When I arrived in Nottingham in September 1985, having been active on packet radio in Cape Town previously for about a year, I discovered that packet radio in th U.K. was in a state of infantile nor existence! After sitting around and twiddling my thumbs for about 6 months talking only to mysel I decided the time had come to try and promote packet radio as a mode of digital communications. All attempts at trying to get people interested in the mode locally were frustrated by one basic fact; TNC's were very expensive! As a result of this, I decided to produce a printed circuit board for a very simple, reliable and cheap controller design similar to the one we in Cape Town had used and based on the commercial design from GLB electronics in Buffalo, New York from whom I obtained permission to distribute the firmware on a royalty basis. This article is a basic description of the unit and documents its foibles, advantages and disadvantages.

The TNC can be described as an AX25 level 2 terminal node controller with software HDLC compatible with all existing version-2 tnc's. Inputs to the unit are RS232 Data in, RS232 RTS in, Audio in form radio and Squelch line input from rig (optional). Outputs are: RS232 Data out, RS232 CTS out, Audio Out to radio and PTT to radio. It is also possible to run the unit using a simple Teletype as a terminal.

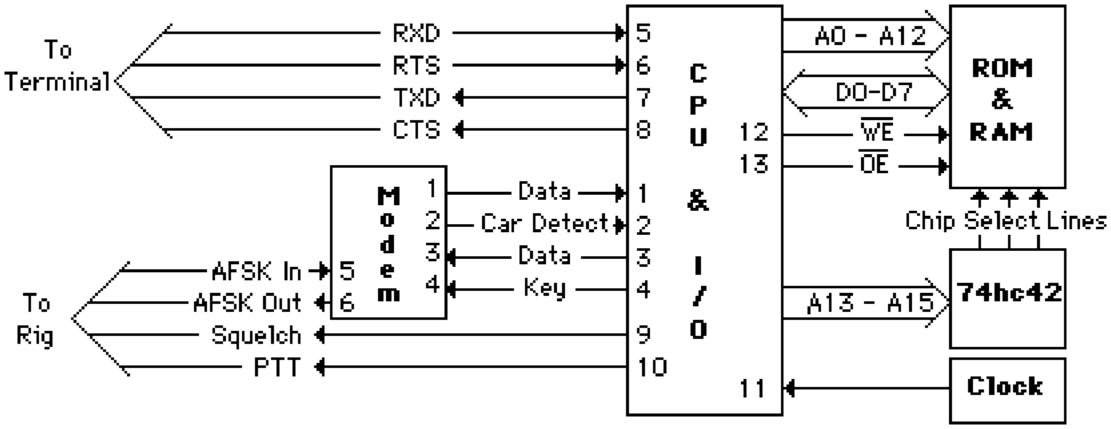
The main advantage of this design is its cost. It is possible, with judicious buying of components, tc produce a working TNC for about £50, i.e. 1/3 the cost of a TNC-2 clone.

The main disadvantages also stem from its very cheap design: Firstly, since the HDLC protocol is implemented totally in software, it is not possible for the unit to serve both the host and the radio link at the same time. This means 1. that full duplex operation is not possible since coding and decoding operations are handled by the same processor, 2. packets may be missed or aborted when input from the host computer is being served since this takes precedence over the link and 3. due t the absence of -12V line the "RS232" does not comply with the full voltage specifications of the standard. In practice, however, these limitations are not at all serious with the addition of a suitable driver program (qv) and simple hardware will prevent packet loss and all "RS232" ports I have come across have accepted the 0-12V signals provided without any problems.

Other disadvantages stem from the amount of firmware use to drive the TNC (8k) This produces th other "disadvantages": 1. the unit only supports single connect only operation, unlike the TNC-2 which supports 10 connect streams. 2. The TNC does not issue a <DM> (disconnect mode) frame to tell a caller it is busy when it receives a connect request from another station - this feature is supported in TNC-2 clones but is not a part of the AX25 protocol.

Two of the big plusses of this tnc, in my view are 1. the excellent block mode facility it provides, making it possible to have a very good idea of what is going on on a channel with a minimal amount of driver software and 2. the ability to configure it to be a remotely controllable stand-alone packet repeater. I have also found the sensitivity of this TNC on weak signals on VHF FM better than the TNC-2 I have despite the absence of input filtering...

Block Diagram of TNC



Circuit description:

MODEM: The Modem circuitry is standard and is similar to that used on the TAPR boards. The presets attached to pins 7 and 8 of the XR2206 control the transmit high and lo frequencies. For VHF, "high" is 2200 Hz and lo is 1200 Hz. For HF it is common practise to use 300 Baud 200 Hz Shift. (If you want to use this TNC on HF, I would recommend that you replace either use a good 30 baud narrow shift receive MODEM and the existing transmit circuitry, or replace the MODEM entirely. A modem is described later in this article...) The other presets are used to set the output level of the FSK generator (useful if you go digging into the rig and bypass the microphone gain control) and the XR2211 PLL center frequency. If the TNC audio output is used to drive the Microphone input of the rig, a 1MΩ resister should be used in series mounted as close to the microphone input as possible.

The only input to the XR2211 is the AFSK from the rig. This signal can be anything from 5 mV to 1 P-P. In an FM rig, it should preferably be taken from the discriminator output before the rest of th audio circuitry distorts the signal. If this is not possible it should be taken as soon as possible afte that point using a 0.001 μF capacitor in series to undo the de-emphasis of the receiver. I have found it to work well with this capacitor if taken from the earphone socket although I subsequently dug into the rig and brought a lead out from just before the volume control via a 0.0 μF capacitor which has worked a little better. If a squelch backoff line is used (recommended on VHF/UHF FM because there are sometimes other non-packet users on the channel when it is desired to use packet), this should be a signal from the rig that rises to above 3 volts when the squelch opens. This input to the TNC is of very high impedance and will not interfere at all with the rig squelch operation. In my rig, the squelch worked "upside down" dropping to 0 volts when it opened, so I put a single transistor invertor in series to provide the proper signal.

CPU & Drivers:The next section is the CPU and drivers section which consists of a minimum configuration Z80 microprocessor. The 4503 is a tri-state input buffer used to drive the data line to the CPU while the 4013 chips are flip-flops to latch data lines to serve as outputs from the system. Inputs are: RS232 In from the host computer, RTS in from the Host computer, Squelch In from the Tx/Rx if it is desired to operate in Squelch backoff mode and, from the XR2211, decoded AFSK Data In and Carrier Detect. The two 4013 "D" flip-flops generate the following outputs: RS232 Data Out to the host computer, CTS to the Host computer, PTT to the Tx/Rx and, to the XR2206, a keying line and a Data line. The 74HC32 provides the various control lines to the flip-flops and the tri-state buffer as well as the Write Enable and Output Enable lines to the Memory. Be careful of connecting any of the data or address lines of the Z80 incorrectly. These lines are NOT buffered and any insult will cause the chip to blow. The RESET line should be taken to a push to make reset button to allow resetting of the system should it not reset itself properly on power-up. The system clock is simply a crystal controlled oscillator constructed using the invertors in a 74HC04 chip. The memory multiplexer provides decoding for the full 64k address range of the Z80. The 0000-1FFF Chip select line (line 0) should go to the CS pin of a 2764 or 27128 ROM. For RAM, I have used 2 6264 chips occupying address space from 4000-7FFF (lines 2 and 3). Each memory chip has all the address and data lines connected as well as the WE and OE lines from the 74HC32. The 5V line comes from an on board 7805 regulator mounted on a heat sink. The circuit is capable c driving 56k of RAM although only the upper 48k are supported by the Software. I have found one 6264 (8k) adequate for most applications although I have used two (16k) on this board. The more memory one has, the larger the message buffer is. In the future when automatic operations are allowed by the DTI (I live in hope!) as they are in most other countries, a large amount of memory will provide a large message leaving buffer for amateurs connecting to your station while you are not in attendance.

Basic firmware description:

A device is only as good as the program that drives it, and this TNC is no exception. The firmware on this unit performs the following tasks:

1. It supports either an ASCII of Baudot terminal at all standard baud rates up to 9600 baud, automatically selected.
2. A simple "chat mode" program is built in. Otherwise the unit may be used in either "transparent" or "block" mode.
3. It supports the full AX25 Level 2 protocol at baud rates of 300, 600 or 1200.
4. It supports, with appropriate hardware configuration, remote control over a packet link.

Construction:

The unit is constructed on a 190 mm by 95 mm double sided through plated printed circuit board. Components are inserted as per layout diagram starting with the IC sockets (recommended), resistors, capacitors and finally solid state components. No components are very critical apart from the 22nF tuning capacitors on the XR chips. where I would recommend that you use high stability close tolerance polycarbonate items. The unit should be housed in a metal box with appropriate RF bypassing of entering leads for best results. The 7805 regulator may require a clip on heat sync.

Extensions:

1. You may find, if you use chat mode a lot, that the system interrupts transmitting packets wh you type. To prevent this, connect a diode from the TNC CTS line to the PTT line (+ve to PTT) and make sure the CTS/RTS configuration works on your terminal.
2. If you wish to stop your terminal interrupting received packets, connect a diode from the TNC CTS line to pin 5 of the XR2211 chip (+ve to Pin 5) This connection is best used with a terminal or program that allows type-ahead and a high baud rate otherwise you may find the TNC misses keyed characters.
3. If you wish to have a "Carrier Detect" display, simply connect a 1k resistor to pin 5 of the 2211 to an LED to +12V. A similar arrangement can be used on the PTT line if desired.
4. Expansion Board (Filter, PTT timer, Full memory) and HF modem are in planning stage.

Hints:

1. Make sure that resistor R10 connects to +5V and not to 0V. Two holes are provided and may cause confusion.
2. If you find that the RTS line does not work with your particular terminal, try replacing R12 with a 100k resistor (or removing it!). Do NOT tie RTS to ground.
3. If you find the +5V line a bit noisy, replace C1 and C2 with 10µF tantalum capacitors.
4. Do use high quality, high stability values for C7 and C16. (22nF)
5. Do use a 0.001µF capacitor in series with the audio lead from the rig unless the audio is taken directly from the discriminator output.
6. Most component values are nominal and the nearest standard values can be used with impunity.
7. If used on HF, the unit needs a different receive modem. Transmit works fine!
8. Do not tie the squelch line low. If not used as described in the doc, leave it floating high.
9. If the clock insists on starting up on an overtone, insert a 1k resistor between pin 1 of the 74HC04 and the crystal.

Connections to the TNC PC Board:

1. Top Left: +12V and Ground. Be sure to get these the correct way around. The +12V line is the one that connects to pin 1 of IC1. *** Be sure to check this!! No protection against incorrect connection is provided ***
2. Middle left: above VR1 and below VR2. These connect to the DPDT switch to select the TX frequency shift. The common pole connects to the terminal that connects to pin 7 or pin 8 of IC Connect the other two terminals consistently. (optional)
3. Center to left of IC3. Here there are two 5 pin connections. The left hand one goes to the Radio transceiver and consists of, from top to bottom:

Left:	Right:
1. Ground.	1. Ground.
2. Audio In from Rig.	2. Data In from Computer.
3. Audio Out to rig.	3. Data Out to Computer.
4. Squelch In from Rig.	4. RTS In from Computer.
5. PTT line Out to Rig.	5. CTS Out to Computer.

4. Center above C21. Reset line to push to make pushbutton. Grounding this line momentarily will reset the TNC.
5. Far right. Expansion socket. For future expansion if desired. No connection necessary.

Testing:

1. With a good magnifying glass, go over the board and look for possible dry joints and bridged tracks.
2. With no chips plugged in, apply power to the board and check that minimal current is drawn.
3. Plug in the 74HC04 chip, switch on and check that the clock is running, if possible on an oscilloscope.
4. Plug in the Z80, 4503, ROM, RAM, 74HC32, 74HC42 and both 4013 chips. Switch on and check that not more than 150mA is being drawn.
5. Connect your terminal to the RS232 (CTS and RTS connections not critical at this stage), switc on, press RESET and press the <Carriage Return> key. A sign on message should appear on your terminal.
6. Now insert the XR2206 chip and set up the transmit tones using the DH and DL commands. These should be 1200 and 2200 Hz. It is immaterial which is which if you get them the wrong way around since AX25 uses a non return to zero protocol rather than <mark> <space> as in RTTY.
7. Now the XR2211 can be set up. This can be done as follows: bend out pin 2 and connect it to pin 10. Then adjust the tuning resistor to give an output frequency of 1600Hz on Pin 3. Alternatively, connect the audio output to the input and a 'scope to pin 7 of the XR2211, issue the following commands: SF-0 SJ-3, and issue packets using the AI command. Adjust the tuning resistor to give a square wave on the preamble bytes.

The unit should now be operational and, if audio is connected, any packets on the channel will be displayed as received. (Use OH-E command to see the packet headers)

How to use the TNC

1. Initial Sign On:

Since the TNC serial I/O automatically supports different codes and baud rates, it is necessary to provide some method for it to identify these. The way this is achieved is to send the TNC a Carriag Return (ASCII 13 or (cr)) character as the very first character after it has been switched on or RESET. If nothing happens when this is done you should RESET the TNC by pulling the RESET line from the TNC down to ground (ie. pushing the push to make button attached to the RESET pin of the Z80. Once the Character has been decoded and the appropriate flags initialized, the TNC will respond with a sign on message in the form:

PACKET V3.92:

or some similar message depending on the exact version of the software in the ROM. This should be version 3.7 or later. Once this sign on has occurred, the TNC is in Command Mode and ready to respond to commands (described in the detailed command description documentation) and, by default, will also display any packet traffic being transmitted on the channel it is on.

2. Entering Callsigns:

To make contact with another station, you first have to enter the link callsigns that will form the address field of the packet. If your callsign has been programmed into the ROM then you need not enter it. If not, you must enter it into the TNC using the SC command.

This command will respond with

XXXXXX-

where XXXXXX are the characters composing the current callsign in memory. If you now enter a space or carriage return, these will be left the same. Otherwise, the new callsign can be entered. If you wish to delete a callsign completely, enter a "-" as the first character. Callsigns are up to 6 characters in length. If yours is shorter, then terminate the entry of the callsign with a space. Following the entry of the callsign, it is necessary to enter the SSID (Secondary Station Identifier) This is a number from 0 to 15 and it allows a single callsign to operate more than one TNC on the same channel without mutual interference. Normally just enter a 0 or leave it alone. It is customary to use a 0 for your first or main station. Normally most stations will all use 0. If you make a mistake in entering the callsign, simply terminate the entry by entering space characters o - characters until command mode is reached (denoted by the appearance of a : character) and re-enter the call.

Once your call is correct, you need to enter the destination callsign using the SD command in the same way.

If another station is used as a repeater in the chain, that callsign of the repeater station must be entered using the SV command. Note that up to 8 repeater stations can be used in any given link. Here, entry of callsigns and SSID's are made one after the other until terminated by entering a "-" for a callsign which will terminate the entry procedure. If you make a mistake in the entry procedure, simply repeat entry using the SV command. Each callsign in the chain will be prompted and you can either accept it as unchanged by entering a space, change it by entering the call or SSID or delete it completely and so shorten the chain by entering a "-".

3. Checking TNC Status:

This can be done using the S(cr) command. This responds with text similar to:

```
1 /G0BSZ 0/------ 0/ //LDU/XQE/RL/0
```

Full details of each field in the line are given in the detailed documentation. In this example, for instance, the caller TNC has the callsign G0BSZ with an SSID of 0, no destination callsign has yet been entered as have no repeaters. The 1 at the start of the display indicates that the TNC is "disconnected". This number changes during the connection sequence and any change to it is displayed as a status update with a number indicating the new state preceded by a "#". If repeater are specified, only the first in the chain is shown in this display. To see the others, it is necessary to step through them using the SV command.

4. Connecting to other stations:

To communicate with another station error free and without interference or interruption, it is necessary to establish a "link" with that station. Once this is done, the TNC will be immune to othe traffic on the channel as well as connect requests from other stations and all information transferred is transmitted until it is acknowledged as correct (usually a limit of 16 repeats is specified, but this can be changed if necessary using the SNn command) Connection can be achieved using the AC command after the appropriate callsigns have been entered. If no destination callsign has been entered, the TNC will respond with a ? and take no action. Once connection has occurred you are informed of the event with the display:

(Bell) - Connected to XXXXXX

Once business is finished, the link is disconnected using the AD command (Or Ctrl-C in Chat mode (qv)) and disconnection occurs automatically. Automatic disconnection also occurs if 6 minutes of inactivity elapses or more than 16 repeats of a packet occur without acknowledgement.

5. Chat Mode:

For most users (ie with dumb terminals) Chat mode is the mode most commonly used for communicating in Packet. Usually the TNC is set to enter chat mode on connection to another station.

The use of chat mode is simply as follows:

1. Enter Chat mode on connect. This is usually done automatically on connect but can be entered manualy if desired using the MS command.
2. Type your message to the other station into the TNC Buffer. It will be echoed on the screen.
3. Send your message to the other station using Ctrl-J (linefeed) to send the packet. It is customary to send text a line at a time, an "over" being terminated by ">>>"; you can achie this in chat mode by simply sending a linefeed character after every <CR>. When the packet has been sent and acknowledged, continue typing. Unless you have inhibited the terminal during transmit or are using a buffered terminal, do not type while the tnc is transmitting.
4. If you hear a packet coming in from the other station and you wish to know what it is before y have finished typing, type Ctrl-D to enable the TNC to display the incoming text.
5. When finished with the QSO, disconnect the other station with Ctrl-C.
6. Note that, unless you have some method of inhibiting your terminal while the channel is busy, typing while there is a packet in transit will cause the interruption of the unit and interfere wit the packet reception, thereby increasing the number of retries on the channel.

Notes:

Other Chat Mode control characters are: (These can be changed with the SY command)

Ctrl-B Initiate Connect - equivalent to AC in command mode. This is used if manual entry is made into chat mode in an unconnected state and connection is desired.

Ctrl-C Disconnect - as described above.

Ctrl-D Enable display of received Data. See notes below.

Ctrl-ECause the CW identification of your station to be sent - equivalent to MI in command mode.

Ctrl-Q Allow the display of a single frame of received data - equivalent to OF in command mode.

Ctrl-T Retype the contents of the transmit buffer - equivalent to T(cr) in command mode

Ctrl-U Delete the current line in the input buffer.

Ctrl-X Delete the entire contents of the transmit buffer - equivalent to K(cr) in commanc mode.

Ctrl-S Stop display of text - equivalent to OQ(E) in command mode.

Escape Allows the user to exit chat or insert mode to command mode without disconnecting. If using a Baudot terminal, cr-cr will perform the same function.

Although this explanation of chat mode operation may seem a little confusing at first, there is gooc reason for its seemingly tortuous logic. The TNC is a minimum cost design and therefore, compromises had to be made. In the field of supporting a dumb terminal, a simple buffer editor facility was needed and this had to operate without interruption from incoming packets otherwise no sense would be made of the screen display after a few packets have been received. Therefore the following strategy was used:

The terminal takes precedence over the link. This means, that while you are typing text into the text buffer, the TNC services the terminal only and any text waiting in the TNC to be displayed is not sent to the terminal until either 40 seconds of inactivity has elapsed, a Ctrl-D or Ctrl-R character has been received from the terminal or the TNC input buffer contents have been sent using the Ctrl-J command. Text can thus be entered into the buffer without interruption by a dumt terminal and simple text editing performed using the commands mentioned above until you are ready to display received text.

Note also that, while you are typing with a dumb terminal, the TNC may miss parts of packets bein received on the link due to the fact that each incoming character on the RS232 line causes the Z80 processor to be interrupted. The simple solution is either to monitor the channel audibly and just not type while packets are being received on the channel or use a computer with a simple split screen terminal program that assembles and edits lines of text for the TNC and then sends it to the TNC at high speed followed by a linefeed and a Ctrl-J instead of a purely dumb terminal. With this type of operation, console echo may be turned off using the SE(D) command if desired. Although in theory, operation of the TNC with a dumb terminal seems to be a bad compromise, in practice it turns out to be fairly satisfactory. Ideally, a program should be written that supports the TNC as blocked device in its block mode (qv).

6. Using the TNC in Unconnected Mode:

It is possible to use the TNC to transmit and receive packets without being connected to a specific station. To allow unconnected transmitting, you must first enable unconnected operation using the MU(E) command which prevents the TNC from responding to SABM Packets (connect requests). Now, when you enter chat mode, transparent mode or use block mode, operations are exactly as in connected mode except for one major difference. When connected, information in the transmit buffer is deleted when acknowledgement is received. In unconnected mode, no acknowledgement will be received and therefore you must manually delete the information with the Ctrl-X command in chat mode or K in command mode.

Unconnected receiving is enabled by default and all packets received will be displayed.

Commands which modify receiving characteristics are:

SG(E/D) This command allows the reception of "garbage" and will display every packet received whether it is validly decoded or not. Useful sometimes if propagation is poor. Connected operation is not possible with this mode enabled.

OA This Command allows the specification of two callsigns in the usual fashion and allows only packets to or from these stations to be displayed.

OO(E/D) Allows no packets to be displayed other than those addressed to your TNC.

7. Queue Mode:

Queue mode is controlled by the OQ(D/E) and OF commands. Ctrl-S is equivalent to OQ(E) and Ctrl-Q is equivalent to OF. When queue mode is enabled, nothing is transmitted from the TNC to the terminal and all messages and information is queued in the TNC. The terminal can thus be disconnected and used for other things or switched off. Furthermore, if queue mode is invoked in the unconnected state, and another station connects to your TNC, the contents of the Beacon Message Store (which normally contains the characters "-Q-" but can be filled with any message desired by entering it into the transmit buffer using the I command and then transferring it using the BS command) is transmitted to the calling station. When memory is full (up to 26 A4 pages of info with 16k of RAM) no further incoming messages are stored, and any station trying to connect is sent a "Wait" information frame and no further messages will be acknowledged until you unload the memory with an OQ(D), Ctrl-Q or OF command. To limit the amount of data saved in Queue mode, it is recommended that the OO-E command is issued to prevent the buffer from filling. A busy channel can fill the 16k buffer in about 15 minutes if this is not done!

8. Transparent Mode:

Transparent mode offers the user the facility of using the TNC as a perfect invisible link, giving th user the opportunity of simulating direct connection to a remote computer as if connected to it vi RS-232 Cable. In other words, all characters sent down the line to the TNC will appear at the other end. Transparent mode is enabled using the MX(E) command and entered - as in chat mode - either automatically on connect (if the ML(E) command is enabled, as it is by default) or manually using the MS command. The rules the TNC follows in transparent mode are as follows:

1. No characters are echoed to the terminal regardless of the setting of the SE(E/D) command.
2. Every byte, 00 to FF hex, received from the terminal is transmitted by the TNC.
3. Only data received by the TNC is sent to the terminal. In other words, no headers, control fields status updates, etc, are displayed.
4. No data transmissions can occur while the TNC is not connected to another.
5. No TNC commands are accepted; they will simply be transmitted as data.
6. Packets are transmitted when either (i) the transmit buffer is full and the terminal is held off l the CTS line; the buffer is then transmitted until empty, whereupon the CTS line is set to true again, or (ii) there is a pause in the string of characters received from the terminal. This pause is set by default to 1 second, but may be altered to any time interval using the SKn command where n is the number of 50ths of a second.
7. Exiting of transparent mode occurs automatically on disconnect if the ML command is enabled or can be exited manually using the following procedure. Wait until the transmit buffer is empty an a further second has elapsed (the same time as specified in the SK command) and then send three exit characters (Ctrl-C by default but can be specified to be any character of choice by the MEn command) and then waiting a further second. The Command mode prompt (:) will then be displayed.

One of the main advantages of using transparent mode is the ability to use sophisticated modem programs or mailbox programs with the TNC without having to program a special interface to access it as a blocked device. I have, in fact, used terminal programs using the XMODEM or Christensen protocol to transfer files over a transparent link.

9. Block Mode:

Block mode allows the TNC to be addressed and accessed as a blocked peripheral device on a computer through the RS-232 port. It is enabled by using the OB(E) command. The basic rules for block mode data input/output are as follows:

1. You are always in command mode and therefore the TNC will accept commands at any time when the : prompt has been displayed.
2. All output from the TNC is consistent and is preceded by a flag byte followed by data. (See below)
3. All unnecessary blanks, carriage returns and linefeeds are suppressed.
4. The only way of entering Data for transmission is by using the I command in the form: I followed by 2 byte binary count (low byte first) followed by the bytes of data. If the count entered is too large to fit into the available memory allocated to the input buffer, a ? is return immediately after it has received the count bytes. Once information has been entered, it may be transmitted to the destination station if connected using the AT command.
5. If Queue mode is enabled, either by OQ(E) or Ctrl-S, the prompt character (:) will be returned with its high bit set if there is Data waiting in the TNC for the terminal. This allows polling for data while in Queue mode and can make for more efficient programming by only allowing data to be output by using the OF or Ctrl-Q commands when data is present.

Block Mode output data presentation:
Data output starts after a ' character and a one byte binary count (cnt) followed by cnt bytes data.
Unnumbered Information Data starts after a "/" character and a one byte binary count (cnt) followed by cnt bytes of data.
Packet Address Field follows after a "." character.
Packet Control Field follows after a "-" character.
Packet PID Field follows after a ";" character.
Packet Status Updates follows a "#" character.
Hints when using Block Mode:

1. Enabling leading zeros may facilitate program efficiency in decoding numbers
2. It may be useful to use a different number radix (hex, binary or octal) depending on how you wish to decode numbers in the computer.
3. When using high baud rates it is advisable to wait for a character interval between sending characters to the TNC when entering commands. This allows the TNC time to respond to them and allows error detection if the TNC is monitored for a "?" character after each character is sent to it. (See the I command mentioned above) With the I command, once the count has been successfully entered, data may be sent to the TNC at the maximum rate without pauses.
4. No count byte is sent after the "." character and before the variable length Address field header. If you are not too worried about decoding the PID and Control fields which usually follow the Address field, simply scan the input stream for a carriage return which marks the end of each packet header.
5. It will be noticed that all packet headers are displayed in a fixed format, i.e: Address, [PID], CTRL. PID is only given in information frames and is hence in brackets. This simplifies the decoder somewhat in that only `',#,/` and `.` characters need to be scanned for, with the - and ; characters being expected after the Address field only. Indeed, one "bug" is to scan for th ; character in the main loop; this will cause the system to "hang" when someone repeats throug the TNC since this also causes the emission of a ; character.

Miscellaneous Instructions and Information.

1. Aborting Operations:
If you wish to stop the TNC from repeating transmissions of any packet, issue the AX command.

2. Alteration of default values for HF Operations:

1. Baud Rate. HF Baud rate of 300 Baud is set using the SJ-3 command.
2. Preamble. With the slower baud rate, less preamble bytes are needed. The number may be altered using the SFn command. 15 is a reasonable number use for HF operations while 50 is used for VHF 1200 Baud.
3. Number of retries. This may need to be altered to suit conditions. The default is 16 but may be set to anything from 1 to 255 using the SNn command.
4. A lot of experimentation is needed to set up the TNC for reliable HF communications. Things to experiment with are , better input filtering and alternative input modem which can be linked int the 4503.

3. Repeater Operations:
Since the TNC supports the complete AX.25 level 2 protocol, it is able to be used as a packet repeater. When the station is called on to repeat packets, it does so automatically and merely displays a single ";" character to indicate to the user that is has done so. It is possible to monitor what has been repeated using the MV(E) command and also to disable all AX.25 repeat facilities in the TNC with the MR(D) command.

4. Beacon Operations:
Sometimes it is desirable to send out a regular message as an unnumbered information frame. This could be to call CQ or to identify a repeater as active. This is done by using the beacon commands o the TNC. One use of the beacon message is mentioned above while discussing Queue mode. The other use is mentioned here. To activate the beacon, the following steps should be followed:

1. Use the I command to insert the beacon message into the transmit buffer.

2. Transfer the message to the beacon message store using the BS command.
3. Set the beacon destination and repeater callsigns using the BD and BV commands in the same way as you would use the SD and SV commands. This allows you to call via a repeater if you wish. The beacon destination can be set to anything you desire such as nothing, CQ, QST, BEACON, etc.
4. Set the beacon timer to enable regular transmission of the message by using the BTn command where the total interval between transmissions is about 10*n seconds. If the channel is busy, the TNC automatically prolongs the beacon timer interval. Setting the timer interval to 0 switches it off.

One further use for the Beacon system is to allow the storage of callsigns for a repeater chain which can be rapidly recalled to active use by using the BC command. NOTE: PLEASE do NOT send unattended beacons!

5. Squelch Backoff Operation:

This TNC allows the connection of a squelch line allowing interference free sharing of a voice or RTTY channel. Normally, the TNC "listens" on the frequency before transmitting and does not do so until the channel is clear. Unfortunately, it only detects other activity that will trigger the Carrie Detect circuitry in its modem; something that voice communications or weak, undecodable signals may not do. For this reason, an extra "ear" in the form of the squelch line can be used to inhibit it provided this facility is enabled by using the SB(E) command. The Squelch backoff line can also be used to allow multiple TNC's to operate on a single transmitter by connecting an inverted PTT live to it.

6. TNC Memory Management:

On startup, the TNC software automatically detects how much memory is in the system and reserves a certain amount for various operations as follows. 330 bytes (approx) are reserved for variable and flag storage and processor stacks, 2048 bytes are reserved for the transmit buffer while the rest of the memory is available for packet reception and decoding, received data buffering, packet encoding for transmission and repeater packet storage awaiting retransmission. I' is possible to find out how much memory is available on the system by issuing the MM command which will return the number of 256 byte pages available and also to reset the number of pages to reserve for the transmit buffer using the SM command. Note: the execution of the latter commanc will erase the contents of the transmit buffer as well as any date queuing in the TNC.

7. VADC Operation:

The TNC is capable of operation under the VADC protocol as well. Although VADC is now not as popular as AX.25, It has some advantages in that it uses less overhead in terms of header bytes, providing a faster link. Differences are as follows:

1. Every station must have a VADC "address", this being a number between 1 and 254 and is entered using the SA Command. Connection is not possible without entering this number. Connection is established using callsigns while subsequent communication is performed using the Address byte. Callsigns are entered as in AX25 but without the SSID. Repeater operation is not possible using VADC.
2. VADC mode is selected by the originating station by using the SX-E command. The receiving station (if using one of these TNC's) will be automatically selected.
3. The status report is slightly different in VADC mode: 45 /G0BSZ / ------ $ / //LUD/E/RL
The initial number is your VADC Address while the $ indicates an unconnected state; this being replaced by the Address number of the destination station when you become connected.

Unconnected operation is also possible but Unnumbered Information frames are not defined in VADC and therefore numbered information frames are used. No callsigns are sent in VADC except in the action of connection, therefore it is wise to send a CW identification occasionally.

Header information consists of a "." character followed by the source address number of the packet.

Repeater operations are not supported by the VADC protocol.

8. Unattended Repeater Operations:

It is possible to use the TNC as a remote, remotely controllable repeater as follows:

1. A PTT Timer and hardware transmitter timer are essential.
2. Connect the Serial Data In connection to +5V for remote operations where no terminal is connected.
3. If it is desired to control the TNC remotely by connecting to it and sending it commands, conne R10 to ground rather than to +5V.
4. When in unattended mode as defined selected by 2 above, the CTS line will provide a square wave all the while the TNC is functioning normally. A watchdog circuit is then necessary to detect this square wave and reset the TNC if it disappears for more than 20 seconds.

If the remote control function is enabled as in 3 above, the TNC will respond to connect requests and respond to the following text commands:

ON	Activates repeater function
OFF	Deactivates the repeater function
FLAGS n	Sets the number of preamble flags to n
STATUS	Displays the TNC Status.

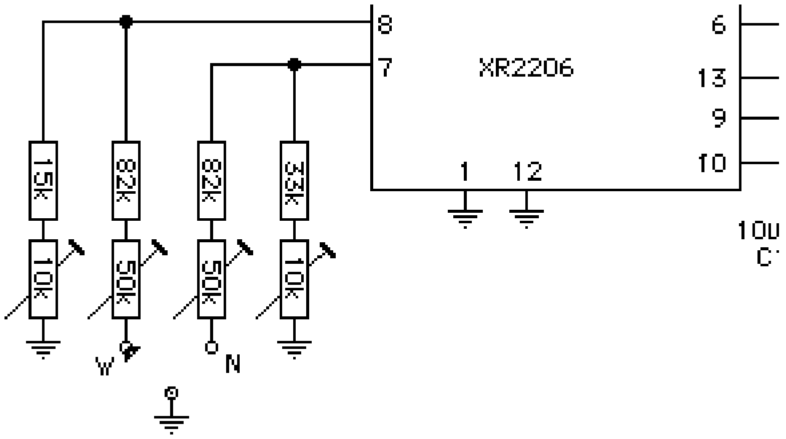
In addition, any valid TNC command can be sent to the TNC by preceding it with a "/" character anc ending it with a carriage return (cr). Eg, "/SFn(cr)" will set the number of flag bytes to n, and s on. Take care with these commands though, it is possible to issue "fatal" commands such as, eg. putting the TNC into standby! Command errors are indicated by the transmission of "ERROR" in response to the bad command.

In conclusion:

The TNC described here is not intended to be a state of the art design; indeed, it is a minimum cost minimum component design for people wanting to experiment on packet without having to spend the high prices of more sophisticated devices. I have been running this TNC in one form or another for almost 4 years and have had no trouble with it. Indeed, I have found, under some circumstances, that is has outperformed the more expensive TNC-2 clones!

Modifications:

1. C4 can be 2n2F to prevent signal degradation by wide-band noise.
2. R8 can be taken to +12V to increase decoder hysteresis.
3. R3 can be altered to 470kΩ or 510k to increase decoder hysteresis.
4. If you really want to try the on-board RX modem for narrow shift working, add 82kΩ to R5. This improves the time-constant. This could be switched in and out.
5. If you only want to use a single pole switch on the 2202 chip to switch the TX tones use the circuit below:


8
7
XR2206
6
13
9
10
1
12
10U
C
15k
82k
82k
33k
10k
50k
50k
10k
W
N