEX pin to one end of another wire.

( ) Solder the free end of this wire to pin 2 of U24.

( ) Insert the end of this wire with the MOLEX pin into the power connector housing, pin 7.

( ) The remaining wire must be attached to the case of U24 (ground).

( ) The free end of this wire will be attached to TNC ground at the mounting screw that passes through the silk screened outline of U24. For purposes of testing, attach it to this point on the TNC board now, using a 6-32 screw, #6 washers and a 6-32 nut.

Transformer Primary Wiring

One side of the ac line will connect to the dotted terminal, and the other, via a Ø.5 amp fuse and switch which you must supply (included in the TAPR TNC cabinet kit) connects to one of the three remaining terminals, depending on your local line voltage. The tap nearest the dotted tap is for 1Ø5 vac nominal, the next is for 115 vac and the last is for 125 vac. Be sure to use the proper tap, as low output will degrade the TNC's performance and high output will overheat it! If you have no way to measure your line voltage, use the middle (115 vac) tap. (See Fig. F.12.)

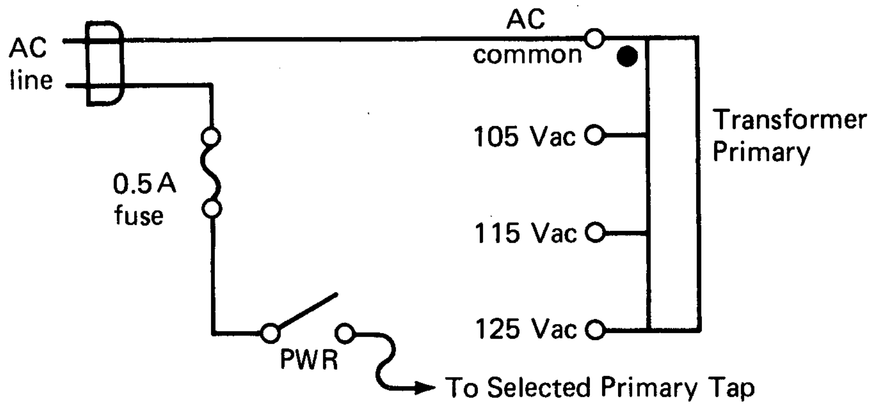


Fig. F.12 Transformer Wiring

If you are using the TAPR TNC cabinet, follow the directions supplied with it for wiring the transformer primary. If you are using your own cabinet, or no cabinet, follow the guidelines given below. Since each installation is different, parts are not supplied with the TNC kit for the primary circuitry.

( ) The primary lug nearest the black dot on the transformer should be attached to one side of the ac line.

( ) The selected tap should go to a power switch.

( ) The other side of the power switch should go to a fuse of not greater than 1/2 amp.

( ) The other side of the fuse should go to the remaining side of the ac line.

At this point, the TNC is wired. You should have a wired transformer, three headers (U3Ø, U34 and U35) and a number of ICs.

Initial Kit Checkout

For initial checkout you will need a voltmeter capable of measuring up to 3Ø volts dc. If you encounter any problems, such equipment as an ohmmeter, oscilloscope, and logic probes will prove very useful.

The first test to perform is the power supply test. With no ICs socketed, but with U24 installed (on or off board as you chose), proceed with the following steps:

( ) Be sure the transformer primary is NOT connected to a power source.

( ) Attach the power plug to J4, on the TNC circuit board.

( ) Ensure that the lip on the power connector is against the wall on J4 and that the plug is not misaligned. See Fig. F.13.

In the following step, power will be applied to your TNC. Be ready to remove power quickly! If you smell anything that resembles a burning TNC, remove power immediately.

WARNING! Dangerous voltages may be present on the TNC and its associated wiring! Take steps to insure your safety. Do not contact any transformer primary voltages or otherwise endanger yourself!

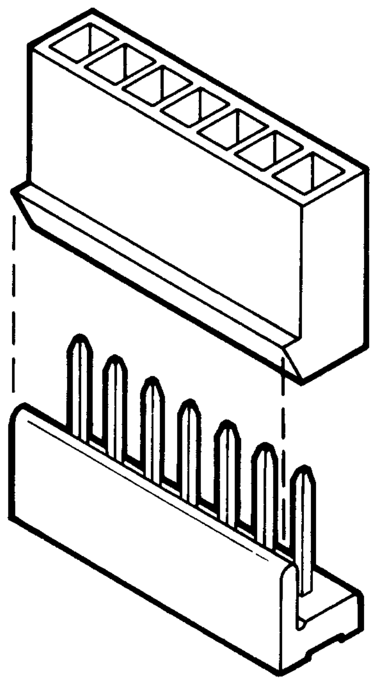


Fig. F.13 Power Connector Mating

If you have problems at any point disconnect power and determine the cause of the problem before proceeding!

( ) Apply power to the transformer.

( ) Measure the +5 volt line to GND. A convenient place to measure this voltage is the bus in the wire wrap area. The 5 volt bus is marked on the silkscreen of the TNC circuit board.

( ) Verify that the voltage is 5 volts +/- Ø.3 volts. (Measured value: __________)

( ) Measure the +12 volt line to ground. Again the wire wrap area is the best place to test this.

( ) Verify that the voltage is 12 volts +/- 1.3 volts. (Measured value: __________)

( ) Measure the -12 volt line to ground. As before, the marked bus in the wire wrap area is a convenient place to test.

( ) Verify that the voltage is -12 volts +/- 1.3 volts. (Measured value: __________)

( ) Measure the analog +5 volts. A convenient place is U28 (located in quadrant 4's lower center), pin 7 or 8. Be very careful not to damage the socket by forcing in probe tips!

( ) Verify that the voltage is 5 volts +/- Ø.3 volts. (Measured value: __________)

( ) Measure the analog -5 volts. (U28 pin 6, 13 or 14.)

( ) Verify that the voltage is -5 volts +/- Ø.3 volts. (Measured value: __________)

( ) Remove power from the TNC transformer and disconnect the connector at J4.

If the above tests were successful, you are ready to install the ICs in your TNC! If the tests were not successful, check the power supply per the instructions in Chapter VI.

IC Installation

The ICs were shipped to you in an anti-static carrier. Many of them are VERY sensitive to static discharge. These parts were purchased from franchised distributors and are of top quality. The probability of them being defective as shipped is extremely low. They should be handled with extreme care.

If you live in a climate with low humidity, or if you have carpeting in your work area, take special precautions. You might try spraying the carpet with some sort of fabric softener/anti-static preparation (such as a 5Ø% solution of "Downy" and water) to reduce static.

If you have a metal working surface, ground it! Place a hand on the working surface before you touch the ICs and be sure to frequently ground yourself to the working surface. If you don't have such a surface, use a large cookie sheet or aluminum foil. Handle ICs only while sitting quietly, never while moving or walking!

Avoid static discharge near the ICs!

( ) Anti-static precautions understood and implemented.

Install the ICs using special IC handling tools, such as those produced by OK Machine Tool. If you lack these, you may gently rock both sides of an IC on your metal work surface to make the leads straight and parallel to each other and install it in its socket with firm, even pressure applied over the length of the IC body.

WARNING: The TNC circuit board will flex excessively (when installing ICs) if it is laid on a flat surface, due to the mounting screws on U24. Do not allow the board to flex when inserting ICs.

If you make a mistake, or have to remove an IC for any reason, special anti-static removal tools are available. If you don't have access to such tools, you may try using a small screwdriver and gently prying the IC out of the socket, a little at a time, from alternate ends of the IC, but to use other than the proper tools exposes you to risking damage to your IC.

NOTE: It is very easy to bend a pin or fold it under the IC during installation. Be sure to check each IC as you install it in the steps below to be sure that all pins are actually in the socket and not tucked under the IC or hanging over the socket! It is easy to make this error!

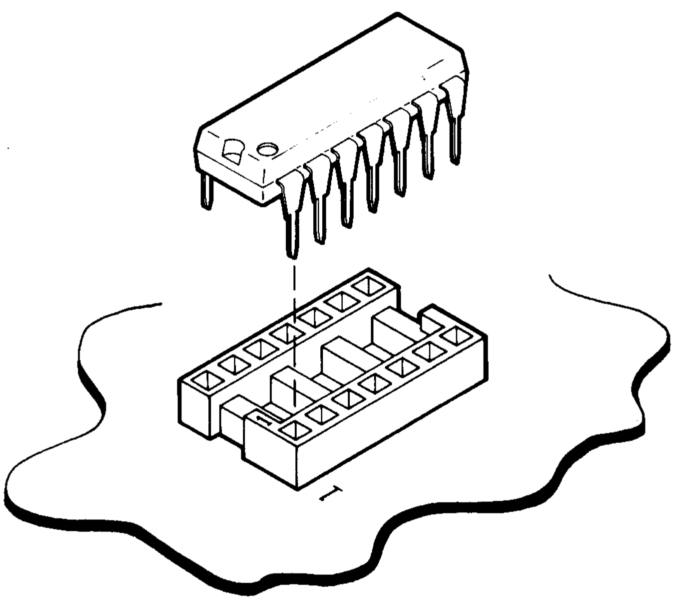


Fig. F.14 IC Pin 1 Identification

Now you may begin.

(quadrant 1)

( ) U1 74LS04
( ) U5 MC6809
( ) U7 6264
( ) U8 (No IC is installed in this position.)
( ) U9 2764 EPROM marked "8000"

(quadrant 2)

( ) U10 2764 EPROM marked "A000"
( ) U11 2764 EPROM marked "C000"
( ) U12 2764 EPROM marked "E000"
( ) U6 6522
( ) U27 2212
( ) J6 (No IC is installed in this position.)
( ) U32 74LS86
( ) U21 7406
( ) U31 1458

(quadrant 3)

( ) U2 74LS393
( ) U4 74S288 (or 82S123 or 7603)
( ) U3 74LS00
( ) U14 6551
( ) U13 6821 (or 6520)
( ) U36 74LS10
( ) U15 1488
( ) U16 1489 (or 1489A)

(quadrant 4)

NOTE: The next step is to install the HDLC controller, U17. This IC is very expensive and the version with a "hump" in the middle is very delicate. Exercise extreme caution when installing this IC. If you ever have to remove it, standard IC handling tools tend to break it into a pair of non functioning 2Ø-pin ICs! Be very careful!

( ) U17 1933 (or 1935)
( ) U35 14-pin header with capacitors and resistors.
( ) U19 22Ø6
( ) U18 2211
( ) U34 16-pin header with capacitors and resistors.
( ) U25 74Ø6
( ) U26 555
( ) U3Ø 16-pin header with resistors only.
( ) U28 MF-1Ø
( ) U2Ø 1458

At this point you should have all the ICs installed. The only parts remaining should be the alternate capacitor for C19 (Ø.1 uF or 1 uF), a 9-pin male DE9P connector and the backshell for that connector, and possibly R59 (56Ø ohm 5%). If you elected to mount U24 off board, you may have some hardware left as well.

( ) Carefully inspect all ICs again to be sure they are all seated in their sockets and there are no bent or otherwise improperly installed pins. <u>Be sure pin 1 of each IC matches pin 1 of its socket or you will destroy the IC</u>.

The next step will be to apply power to the TNC. If it is working properly, you will see a certain sequence of LED activity. Please read the following eight steps before you perform any of them!

( ) Be sure the transformer primary is NOT connected to a power source.

( ) Attach the power plug to J4 if it is not already connected.

( ) Ensure that the lip of the power connector is against the wall on J4 and that the plug is not misaligned.

( ) Be sure the DIP switches are set as follows:

Switch 1 (top)	ON (to the right)
Switch 2	ON
Switch 3	OFF
Switch 4	ON

( ) Apply power to the transformer.

( ) Note that LEDs D1 (RESET) and D6 (SPARE) illuminate.

( ) After about 1 to 2 seconds, LED D2 (CWID) will blink on.

( ) Remove power from the TNC.

If this test is successful, it means that most ICs on the TNC have been exercised, the EPROMs read and the TNC digital circuitry is probably functional. If this test was not successful, please refer to Chapter VI for troubleshooting information.

The next step in bringing your TNC to life is to attach a terminal to the serial port. If your terminal (or computer with a terminal emulation program) has a "standard" RS-232C serial port using a DB-25 type 25 pin connector, your task is probably simple. Construct a pin-for-pin cable with wires in at least positions 1 through 8 and 20, using a male connector for the TNC end and whatever sex connector matches your terminal at the other end.

If, however, you have a terminal that uses a nonstandard connector or uses some of its serial port pins for other purposes, please refer to Chapter V (under the heading "Terminal Interfacing") for details on your TNC's serial port.

Unfortunately, there are many nonstandard implementations of RS-232C serial ports, so a mating cable to interface your TNC and terminal is not included. Furthermore, many computer hobbyists already have a standard cable to interface to a modem, and since the TNC is configured to look like a standard modem at its serial port connector, this cable can be used. One likely source for such a cable is Radio Shack, or any typical computer store.

( ) Terminal interfacing cable constructed.

Now set your terminal to 300 baud. Set the word length to 7 and parity to space, or word length to 8 and parity to none. If the number of stop bits is settable, select 1.

( ) Terminal set to 300 baud with proper word length and parity.

( ) Be sure the TNC transformer primary is NOT connected to a power source.

( ) Attach the power plug to J4 if it is not already connected.

( ) Ensure that the lip of the power connector is against the wall on J4 and that the plug is not misaligned.

( ) Attach the TNC end of the terminal interfacing cable to J2. Do not force it in place -- it should press on snugly and remain in place. Excessive force may damage the connector or PC board.

( ) Attach the other end of the same cable to your terminal.

( ) Apply power to your terminal and allow it to warm up for a minute or two.

( ) Be sure the DIP switches are set as described above.

( ) Apply power to the TNC transformer primary.

( ) The TNC will flash its LEDs as above, then send the following message to your terminal:

Please type an asterisk (*) for autobaud routine

If this message does not appear, toggle the reset switch (switch 3) on then off. If the message still does not appear, consult Chapter VI for troubleshooting hints.

Assuming the autobaud message appears, proceed with the following steps.

( ) As the autobaud message starts to appear on your terminal, strike (gently!) the * key.

If you allow the message to complete before you strike the key, strike it two more times with about 1/2 to 2 seconds between key presses. If the autobaud message starts to appear again, strike the carriage return key WHILE THE MESSAGE IS BEING TYPED ON YOUR SCREEN.

( ) The following message will appear on your terminal:

```
Tucson Amateur Packet Radio Corporation
TAPR/AMSAT AX.25 level 2 version X.Y
RAM length is 2000
cmd:
```

where X.Y represents the software revision level.

The "RAM length is 2000" portion of the sign-on message indicates the number of "bytes" (in hexadecimal) that the software exercised and found operational on your TNC when it went through the power-up reset routine. The value $2000 (where the prefix "$" indicates a hex number) is the same as 8192 (decimal) or "8 k". A different number than "2000" indicates that the RAM IC in socket U7 may be improperly installed or otherwise defective.

If you don't get this message, try again by toggling the reset switch (DIP switch 3) on then off. If it still doesn't work, consult Chapter VI.

Assuming all is well at this point, turn to the "Modem Calibration" section in Chapter V and proceed with the calibration. Please note that these instructions were written for use during the life of the TNC, and not just for the initial calibration, so ignore any messages to remove jumpers that you may not have installed on your TNC. When you are done, return to this section for further instructions.

( ) Modem calibration complete.

Now that your modem is calibrated (with the possible exception of the transmitter drive level and tone null if you weren't able to get access to an oscilloscope or AC voltmeter), it is time to connect your radio to your TNC.

The TNC radio port is designed to work with almost all common (and many not so common) transceivers or transmitter-receivers.

( ) Read the Chapter II "Radio Interfacing" sub-section.

Since there is such a wide variety of radio connectors in use in Amateur radio, the mating connector for your radio, as well as the wire to connect to it, are not included in your TNC kit. However, we have included a mating connector for the radio port on the TNC, along with a backshell. Please get these parts out and warm up your soldering iron for the following steps.

In constructing the radio interface cable, you will need to obtain a mating connector(s) for your radio's microphone and headphone or external speaker jack(s). We recommend you keep the length of the radio interface cable to under about four feet (1.3 m). If possible, use shielded wire for the microphone audio connection. The instructions below assume you are using such wire.

NOTE: Hookup information is contained in Appendix D for some common Amateur radio equipment. Please check this Appendix for information regarding your radio. If your radio is not documented in Appendix D, please take a few moments of your time to fill out and submit to TAPR the "RADIO INTERFACING DATA" page found at the end of Chapter II once you have successfully interfaced your radio and TNC. This is so we may include it in future editions of this manual.

( ) Locate the DE9P connector.

( ) Prepare a length of shielded wire. This means to separate the braided shield from the insulated center conductor, then strip about 1/4" of insulation from the center conductor.

( ) Solder the center conductor to the audio output terminal of your radio headphone/speaker connector.

( ) Solder the shield to the audio common (or negative power common, NOT audio common if such point is at a positive potential with respect to your radio's power "ground").

( ) At the other end of the same shielded wire, solder the center conductor to pin 3 of the

DE9P.

( ) Solder the shield to pin 7 of the DE9P, being careful to insulate or otherwise prevent the shield from contacting any other pins of the DE9P.

( ) Prepare another length of shielded wire.

( ) Solder the center conductor to the microphone audio input contact of your radio microphone connector.

( ) Solder the shield to the microphone audio common (this common MUST be at the same dc potential as the headphone/speaker audio common as well as dc power common.

( ) Solder the center conductor of the other end of the wire to pin 5 of the DE9P.

( ) Solder the shield to pin 9 of the DE9P, again taking care to ensure that the shield doesn't contact any other pins.

( ) Prepare a length of unshielded wire the same length as the microphone audio cable just installed.

( ) Strip 1/4" of insulation from each end of this wire.

( ) Solder one end of this wire to the PTT (microphone key) contact of your radio microphone connector.

( ) Solder the other end of this wire to pin 4 of the DE9P.

( ) Verify that there are no shorts or opens in the cable you have just fabricated.

( ) Assemble the backshell(s) (if any) onto your microphone and headphone/speaker connectors.

( ) Remove the cover from the rear or side cable entrance hole of the backshell provided for your DE9P connector (depending on which one you desire to route your cable through). See Fig. F.15.

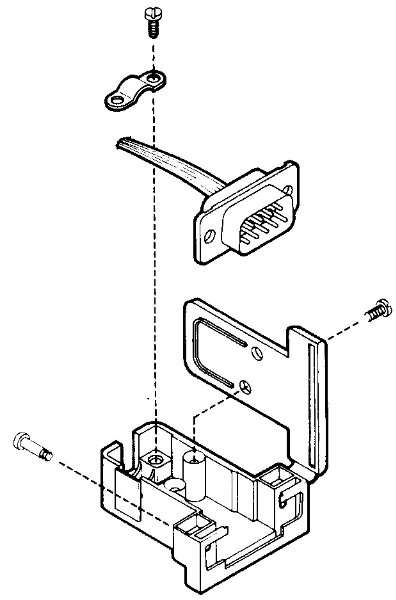


Fig. F.15. Radio Interface Connector

( ) Using the supplied screws, attach the cable clamp to the backshell, allowing a slight amount of slack in the cable to prevent strain on the soldered connections on the DE9P.

( ) Fold the backshell over the DE9P and assemble it. Fasten it with the supplied screws.

( ) Finally, thread the two captive screws through the flange on the DE9P. These screws will be used to attach the connector to the radio port connector (J3) on your TNC circuit board.

( ) Now attach the radio port connector to J3 on your TNC and tighten the two captive retaining screws snugly.

( ) With your radio switched off, attach the microphone and headphone connectors from your TNC radio port cable to your radio.

( ) Verify that your TNC's DIP switches are set as described above.

Congratulations! You are now ready to operate your new Amateur packet radio station!

Please refer to Chapter III (Operation) where you will be guided through the initialization procedures for your new TNC and place your packet station on the air.

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. 14-25.

Margaret Morrison, KV7D, Dan Morrison, KV7B, and Lyle Johnson, WA7GXD, "Amateur Packet Radio: part 2," Ham Radio, August 1983, pp. 18-28.

National Semiconductor, "Technical Product Information, MF1Ø", 29ØØ Semiconductor Drive, Santa Clara, CA 95Ø51.

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.