

INTRODUCING TheNet X-1J Release 4 by G8KBB

This version is a modification to TheNet X-1J release 2 and includes a small number of changes that I had promised. Specifically the changes from TheNet X-1J release 2 are :

- The buffer loss problem has been fixed
- The ARP table is automatically updated from ARP requests & replies
- The node maintains a second heard list of L3 Netrom nodes that gateway through the node
- The RxDeviation & Smeter ADC channels may also be used as arbitrary inputs instead
- A bug in the copying of digipeated frames in KISS mode 2 has been corrected
- A defensive port flush function has been added
- Link list integrity checking has been extended to check reverse links
- A bug in handling of BBSALIAS, HOSTALIAS and DXCALIAS has been fixed
- The USER list shows circuit choke status for patchcord connections.
- BBS, HOST and DXCLUSTER are cleared properly on cold start
- The ADC\* text messages are now 12 characters long
- The ACL checks the end user call as well as the node call in accepting connections.
- Lines that start with '#' are ignored by the switch
- A default IP route entry may be made

The ARP table is managed by sending ARP requests and by monitoring ARP messages received. The BBSALIAS etc commands were compromised by the inclusion of the TexNet interface and have been corrected.

The L3MHeard command controls a second heard list. It shows each station that passes Net/Rom frames to the node for gatewaying. Each entry shows the time since last gateway, port, frames heard and the callsign of the last destination node that the sending node was sending to.

The USER command has been enhanced to show, for each patchcord entry, whenever one or other circuit has been choked.

If one or other of the ADC channels used for measuring RX Deviation or signal strength are not required for those purposes, they may be used for other purposes in a manner similar to the other 2 ADC channels.

A menu driven windowing patch utility with context sensitive help is included.

Utilities for printing passwords in a sysop-friendly manner and a utility for changing the help text attributable to John Bednar WB3ESS are also provided. Modifications for the MFJ1278C by WB0YRQ are included in the bankswitching details. A utility to manage the node configuration by N8PTT is included.

The previous releases introduced the following :

- An S-meter function in the heard list
- A Voltmeter function for the other two ADC channels
- Support for the TexNet '\*\*\* LINKED to' syntax
- A bug fix to the BBS, ALIAS and DXCLUSTER commands
- The ability to change the user's command prompt list
- The ACL function has had a feature added to speed it up
- A complete IP router,
- The Receive Deviation meter,
- Control of 'Slime Trails',
- Information text messages 160 characters long,
- Multiple line INFO, CTEXT and BTEXT messages,
- Nodes broadcasts occur 60 seconds after power up,
- Optional reconnect to the node after remote disconnect,
- PARMs, MODE etc may be managed by offset & value,
- Digipeat uplink & downlinks may be selectively enabled,

- Level 4 retries ( min ) is now 1,
- An MTU command allows MTU settings to be changed for IP use,
- Node alias handling is optionally case sensitive,
- TALK will optionally pass 8 bit data,
- The RESET bug in versions prior to TheNet X-1J has been cured,
- The ability to remotely set the node's alias,
- The ability to listen for 3 extra aliases & invoke BBS etc,
- Selective node broadcast control for 'hash' nodes,
- A UI command to send arbitrary UI commands,
- Access control list capabilities,
- Multi-user conferencing ( the 'TALK' command ),
- A CWID keyer,
- Better SYSOP authentication,
- MHeard list showing callsigns, packets heard & time since last heard,
- MHeard list shows whether a station is a node and / or TCP/IP station,
- A Closedown command to remotely shut the node down,
- A DXCluster command that operates like the BBS & Host commands,
- A Btext command to set the node's beacon text message,
- A Ctext command to set an optional alias connect text message,
- The ability to enable or disable any command,
- Improved command prompting with only valid commands shown,
- Routes show optionally as alias:callsign or callsign alone,
- Additional control over system reset,
- KISS as an alternative to the crosslink protocol,
- Hardware handshake controlled host mode operation,
- MODE command for configuring additional parameters,
- BBS command to auto connect to a remote BBS,
- HOST command to auto connect to another BBS or Host,
- BYE or QUIT commands to disconnect,
- STATS command to display internal statistics,
- MANAGER command for system manager access,
- AUDIT command to set system audit levels,
- Bug fixes ( e.g. info messages too long crashing the node ),
- Changes to the NODES command,
- An improved nodes broadcast algorithm for the crosslink port,
- Split port nodes broadcast intervals,
- Ability to enable & disable nodes broadcasts selectively on each port,
- CQ apologises nicely if disabled,
- Most Escape commands have been replaced with MODE parameters,
- Beacon messages may be digi'd,
- CALIBRATE command for remote checking of Tx deviation,
- LINKS command to show current level 2 links,
- Configuration of the beacon period,
- Auto routing of 'connect' to either BBS, DXCluster or HOST,
- Remote dump of entire neighbour lists for all nodes.

Dave G8KBB @ GB7MXM  
QTHR

This disk contains the distribution version of TheNet X-1J release 4.

#### 1. Hardware Requirements

The software runs in a Z80 based TNC2 or similar clone such as the BSX2. It is installed as described in the bankswitch mods file, but essentially all it needs is a single piece of wire from pin 8 of the modem disconnect header to pin 1 of the EPROM.

There is also a version for the PK96 that needs no bankswitch mods but may need change for the DCDB select on pin 23

The EPROM needed is a 27512, rather than the 27256 of a normal TNC2. Pin 1 of the EPROM is bent out from the socket and connected as described above.

The TNC2 needs to be set to the higher clock speed of 4.9 MHz as a minimum. Whilst this version of TheNet is more efficient than others, it is not sensible to run at the slower speed.

The TNC needs at least 32K bytes of RAM. If a PK96 is used with extra RAM only 32 K will be used.

The TNC needs to have DCDB from the SIO/0 connected to pin 23 of the RS232 connector. See the section on RS232 later in this note for more information.

## 2. Installing over TheNet X previous versions

If you are replacing a previous ROM with TheNet in it, be sure to do a coldstart ( you may need to remove the battery link to force this ).

## 3. Files

The files on this disk are :

thenet1.xlj	Part 1 of the code
thenet2.xlj	part 2 of the code
readme.*	This file
configur.*	Installation guide
userguid.*	A user's guide to the node
overview.*	The sysop's manual
quickref.*	A handy quick reference guide
bankswit.*	Information on the hardware bankswitching
intro.*	The brief release note
smeter.*	Details on the Smeter hardware
devmeter.*	Details on the deviation meter
patch.exe	A windowing driven patcher
intel.exe	An Intel hex file dump utility
intel.c	The source of the above
motorola.exe	A Motorola S1 type file dumper
motorola.c	The source of the above
sethelp.exe	Utility to change the ROM help text
sethelp.c	The source of the above
printpwd.exe	Utility to print a password nicely
printpwd.c	The source of the above
smeter.csv	Sample data for smeter calibration
smeter.ljt	Circuit for an S meter in HP PCL format
ipxlink.doc	Ideas on node IP addressing
xljconv.c	Source for xljconv.exe
xljconv.exe	Utility to help manage node configuration
xljconv.txt	Documentation for xljconv.exe
thenetx.hlp	Windows 3.1 Help file of main documents

The files denoted '\*' are stored in two formats

.TXT    ASCII versions of the files  
.DOC    Word for Windows 2.0 format

The files contain the same except for the DEVMETER and SMETER, as the ASCII version does not contain the circuit diagrams or graphs. The smeter circuit diagram is shown in laserjet printer format ( i.e. HPPCL ) in the file smeter.ljt. As this is a large file, it has been compressed into the zip file smeter.zip.

The file thenetx.hlp is a MS Windows 3.1 help file. Whilst running windows, open file manager, select the file and either run it or drag it to the desktop to create an Icon under windows that contains most of the main documentation.

#### 4. Using it with a TCP/IP system

One of the reasons for the inclusion of the IP router was to help the development of IP networks. This is in two different ways :

- It allows a IP station that does not run 24 hours to run an IP router for the others in the area without leaving the PC running,
- It allows existing nodes to double as IP routers.

In scenario 1, where a station runs a TNC2 clone with a KISS ROM or similar, this software may be used instead of the KISS ROM. It should be configured to run KISS, Selective Copy on the RS232 port.

#### 5. The other version

The version that includes the HIGH and LOW commands is not included here. It used a form of bankswitching that clashes with the deviation meter.

#### 6. Problems

One of the main problems with release 2 was the loss of buffers over time. It was cured by a fault in the L2 receive handler when a reject frame was sent on connect establishment. In this case the receive buffer was not released. This has been cured.

One common problem. If your node appears to 'lose' nodes, look very carefully at the rates of node broadcasts and the setting of the algorithm control. The RS232 rate should be faster than the radio rate or it should be set to zero, and the alternative algorithm should only be enabled on the RS232 port if at all. Try switching it off to see if that helps.

The patcher has been altered. The code now comes as two parts and the patcher patches both at the same time. It needs & expects to be able to access both parts.

#### 7. RS232 connections

It is important that you understand the RS232 connections.

The RS232 port will operate in one of two basic modes, a dumb terminal mode and a protocol mode. Each of these modes has two basic types of operation. These are detailed further below.

The decision as to whether the node operates in dumb terminal or protocol mode depends on the setting of the DCDB input of the serial communications chip ( the SIO/0 ). In TNC2 clones, this is the signal on pin 23 of the RS232 connector, but in most TNCs you will need to either insert a link or a piece of wire, as detailed in your TNC manual. After connecting the links if needed, then leaving pin 23 open circuit will cause the TNC to operate in dumb terminal mode, and setting it to a negative voltage ( ie connecting it to pin 10 ) will cause it to operate in protocol mode.

If you use a BSX2 TNC, then you should connect pin 23 to pin 7 rather than pin 10 for protocol mode. You will also need to insert a link into JMP14 & make sure there is no jumper in JMP9.

If you use one of the other common TNC2 clones such as the PK80, MFJ1270, TNC-200 etc, then I understand ( but cannot confirm ) that you will need to connect a wire from JMP9 pin 1,2 or 3 - they are all joined on the PCB - to RS232 pin 23.

Incidentally, don't forget to increase the watchdog timer capacitor to give a 1 minute timeout ( C31 to 47uF in TNC2 clones ).

Finally, the protocol modes are CROSSLINK or KISS, selected by the firmware, and the dumb terminal modes are STANDARD or MODEM, again set by firmware. In the MODEM mode, hardware handshaking is used for connect request / disconnect. This is NOT the usual method of operation. Unless you are certain you understand, DO NOT use this mode !.

If you use the PK96 there are some problems with the RS232 port. See the text in the overview guide for details.

## 8. Introduction to nodes

If you have not used a TNC based node before, the following notes on the operation of the 'dumb terminal' port may be of use.

When pin 23 on the RS232 interface is high, and with a terminal or terminal emulator on the RS232 port, the node will not appear to respond to anything. If you send an 'escape' character, the node should respond with a '\* '. If it does not, look at the TNC status LED. It should be dimly lit with a little bit of a flicker. If not, the node is not operating. If it is, you have an RS232 problem. If you receive unsolicited garbage whenever the TNC receives data, you have pin 23 tied low and you are receiving crosslink or kiss frames.

If it does respond with a star and a space, you can now type one of 3 characters - C, D or P.

P will display or set the current password.

D will disconnect you from the node.

C will connect you to the node.

Try the C command. You should get a 'connected' message. Hit '?' followed by return and you will get a list of all commands.

Escape commands only work on the RS232 port when in dumb terminal mode. Don't forget to disconnect from the node before you disconnect the terminal.

The main documentation details the new commands added to the original TheNet 1.01. It does not cover the original commands unless they have been changed. The commands of interest are :

Connect  
Node  
Route

---

User  
CQ  
Parms  
Sysop  
Info  
Reset

These are described in user guide, with the exception of the Sysop aspects of the commands. These are as follows :

### 8.1 Node

The Node command may be used to make a manual entry in the Net/Rom node table. When this is done, it may also make an entry in the routes table if necessary. The syntax of the command is :

**NODE Callsign + Ident Quality Count Port Neighbour**

callsign is the callsign of the destination node

ident is the alias of the destination node

Quality is the node quality for the entry

Count is the obsolescence count to be given to the entry

Port is the level 2 port ( 0 for radio, 1 for RS232 )

Neighbour is the callsign of the neighbouring node to route through

So to make an entry that will never expire to ANODE:G8KBB-1 where the node is accessed directly on the radio port with a quality of 123, enter

**NODE G8KBB-1 + ANODE 123 0 0 G8KBB-1**

If the same station is not heard directly but is accessed through node GB7TXM-2 over the radio, and the entry will expire when its obsolescence drops to zero from an initial count of 8, enter :

**NODE G8KBB-1 + ANODE 123 8 0 GB7TXM-2**

An entry may be deleted by substituting '-' for '+'.

### 8.2 Route

The routing table tells the node the routes to neighbouring nodes. It can also be used to make a manual entry as follows :

**ROUTE Port Callsign [ Digilist ... ] + pathquality**

Port is the AX.25 port number, 0 for radio and 1 for RS232

Callsign is the neighbouring node's callsign

### 8.3 Parms

The PARMS command sets a number of basic node parameters. See the quick reference guide for a list of them.

#### 8.4 Sysop

Sysop is used to attain Sysop status. The node will challenge with 5 numbers. These are used to look up the corresponding password characters in the Sysop password table. This password is returned to the node on the next line. The password is stored in the ROM and is copied into RAM following a coldstart. From a locally connected terminal the password may be displayed or changed. using the <escape>P command.

#### 9. 'User channel' nodes

To create a 'user channel' node ( ie one that supports Net/Rom protocols on its RS232 port but not on its Radio port ), do the following :

- Set the radio port default quality to one less than minqual
- Bar node broadcasts on the radio port by setting MODE parameter number four to 2
- Ensure that no manual routes exist in the route table for port 0.

#### 10. EADG Nodes

If you put a node up in the UK that can be heard from East Anglia, may I encourage you to put the QRA locator in the Beacon text to enable the G1TLH Beacon analysis program to report accurately on propagation conditions.

73's

Dave G8KBB @ GB7MXM.#36.GBR.EU  
g8kbb@g8kbb.ampr.org

TheNet X-1J release 4.

## Overview of Operation

### 1. INTRODUCTION

This paper introduces the main features of TheNet X-1J release 4. This is an update to the previous paper on version X-1J2, and incorporates a number of changes including the following :

- The buffer loss problem has been fixed
- The ARP table is automatically updated from ARP requests & replies
- The node maintains a second heard list of L3 Netrom nodes that gateway through the node
- The RxDeviation & Smeter ADC channels may also be used as arbitrary inputs instead
- A bug in the copying of digipeated frames in KISS mode 2 has been corrected
- A defensive port flush function has been added
- Link list integrity checking has been extended to check reverse links
- A bug in handling of BBSALIAS, HOSTALIAS and DXCALIAS has been fixed
- The USER list shows circuit choke status for patchcord connections.
- BBS, HOST and DXCLUSTER are cleared properly on cold start
- The ADC\* text messages may now be 12 characters long
- The ACL for incoming L4 circuits checks the user callsign as well as node callsign
- Lines that start with a '#' are ignored by the switch
- A default IP route entry may be made

The software is a derivative of TheNet 1.01 by NORD<>LINK. Additional commands and bug fixes have been included in the release. In addition, the Patcher has been upgraded and a bug fix to MOTOROLA.C attributable to KH6ILT has been included. A utility to create remote reconfiguration files by N8PTT is included. My thanks to K4ABT, KA2DEW and WB4DDP to name but a few for assisting with ideas, encouragement etc. Credit is also due to John Bednar, WB3ESS, for the SETHelp utility and Al, WB0YRQ, for the MFJ1278C details. The file describing how to set up a node stack so that all or a subset of the nodes share one IP address is attributable to Fiona, G7ANH.

If your reaction is 'What I really want is .....', then please read on anyway, especially section 6.

### 2. STRUCTURE

One of the problems to extending TheNet is the 32 K EPROM limitation imposed by the architecture of TNC2 clones. The solution to this is to implement bank switching. For the BSX2 TNC and similar TNC2 clones, this can be achieved by the addition of a single wire as detailed in the bank switch modification file. This is at the expense of the HIGH and LOW commands. The other version that was previously available with HIGH and LOW in it is no longer supported as it is incompatible with the deviation meter.

The PK96 version supports the specific hardware features of the PK96. Baud rates for the ports may be set by use of the BRATE command and the software automatically selects either the 1200 or the 9600 baud modem. This file covers mainly the TNC2 version. At the end will be found a PK96 specific section that describes the differences.

### 3. NEW COMMANDS

The following commands have been added to the release.

BYE  
BBS  
HOST  
STATS  
MHEARD  
MODE  
MANAGER  
AUDIT  
TALK  
CALIBRATE  
LINKS  
ACL  
CLOSEDOWN  
BTEXT  
DXCLUSTER  
HELP  
CTEXT  
ALIAS  
BBSALIAS  
HOSTALIAS  
DXCALIAS  
QUIT  
IPROUTE  
ARP  
IPSTATS  
IPADDRESS  
IPBROADCAST  
UI  
MTU  
METER  
ADC  
ADC1  
ADC2  
ADC3  
ADC4  
ARPTIMER  
L3MHEARD  
BRATE ( PK96 only )

The following commands have been changed

CQ  
NODES  
RESET  
the <escape> commands  
SYSOP

The following features have been added to the code

- An Internet Router
- Ability to respond to three additional aliases
- A CWID keyer
- The command processor has been extended
- KISS mode operation on the RS232 port
- HOST mode support on the RS232 port
- Remote configuration of all parameters
- Additional textual help messages
- Support for a 4 channel ADC used for measuring RX deviation, voltage and signal strength
- Other changes as detailed herein.

In addition, a number of small changes have been implemented to satisfy the needs of specialist situations such as the ability to digi beacon packets.

Network management in this context does not just mean 'setting parameters remotely'. It means the ability to set, read and interpret various monitors and diagnostic tools. Version X-1C included the first part of the network management, the MANAGER privilege and the AUDIT process. Version X-1D extends the auditing and statistics significantly including internal CPU monitors. Version X1E includes most of the additions that are planned, and version 2 will complete the functions. No other release before version 2 was planned, but the need to produce a version with an IP router prompted X1J. Before moving to the joint development, I had agreed to finish off some long standing commitments hence release 2 of X-1J. Release 2 supported TexNet, an S meter, ADC commands for the other ADC channels, bug fixes, speedups and an enhanced command enable / disable syntax.

### 3.1 BYE or QUIT

There are no parameters to these commands. When entered, they terminate the session. Both commands do the same thing.

### 3.2 BBS

The syntax of the command is :

**BBS [ \* | ? | callsign ]**

With no parameter, the command connects to a station previously specified by the sysop. Setting the BBS destination is done by the use of the BBS command with a callsign as a second parameter. Setting the BBS to allow this may only be done by a sysop. The '\*' option may also only be executed by the sysop, this command clears a previously specified BBS.

The '?' option ( or any text if not sysop ), prints out the current BBS station setting.

If no BBS is set, the command issues an error message if it is invoked with no other parameters.

The idea of this command is that, like with the 'BBS' command of the 'BPQ software, a user may connect to the local BBS from the node.

The BBS, HOST and DXCLUSTER commands work by copying the callsign into a buffer. The copy operation does not compress the SSID into a single byte however, so a callsign such as KA2DEW -15 would occupy 9 bytes. I had the buffers set to 8 bytes length max in versions previous to X-1J release 2. Sorry.

---

### 3.3 HOST

The syntax of the command is :

```
HOST [ * | ? | callsign ]
```

This command is very similar to the 'BBS' command. It allows connection to a local host, BBS or other server. The difference however, is that as long as the TNC is not in 'crosslink' mode ( i.e. pin 23 on the RS232 port is high ), and if a callsign is not set, the 'host' command connects to the local port.

The idea of this command is that, like with the 'BBS' command of the 'BPQ' software, a user may connect to the local BBS, another node or server from this node. For example, if a print server were connected to the node in 'host' mode, this command would allow connection to it ( like the 'connect' command with no other parameter ). In KISS mode, setting a callsign or node alias allows connection to another system.

### 3.4 STATS

The STATS command has no parameters. It prints a number of internal TNC statistics. In this version, this is limited to the level 1 stats of the radio channel and the internal clocks, the level 2 ( AX.25 ), 3 and 4 statistics, and the CPU health checks.

For level 1, six pairs of numbers are printed, corresponding to the percentage of time the transmitter was on followed by the percentage of time the receiver DCD was on, for each of the last six 10 minute periods. The data is presented most recent period first. Two pairs of numbers are then displayed showing the transmitter underrun and receiver overrun. These are formatted as per the level 2 stats with port 0 followed by port 1 for the current hour followed by the totals for the previous hour. In the case of the RS232 port, underruns are not possible, and an additional error ( framing ) is included. The RX overrun includes overruns and framing errors.

For level 2, the following are displayed :

- Frame checksum errors
- Total packets heard
- Total packets received by the node ( ie sent to it )
- Total packets sent by the node
- Total receiver not ready packets sent
- Total reject packets sent
- Total receiver not ready packets received
- Total reject packets received
- Total number of link timeouts

For each of the level 2 statistics, four numbers are shown. The first two are cumulative totals over the period of one hour, incrementing in real time. The last two are the totals for the previous hour. Each pair of numbers is the total for the radio port followed by the total for the RS232 ( crosslink ) port.

For checksum errors, port 0 shows CRC errors and port 1 shows ( when in 'crosslink' protocol mode only ), checksum errors. As HDLC errors can be triggered by noise, acceptance of CRC errors is conditioned by the state of the DCD line. If DCD is on and an error is signalled, it will be added to the count. This reduces the false counts, but does not eliminate them. Distant stations that keep the squelch open ( just ) without being properly heard will result in lots of apparent errors.

For level 3, the number of level 4 frames gatewayed between nodes is displayed.

For level 4, the number of transport frames sent and received by the node are shown.

For level 3 and 4 statistics, two numbers are shown. The first is the number of frames accumulating for this hour, and the second number is the total number of frames for the previous hour.

For CPU health checking, two statistics are shown, the CPU loading and the buffer usage. Each looks like the level 1 stats with 6 numbers corresponding to the last six 10 minute periods.

The CPU loading shows the number of times, divided by 100, that the CPU makes it around its basic internal scheduler. For a node just switched on, receiving nothing, this will be about 470 ish for a 4.9 MHz clock. With lots of nodes, a heard list of 20 stations and 70-80% activity on the radio channel for it to listen to, this can drop to about 350ish. If it drops to double figures, worry, as the CPU is beginning to thrash. At low double figures, the CPU is pretty much working flat-out. Time to up its clock rate !.

The BUFFERS statistic shows the minimum number of free buffers that the software had available to it during the last six 10 minute period. This indicates whether the TNC is failing to deliver data passed to it for onwards transmission, as well as how much data is backed up waiting. Additional stats needed to analyse this properly are not yet being collected.

The display also shows the elapsed time since the last warmstart followed by the running time since the last coldstart. Each number is the number of hours of operation.

The final set of statistics shown are the port flush values. These are reset on coldstart. The first parameter is for port 0 ( radio ) the second for port 1 ( RS232 ). Each shows the number of port flushes that have happened since coldstart. A port flush occurs when data is queued pending transmission and cannot be sent. It is designed to protect node integrity when a channel is blocked ( for example by a fault ). More specifically, for each port, the node sets a flag whenever data is queued. It sets a flag when data is sent. Every 10 minutes it has a look at the flags and if data was queued but not sent, it sets an internal warning flag. If the warning flag was also set in the previous period then evidently data is building up and nothing is being sent. It then purges the queue of pending data for that port and increments the counter. This is not a perfect defence, but should usually suffice. If you find that the counter is incrementing, then look for a cause. It may be that the port flow control is stopping transmission. It may be that the radio port DCD signal has been on continuously. It may be a fault in a crosslink.

### 3.5 MODE

This command is similar to the PARMS command, and includes the new syntax described in section 3.32.

It allows a number of other features of the software to be configured remotely. It removes the need for most of the host mode <escape> commands.

The following parameters may be configured :

- The host mode
- The CWID send period
- The CWID keyer speed
- The port nodes broadcast control
- The crosslink / kiss control
- The Tx delay
- The full duplex flag
- The RS232 port node broadcast interval
- The node broadcast algorithm

The beacon period  
The 'connect' redirector  
The 'help message enable' flags, case sensitivity & TALK 8 bit flag  
The 'hash' node broadcast port control  
Whether the node will listen for the extra aliases  
Whether remote disconnect causes reconnection to the switch  
Control over 'slime trails'  
Control over digipeating up/down links

In operation, it behaves just like the PARMS command.

The parameters are as follows :

### 3.5.1 Host mode control

This parameter controls the 'host' mode. This is the mode of operation of the RS232 port when pin 23 is 'high'

The valid values are 0 or 1.

In mode 0, the port operates as per the standard node specification. Mode 1 is designed to allow connection to hosts or modems or similar equipment that expects a 'CD' type of signal to signify that an incoming / outgoing connection is called for.

In mode 1, the <escape> C and <escape> D commands are disabled and the other <escape> commands do not operate when connected. Instead, hardware handshakes are used to control connections to and from the TNC.

The TNC monitors pin 20 to determine the state of the host, and signals state changes to the host with pin 5. When an incoming connect request is received ( by the 'c' command with no parameters or by the 'host' command ), the TNC raises pin 5 to signal the connection and expects pin 20 to change state in response.

When the host wishes to connect to the TNC, it signals on pin 20 and the TNC responds with by changing the state of pin 5.

It handles disconnects in a similar manner. Either the node or the host may initiate disconnects.

This mode is experimental, changes may be made to its operation. It is designed for modems, print servers or hosts such as UNIX system TTY login drivers.

### 3.5.2 CWID control

The next two parameters control the CWID keyer.

The first parameter is the CWID repeat period in seconds. Valid values are 0 to 3600. 0 disables it but do not set it below 120 apart from to disable it.

The second parameter controls the keyer speed. Specifically, it sets the number of 10 millisecond periods per dot and per inter symbol delay.

The speed of sending is 120/n, so setting n to 6 gives 20 wpm. Valid values are 4 to 10, corresponding to speeds of 30 and 12 wpm respectively.

### 3.5.3 Node broadcast control

This parameter allows control to be exercised over which ports nodes broadcasts are sent. Valid settings are 0 - 3.

Value 0 disables node broadcasts. Value 3 ( the default ) works as normal. A value of 1 enables broadcasts on the HDLC port only whilst a value of 2 enables broadcasts on the crosslink port only.

### 3.5.4 Crosslink / kiss

This parameter is used to set the communications protocol used on the crosslink port when pin 23 is tied low.

The valid values are 0, 1, 2 or 3

Mode 0 - standard crosslink protocol enabled  
Mode 1,2,3 - use KISS instead of crosslink.

In mode 1, KISS simply replaces the crosslink protocol In mode 2, packets received from the radio part that are not intended for the node are copied to the RS232 port in KISS mode. Similarly packets received on the RS232 port that are not intended for the node are sent to the radio port.  
In mode 3, all packets received on one port are copied to the other port as well as being analysed by the node.

These modes are not simply KISS implementations that replace the node, they run with the node.

Mode 2 is designed to allow a KISS application and a node to share a radio without interference with each other. The point is that the PC TCP/IP system can be switched off whilst leaving the node running to allow others to use it.

Mode 3 is a debugging mode. One problem when faultfinding on a node is that it is impossible to see what the node is seeing on the channel without replacing the ROM. By setting this mode, it is possible to connect a KISS application to the RS232 port and observe what the node is seeing.

Mode 3 is also designed to allow a PC running AXSTATS to be connected to the RS232 port to allow logging and analysis of channel performance from the node itself. Note that packets initiated by the node for one port will not get copied to the other.

### 3.5.5 Tx keyup delay

This parameter sets the TX keyup delay in 10's of milliseconds. This was previously done using an escape command.

### 3.5.6 Full Duplex

This parameter sets or clears the full duplex control flag. This was previously done using an escape command.

### 3.5.7 RS232 nodes broadcast interval

When a crosslinked TNC is reset, it takes some time to learn about the nodes that the other TNCs can hear. Also, nodes heard by one TNC can take an hour to be notified to the others.

In order to improve this, this parameter may be used to change the frequency of nodes broadcasts on the RS232 port. When set to 0, the node operates as normal. When set to a non zero value, it will

---

broadcast the nodes on the RS232 port at that interval. Hence setting it to 600 would cause nodes broadcasts at 10 minute periods. The nodes broadcasts on the radio port will continue to occur at the basic rate set by the PARMS setting. The obsolescence count will be decremented at the basic rate, not at the faster RS232 rate.

### 3.5.8 Node broadcast algorithm

This value controls the algorithm used. Bits within the value set have significance as shown below. There is a problem with the nodes broadcast algorithm when many TNCs are crosslinked on RS232. In order to address this a variation to the algorithm has been implemented for experimental purposes. Feedback on its use is requested. Bit zero affects the HDLC port and bit 1 affects the RS232 port. When a bit is set to 1, the node broadcast algorithm is modified so that it will not rebroadcast on the same port a node heard on that port when the best quality neighbour is on that port. It makes little sense to use it on the HDLC port but what the heck, it is implemented for completeness. The only settings therefore that make sense are 0 and 2. These correspond to 'normal' and 'modified on the RS232 port' respectively. Setting it to 1 or 3 will result in some pretty weird effects.

### 3.5.9 Beacon period

This parameter sets the beacon interval in seconds. In TheNet 1.01, this was fixed at 10 minutes ( 600 seconds ). In this version, this parameter may be used to change it according the prevailing license conditions.

### 3.5.10 'Connect' redirector

In TheNet 1.01, when 'connect' is given with no destination callsign, the node attempted to connect to the local host port. In a crosslinked system, this vanished down a black hole. In previous versions of this code, the node attempted to connect to the station set by the HOST command, only trying the local host port if no destination was set by HOST. With this version, the node may be configured to connect to the station set by the BBS, DXCLUSTER or the HOST command depending on this parameter. When zero, connect attempts will go to the HOST station, when set to '1', it will attempt to connect to the BBS callsign. When set to 2 it will attempt to connect to the DXCLUSTER callsign.

### 3.5.11 'help message enable' flags

This word controls the sending of help messages, with each bit of the word controlling a separate function. Currently, only 8 bits are effective, these being as follows :

#### BIT    FUNCTION

BIT	FUNCTION
0	Whether the 'please wait, trying xxxx' operates
1	Whether all commands appear in help for sysop
2	Whether the 'goodbye' message is given
3	Whether a welcome message is enabled ( CTEXT )
4	Whether nodes are shown as 'alias:callsign'
5	If set, TALK data is passed as 8 bit data rather than clearing the most significant bit
6	If set, node aliases are deemed to be case sensitive
7	If set, enables the TexNet "*** LINKED to" interface

When bit 0 is set, and the BBS, HOST or DXcluster commands are given, then a message is sent from the node telling the user that a connect attempt is being made. This does not affect the 'connect' command itself, unless the command is given with no parameter as this is then equivalent of the BBS or HOST command.

When bit 1 is set, and if a sysop gives an incorrect command, the help screen shows all commands possible, including those currently disabled ( as by definition they are not disabled for the sysop ! ).

When bit 2 is set, then the use of the 'bye' command will solicit a 'goodbye' message from the node.

Bit 3 switches on and off the 'CTEXT' message. When enabled, and when a CTEXT message is set, then whenever someone uplinks to the node alias, the ctext message is sent immediately on connect.

Bit 4 switches the way in which nodes are shown when the ROUTES command is used. When set to '1', nodes are shown as 'alias:callsign'. When set to 0, they are shown only as 'callsign'.

Bit 5 controls only the passing of data in TALK mode. Normally, all data sent to the node has its most significant bit cleared, to eliminate parity or similar problems. This is not ideal for those countries that use the extended character set. When this bit is set, and only when in TALK, data is passed as 8 bit data. Note that this does not apply to an initial message sent on the same line as the TALK command.

Bit 6 makes node handing case sensitive. Normally, node aliases are forced to upper case for searching in the table and for user 'connect requests'. If this bit is set, these operations will become case sensitive. This could be very confusing for users unless they are aware of it and expect it. It allows node aliases to be entered as lower case, for example in setting the node alias and in forcing routes. Don't set this bit unless it is actually needed !.

Bit 7 controls the TexNet interface. If set, the TexNet "\*\*\*\* LINKED to" command string handling is enabled so that a TexNet to Net/Rom interface may be effected. This is described further in section 4.4. If the bit is not set, then a TexNet "\*\*\*\* LINKED to" string will solicit an error message.

#### 3.5.12 'hash' node port control

In certain networks ( notably the American ), there is a need to restrict the propagation of local nodes. This is done by using node aliases that start with a hash character ( # ) and instructing specific nodes not to broadcast routes to nodes that start with this character. This parameter does this by enabling each port to be individually enabled or disabled in respect to 'hash' node broadcasts. Bit 0 controls the radio port and bit 1 controls the RS232 port. When one of these bits is set, hash nodes will never be broadcast on that port.

#### 3.5.13 Extra aliases

If this is set to '1', then the node will listen for ( and accept uplinks to ) the aliases set in HOSTALIAS, DXCALIAS and BBSALIAS if they are set. If this parameter is set to '0', or if the respective aliases are not set, it will do nothing. If you do not use the aliases, set it to 0 to avoid wasting processor time.

#### 3.5.14 Reconnect to Switch

If this parameter is set to 0, the node operates as normally. If set to a non zero value ( i.e. set to '1' ), it operates in 'reconnect' mode. When a station connects to the switch, then uses the BBS, HOST, DXCluster or Connect commands to connect to another station, and then causes that remote station to disconnect them, then they are automatically reconnected to the node with a 'welcome back' message.

---

### 3.5.15 NoSlime

This parameter controls 'slime trails'. A 'slime trail' is caused when a remote node, whose identity is not known to the node, sends a transport connect request to the node. Subject to the settings of the port qualities, the node may make an entry in the node table in order to reply to them. Such entries are typified by having no alias associated with them.

Each bit in the NoSlime parameter controls a different function. Bit 0, if set, causes any stations without aliases to be 'hidden' when a nodes command is given. Bit 1, if set, causes the node to refuse to make slime trail entries in the node table. Before you use this feature, be careful to make sure that you understand the implications of doing so, as without fixed entries the node will refuse to accept level 4 connections from a station until it has heard their node broadcast.

### 3.5.16 NoDigi

This parameter controls the node's willingness to accept digipeated level 2 connections or to allow digipeated downlinks from the node. Each bit of the parameter controls a different function, as shown below :

BIT	FUNCTION
0	If set, do not allow digipeated connections to the node
1	If set, do not allow digipeated downlinks from the node switch

### 3.6 MHEARD

The TNC can be instructed to keep a list of the last 'nn' stations heard, where 'nn' is an integer between 1 and 100. It can also be disabled. The syntax of the command is :

**MHEARD [ nn ]**

The parameter is optional and only operates for the sysop. It sets the maximum length of the list. Setting to zero disables the function.

The heard list uses free buffers for the list, so a large setting means less RAM for the node software.

The list is maintained as linked list, with the most recently heard station first. The display shows the number of packets heard from that station and the time since it was last heard, in hours minutes and seconds. In addition, it shows the port on which the station was heard together with an indication as to whether the station is a node and / or a TCP/IP or a TexNet station. It does this by examining the PID byte.

Every hour the list is checked for stations that have not been heard for 12 hours, and any such stations are removed from the list.

To disable the internal updating of the list ( and thereby stop the CPU expending effort on the function ), set the size to zero rather than just disabling the command as described in 3.8. Note though that the node will not clear the list as updates have been disabled, so it will be up to 12 hours before the buffers used are freed. To accelerate this process, set the size to 1, wait until it has heard a station ( any one will do ) then set it to zero. This will free up all but one buffer immediately.

The heard list is the user interface to the receive deviation meter. Its operation is explained in section 3.31. If enabled ( ie if the METER parameter is not set to 0 ), then an additional column will be displayed in the heard list that will show the received deviation in kilohertz ( as nn.n ). It must be

---

remembered that this is derived by measuring the received signal audio level, and will not be correct in the case of a badly distorted signal.

The heard list may also show received signal strengths if enabled by the Sysop. In this case, the signal strength will be shown for each station in the list after the deviation level ( if enabled ). The display will be either dBm or the common 'S1 to S9' format according to its configuration. How accurate it is depends on the accuracy of its calibration. At best it may typically be  $\pm$  1dB relative accuracy, at worst totally misleading. In the same way that you can use the deviation meter to set up your station, pay heed of the signal level too. If you are 40 dB over the noise into a local node, consider dropping your signal 20 dB ( for example from 10 W to 100 mW ). Those around you also running high power may however make this difficult. The idea is to get in reliably with the smallest signal not to see how big a signal you can put into the node.

### 3.7 CQ

When CQ is disabled, the command now reports apologetically rather than simply ignoring the request.

### 3.8 ALL COMMANDS

There is often a requirement to be able to disable the connect command whilst allowing level 3 relaying. This is achieved by the use of a command qualifier, the syntax of which is :

**CONNECT [ + | - ]**

If '-' is entered by the sysop, then the connect command will politely refuse to work. This can be reversed by the '+' command.

This has no effect of layer 3 relaying. Also, the BBS and HOST commands will still allow connections to be made if they are enabled and set.

Further, the syntax is valid for ALL commands, for example the CQ command can also be disabled in the same way. Be careful though. The command is only accepted from the sysop, so if you disable the sysop and manager commands you will lock out remote management !.

### 3.9 NODES

When information on a node that is not known is requested, the program prints out an error message rather than giving the names of all known nodes.

When a node entry is made by the sysop, callsign checking is forced ON rather than being determined by the callsign checking parameter.

Don't forget that node alias handling may be case sensitive - see section 3.5.11.

The entire contents of the node table routes may be obtained by the sysop or manager by the command

**NODES \* \***

This will dump info on all nodes, one node per line, with the following format:

**Alias:call route1 route2 route3**

where route1, route2 and route3 comprise the quality, obsolescence count and port followed by the neighbour callsign for each of the 3 route entries for that node. If any of the routes are in use, a chevron will be shown by that route.

The extended command is only for sysop use as it, like auditing and conferencing, causes the node to be a source of a significant amount of data ( dumping a large number of node details can consume hundreds of buffers !!! ). It is quite possible that used indiscriminately, it could cause a warmstart of the node. Be careful.

### 3.10 RESET

The syntax of the command is now

**RESET [ anything-else ]**

Entering the reset command alone will do a warmstart. If any other parameter is entered, a coldstart is performed.

### 3.11 MANAGER

The MANAGER command gives the user extra privileges. In this version, this amounts to the ability to receive audit messages from the node. The level of auditing is set by the AUDIT command.

The privilege remains in force until cleared by a command that affects the user state. Specifically, these are, entering the TALK state, executing the SYSOP command, entering the MANAGER command and getting the password wrong, or disconnecting from the node. Failing to get the second password right when using the closedown command will also remove the manager privilege.

If the MANAGER command is executed by a user who connected to the node by a level 4 circuit rather than by a level 2 circuit, and if the level 2 timeout is less than the no activity timeout, the connection will never timeout as the clearing and reconnecting of the level 2 circuit will keep the process alive provided level 2 auditing is enabled. This allows the operation of the node to be logged remotely and continuously. Alternatively, if the level 4 timeout is greater than 10 minutes, level 1 or CPU auditing will keep it alive if level 2 is switched off. NOTE : I have a nasty feeling that there is something not quite right here - the link sometimes dies !.

A user with MANAGER privilege also has SYSOP privilege.

---

### 3.12 AUDIT

The syntax of the audit command is :

**AUDIT [ new-value ]**

where new value is an integer value. If no value is given, or the user does not have SYSOP status, the current mask value is displayed. Otherwise, the mask is updated and the new value displayed.

The mask controls the auditing of various events in the node. Not all values are used yet, but those that are, are :

BIT	USE
0	Level 1 statistics on 10 minute updates
1	Level 2 connects & disconnects
2	reserved for future use
3	Level 4 connects & disconnects
4	Level 7 limited events ( use of sysop )
5	Full level 7 auditing
6	CPU auditing messages ( 10 minute updates )

It is suggested that the usual settings can simply be 0 or 255.

For level 1, messages are sent every 10 minutes showing the percentage of time that the receiver detected carrier and the percentage of time that the transmitter was on.

At level 2 & 4, the messages are of all connects and disconnects, shown in 4 different ways :

- C Connect message received by node
- CA Connect message sent / Acknowledge received
- D Disconnect message received by node
- DA Disconnect message sent / Acknowledge received

In each case, 2 callsigns are shown. At level 2 these are the source and destination of the AX.25 link. At level 4, it is the remote node callsign and user callsign. Each message is preceded by an indication of the source of the message, such as "L2" or "L4".

At level 7, with bit 4 set and bit 5 clear, the only event currently audited is the use of the Sysop command, either directly or via the manager command. If bit 5 is set, then all commands given to the switch are audited, preceded by the callsign of the user who entered the command.

Bit 6 controls CPU health check auditing. If set, then whenever the internal CPU statistics are updated, messages are sent showing the CPU processor loading total and the minimum buffers level ( see STATS for more information ).

The audit mask value should be set to 0 when not actually being used. Do not leave it set to another value as this wastes processor time. Note also that full auditing on a busy node makes things worse. Treat it as a debugging feature !.

### 3.13 TALK

Talk is a conferencing command. It allows a number of stations to hold a simultaneous conference ( a bit like the CONFERENCE command of a DX cluster ). There is only one conference, and stations may connect to it by connecting to the node and issuing the TALK command. It may be exited by

---

disconnecting or issuing the command '/EXIT' at the start of a line. ( /EXIT may be abbreviated to /EX, and it is not case sensitive ).

Each line sent by a user is copied to all other users in the conference, preceded by the callsign of the user. The data will be sent as 7 bit data ( ie the most significant bit will be cleared ) unless the appropriate bit is set in the help flags ( see section 3.5.11 ).

Whenever a new station enters the conference, or a station leaves the conference using the '/EXIT' command, the other conference users get a message informing them of the event. These status messages are sent with the callsign of the node rather than the user.

Finally, when entering the TALK command, a message may be sent to all those users who are connected to the node but not otherwise doing anything. For example if GxABC enters the line

```
TALK Hello fred, can I have a chat, type 'TALK'
```

Then all other stations connected to the node, present in the USER list but idle, get the message

```
GxABC>> Hello fred, can I have a chat, type 'TALK'
```

displayed on their terminal.

Note that merely connecting to the node does not constitute being connected to the switch. Stations connected to the switch appear in the USER list.

### 3.14 SYSOP

The SYSOP command has been enhanced to increase the level of security offered. One problem of the old system is that the password is easily visible unless the user repeats the SYSOP command a number of times. Even then, correlation between passwords is easy, so the password needs frequent changing. To reduce the change period, and make it harder to discover, the node will accept a string of characters and scan it for the password. Hence a response of, say, 30 or 40 characters can be sent, with a random number of random characters preceding the actual data and a random number following it. This does not eliminate such attacks, but if used carefully, it makes it quite a bit harder to attack.

### 3.15 LINKS

This command shows the current level 2 links to the node. Displayed one per line, the two callsigns are shown followed by the link state, port number and current retry count.

### 3.16 CALIBRATE

This command allows remote calibration checks of the transmitter deviation. Its syntax is

```
CALIBRATE period [ toggle ]
```

The period ( 1 to 60 seconds ), is the time for which the transmitter will key up for with constant tone. It is undefined as to which tone will be sent. If the second parameter is given, the node will toggle between the tones every [toggle] seconds. The toggle must be between 1 and [period] seconds. If a period is not given, the user is not sysop or manager, or if it is out of range, the command is ignored. If the tone generator is busy because it is about to send a CWID sequence, a 'busy' message is returned. Note - quite often it can appear that the node has locked up having failed to transmit the full calibrate period. In fact, this is usually the hardware PTT watchdog in the TNC. The node thinks it is still sending but the hardware timer has removed the PTT signal.

---

### 3.17 DXCLUSTER

The DXCLUSTER command operates just like the BBS command in that it may be used to effect a connection to a DXcluster (assuming there is one nearby). It should be disabled if it is not intended to be used to access a cluster.

The syntax of the command is :

```
DXCLUSTER [ * | ? | callsign ]
```

With no parameter, the command connects to a station previously specified by the use of the DXCLUSTER command with a callsign as a second parameter. Setting the DXCLUSTER to allow this may only be done by a sysop. The '\*' option may also only be executed by the sysop, this command clears a previously specified DXCLUSTER.

The '?' option ( or any text if not sysop ), prints out the current DXCLUSTER station setting.

If no DXCLUSTER is set, the command issues an error message if it is invoked with no other parameters.

The idea of this command is that, like with the 'bbs' command of the 'BPQ' software, a user may connect to the local DXCLUSTER from the node.

### 3.18 HELP

The HELP command gives a message from the ROM. In general, it is expected that the message will be designed to assist new users in understanding the operation or configuration of the node. The message may span many lines, and may be changed when the ROM is programmed. As delivered, it contains a brief help screen detailing the main ( user ) changes to the code.

### 3.19 CTEXT

The CTEXT command sets or displays a message sent to a user who connects to the node by uplinking to the node's alias.

The syntax of the command is :

```
CTEXT [ * | message ]
```

With no parameter, the current message is displayed. If the user is also a sysop, and if text follows the command, that text is added to the current connect text. If the message starts with a '\*', the connect text message is deleted. Hence, to clear the message, type the command 'ctext \*'. This is a change in version X-1J from previous versions. For further information, see section 3.33

A message is only sent if there is a ctext message set and if the relevant bit is set in the mode command parameter as described in section 3.5.11.

### 3.20 BTEXT

The BTEXT command sets or displays the additional beacon text sent along with the beacon packets.

The syntax of the command is :

```
BTEXT [ * | message ]
```

---

With no parameter, the current message is displayed. If the user is also a sysop, and if text follows the command, that text is added to the current beacon text. If the message starts with a '\*', the beacon text message is deleted. Hence, to clear the message, type the command 'btext \*'. This is a change in version X-1J from previous versions. For further information, see section 3.33

Normally, beacon packets are UI frames that contain the node callsign and alias. If a beacon message is set, the text of the message follows the alias in the same packet. It is strongly suggested that beacon packets be kept brief !!!.

### 3.21 ACL

This is probably the most complex additional command in the program. It should be used with care, and only when you really understand its operation - mistakes can result in the need to go out to a remote site ( probably when it is cold and wet ) to reconfigure the node.

The command allows selective control, based on callsign, of a list of different events. The ACL contains two types of entry, a default value and zero or more callsigns, each of which are associated with a value. When one of the controlled events occurs ( such as an incoming level 2 connection or a nodes broadcast ), the ACL is scanned for an entry that matches the callsign of the sender. If a match is found ( but see below ), the value associated with that callsign is used to determine the action the node will take. If no match is found, the default value is used.

An ACL function mask is provided. This enables specific ACL functions to be disabled to speed up the node. Each bit of the mask enables ( if set ) or disables ( if clear ) the corresponding function. For example, if the mask is set to 2, the node will only check its ACL for outgoing level 2 connections.

Each bit of the value or mask controls a different function, as shown below :

BIT	OPERATION
0	bar incoming level 2 connection
1	bar outgoing level 2 connection ( downlink )
2	ignore nodes broadcasts from this station
3	bar gatewaying at level 3 to/from this station
4	bar incoming level 4 connections
5	bar outgoing level 4 connections
6	ignore SSID in matching an entry



So if for example an entry exists for a callsign G99XXX of 6, then the node will not allow outgoing level 2 connections to the node ( downlinks ), and will ignore node broadcasts from that station. Note that these commands only operate on the events themselves - if G99XXX creates a level 2 connection, the node will quite happily use it itself.

The 'ignore ssid' bit is used to match a callsign without regard to its SSID. This makes life interesting when finding a match, so the list is scanned twice, once for an exact match, and then for a match ignoring SSID if an exact match is not found. There can only be one exact match, but when searching for a match without using SSID, the first entry found will be used.

The syntax of the command is as follows ( 4 versions )

```

ACL * value
ACL & value
ACL callsign + value
ACL callsign -

```

If you are not sysop, or if ACL is given on its own, the current contents of the ACL are shown. The first form of the command changes the default value, the second form sets the mask enable value, the third makes an entry in the list, and the last form removes an entry from the list. It complains about syntax errors.

A few moments thought will show that the sequence of commands

```
connect to node
execute sysop or manager command
type the command ACL * 127
type the command ACL & 127
disconnect
```

is quite catastrophic. You will not be able to get back in again apart from via the host port and noone will be able to connect to or from the node. If you intend to experiment with the command, you should start by entering your own callsign with a value of zero to ensure that you can get back in again !!!.

The list can be used as an 'accept' or 'reject' list by judicious use of the default. To create a list that excludes specific calls, put them into the list with the required bits set in the value. The default should be zero. To create an 'accept' list, put entries in with the required bits zero and set the corresponding bits in the default. Individual bits may be used to create accept or reject lists for each function.

The command steals buffers at a rate of one buffer per four entries in the ACL. Also, a long ACL will slow the node down nicely - so think before you enter a long list.

This command is for experimental purposes - if you find any bugs, let me know please ( I have not fully tested the gateway bit for example ). Also, it is not intended for malicious use but to allow fine control to be exercised over backbone networks. If I get lots of negative responses back, the command will go !

The mask value should be set so that only those functions actually being used are enabled. For example, if the node has an ACL set to control level 2 uplinks and downlinks, it will also check its ACL for all other controlled operations such as level 3 gatewaying. This will result in much time being wasted searching the list for a result that is going to always succeed. In the example cited therefore, the mask would be set to 3. On reset, the mask value is set to zero, SO ALL ACL FUNCTIONS ARE DISABLED AND MASK MUST BE SET TO ENABLE THEM !.

The level 4 function has been extended so that it examines not only the originating node callsign but also the user callsign from that node in deciding whether to refuse a connection. Previous to X-1J release 4, it only looked at the node callsign.

### 3.22 CLOSEDOWN

The closedown command is used to shut down the node remotely. If successfully executed, the node will effectively stop operating until it is reset ( e.g. by a power up ). The node's configuration ( routes, messages etc. ) are not destroyed - the node simply hits a HALT instruction. You must be sysop to execute the command.

The syntax of the command is:

**CLOSEDOWN A**

The node will respond with 5 numbers just as for when the sysop or manager command was executed. Yes, you guessed, the node expects another password. Give it correctly and the node closes down completely. Get it wrong and you lose your sysop status. This obtuse and awkward syntax is designed to make sure it is not accidentally executed.

### 3.23 ALIAS

The ALIAS command allows the node's alias to be changed. The syntax is :

**ALIAS [ \* | new-alias ]**

If no parameter is given, or if the user is not SYSOP or MANAGER, the current alias is displayed. If the alias is deemed to be a valid alias, the node's alias is changed to the new one entered. Note that the algorithm that checks for the alias structure is a bit queer. It is however, the original algorithm of TheNet and I am loathe to change it for fear of side effects. Note too that the companion CALLSIGN command is NOT included - chaos is not something I crave. If the sysop gives the parameter of '\*', the node's alias is cleared.

### 3.24 BBSALIAS HOSTALIAS DXCALIAS

These commands are used to enable the node to respond to up to three additional aliases. The syntax of each is the same, and by way of example the BBSALIAS syntax is :

**BBSALIAS [ \* | new-alias ]**

If not sysop, if no new alias is specified, or if it does not pass the weird alias syntax checker ( see 3.23 ) then the current alias is displayed. If not, the alias is changed. If '\*' is given, the alias is cleared.

The aliases so entered are nothing to do with the node's identity. If a BBS alias is set, for example to MXMBBS, then the node will listen for level 2 connects to that alias. It will respond to them and will automatically invoke the BBS command. The user will also get the optional welcome (cctx) message and 'trying to connect to ....' messages if enabled by the appropriate 'mode' parameter.

The idea is that where a node sits on a channel that does not have access to the local host, BBS or cluster, the normal aliases of those stations may be enabled in the node to allow consistent access to the local services. Note that the three stations do not have to be a BBS, Host and cluster, it could be three BBSes or any other combination.

### 3.25 IPSTATS

The IPstats command has the same basic syntax as the PARMs and MODE commands. When invoked without parameters, it displays the current stats. Each statistic may also be altered by sysop, as defined in section 3.32.

In addition to the standard IP MIB, there is an additional parameter used to set the level 2 default modes, and the first entry in the MIB is used to enable or disable the router.

The complete set of IP MIB stats are included for compatibility with other IP systems, but several are not used. Also, the stats are 16 bit counters not 32 bit counters as in NOS. Like NOS however, the stats do not reset every hour, they must be cleared by the sysop. They will however wrap around at zero.

The entries are:

- 1 Port default modes
- 2 Enable / Disable the IP router functions
- 3 Default IP Time To Live
- 4 IP Received frames
- 5 IP Header Errors
- 6 IP Input Address Errors
- 7 IP Forwarded Datagrams
- 8 IP Unknown Protocols
- 9 IP input frames Discarded
- 10 IP Input frames Delivered
- 11 IP Output Requests
- 12 IP Output Discards
- 13 IP Output No Routes errors
- 14 IP Reassembly Timeout errors
- 15 IP Reassembly Required errors
- 16 IP Reassembly OKs
- 17 IP Reassembly Fails
- 18 IP Fragmentations completed OK
- 19 IP Fragmentation Failures
- 20 IP Fragmentation Creates

The default mode word may be set to 0, 1, 2 or 3. Each bit controls a port, with bit 0 controlling port 0 ( radio port ) and bit 1 controlling port 1 ( RS232 port ). When set to 1, the default mode for that port when sending on a level 2 connection will be Datagram. When set to 0 it will be by Virtual Circuit. The default mode is used when no other information is given, either by the ARP table or by the TOS bits in the IP header.

The enable / disable word may be set to 0 or 1. When set to 0, the operation of the router is stopped, when set to 1 the router functions.

The IP Time To Live ( TTL ) word is used to set the number of routers through which an IP frame may pass before it is discarded. It is similar to the node layer 3 TTL word. It may be set to any value up to 255, but values below 2 make no sense and are therefore not permitted.

The IP fragmentation reassembly timeout counter is not used as the node is just a router. It is left set to 30 seconds just to show which one it is !

The rest are just statistics. The patient user can have hours of fun working out which ones are not used ( or just think about it for a second or two ).

---

### 3.26 IPADDRESS & IPBROADCAST

These commands are used to set or display the IP addresses used by the node. The syntax of each is (by way of example):

```
IPADDRESS [ ipaddress ]
```

where ipaddress is in the form

```
nnn.nnn.nnn.nnn
```

where nnn is an integer in the range 0..255

So to set the node IP broadcast address to that used over here, the command would be :

```
IPBROADCAST 44.131.0.0
```

The IPADDRESS is the address that the node will respond to. It is used only as detailed in section 7. The IP broadcast address is the one used to denote broadcast packets that will be largely ignored. Note that port addressing is NOT currently supported. Anyone who finds this limiting, drop me a line and I'll see if I can change it.

### 3.27 IPROUTE

This is one of the two main databases used by the node. The IP Route table is used to tell the router where to send a frame for a specific destination. It maps addresses or address ranges to a gateway IP address and to sub-network ports. The ARP database then tells the node what station corresponds to that address and protocol. The node supports two subnet protocols, AX25 and Net/Rom.

The database is stored in an ordered list, in decreasing order of the number of relevant bits. This is to permit searching of the database when trying to find a specific destination. Given an address, it scans addresses with decreasing numbers of bits until it finds a match. The syntax of the command is as follows :

```
IPROUTE [address [ / bits ][ + port [gateway [metric]]]]
```

or

```
IPROUTE [address [ / bits ][ - ]]
```

In the first form, it makes an entry in the table, in the second it deletes one. Only sysop or manager may effect such a change. The parameters are as follows :

address	The amprnet address in the form nnn.nnn.nnn.nnn
bits	The number of significant bits (eg 44.131.0.0 / 16)
port	The port, either 0 or 1 for AX25 or n for Net/Rom
gateway	The optional gateway for this dest. nnn.nnn.nnn.nnn
metric	Currently not used, a numeric value

When an entry is made with a specific number of bits, the address is 'masked off' to that many bits, so enter an address of 44.131.16.31 / 24 and it will get entered as 44.131.16.0. The valid range for the number of bits is 0 - 32.

If the number of bits is set as zero, an IPRoute entry will be created of 0.0.0.0 and will function as the default route entry.

---

### 3.28 ARP

The ARP table maps a pair of address+port to hardware address+subnetwork mode. The address is either a destination or a gateway in the form nnn.nnn.nnn.nnn. The protocol is either Net/Rom or AX25. The hardware\_address is a callsign and the subnetwork mode is DG or VC ( only has significance for level 2 links ).

The syntax of the command is :

```
ARP [ destination [ + [ P ] protocol callsign [ mode ] ] ]
```

or

```
ARP [ destination [ - protocol ] ]
```

In the first form an entry is made in the table, in the second an entry is deleted. This is only permitted for sysop or manager.

The parameters are :

destination	An address of the form nnn.nnn.nnn.nnn
P	If present, marks the entry as 'published'
protocol	AX25 or Net/Rom
callsign	A valid amateur callsign, e.g. G8KBB-5
mode	DG or VC

If 'P' is entered, then the node will publish the address. Specifically, if an ARP request is seen by the node for a station with the address given, it will send a response advising the caller of the callsign to be used.

More details on the operation of the router are contained in section 7.

The ARP table also shows a further parameter of TIMER. This is only shown when it has a value other than 0. When non zero, it is a timer showing the number of minutes before the ARP entry is automatically deleted. Entries made by means of the ARP command have timer values of zero and therefore persist. Entries made from the network by ARP protocol frames have the timer entries set to the ARPtimer initial default.

### 3.29 UI

The UI command allows a string to be sent as a Level 2 UI frame. The syntax is

```
UI dest string_of_text_ending_in_return
```

Dest is a callsign like destination such as 'MAIL'. What will happen is that a single UI frame will be sent with a source callsign of the user who entered the command, a destination callsign of dest, and the rest of the string as text.

It is designed to be used in situations where a local BBS does not have access to a common channel and wishes to send mail notification packets. Not surprisingly, the ability to do this is BBS specific.

---

### 3.30 MTU command

The MTU command is used to configure the node's Maximum Transmission Unit figures ( primarily for TCP/IP support ). In general, they should be left at the default values. Do not experiment unless you are sure you understand the significance of what you are doing !!!!

The syntax of the command is identical to the syntax of the PARMs and MODE commands, as defined in section 3.32.

There are 5 values configured by the command. These are :

Number	Default	Function
1	256	Sets the MTU for the IP router, AX.25 Level 2, Radio port ( port 0 )
2	256	Sets the MTU for the IP router, AX.25 Level 2, RS232 port ( port 1 )
3	236	Sets the MTU for the IP router Net/Rom interface
4	257	Sets the maximum number of data bytes permitted in an AX.25 level 2 frame before an error response is returned to the sender ( frame too long )
5	328	Sets the maximum number of bytes permitted in a level 2 packet. Above this, frames are discarded. If a packet may contain 256 data bytes, and a maximum length address of sender, recipient and 8 digis comprises 70 bytes, and 2 bytes are used for control bits, the total ( 256+70+2 ) is the setting of this parameter.

This command replaces the ROM patching needed for TheNet X-1H.

The minimum that an MTU may be set to is 64, the maximum is 1024, but large packets increase the probability of crashing the node. Beware !!!!!.

The MTU for the Net/Rom port should not in general be set higher than 236 or it will not be compatible with Net/Rom.

The limits on the other two correspond to those necessary to support frames in the range 256 - 1024 data bytes long.

### 3.31 METER command

This command's syntax is similar to the PARMs command, and includes the new syntax as described in the overview guide, section 3.32.

It allows the following parameters to be controlled. Note that the parameter list differs from TheNet X-1J, where the first ( and only ) parameter was the deviation meter scaling factor.

- 1 The meter mode flags
- 2 The deviation meter scaling factor
- 3 The signal strength meter noise floor value
- 4 The S meter display format multiplier
- 5 The dBm meter display format multiplier
- 6 The dBm noise floor value
- 7 The voltmeter channel 1 multiplier
- 8 The voltmeter channel 2 multiplier
- 9 The voltmeter channel 1 offset value
- 10 The voltmeter channel 2 offset value
- 11 The voltmeter channel 3 multiplier
- 12 The voltmeter channel 4 multiplier
- 13 The voltmeter channel 3 offset value
- 14 The voltmeter channel 4 offset value

---

### 3.31.1 The meter mode flags

Each bit of this parameter controls a different aspect of the meters as shown below

BIT    If set, then ....

0	The deviation meter is enabled
1	The signal strength meter is enabled
2	The signal strength is shown as S points rather than dBm
3	ADC channel 3 is enabled ( voltmeter channel 1 )
4	ADC channel 4 is enabled ( voltmeter channel 2 )
5	Voltmeter channel 1 divisor is 1000 rather than 100
6	Voltmeter channel 2 divisor is 1000 rather than 100
7	Voltmeter channel 1 displays fine resolution rather than integers
8	Voltmeter channel 2 displays fine resolution rather than integers
9	ADC channel 1 is enabled ( voltmeter channel 3 )
10	ADC channel 2 is enabled ( voltmeter channel 4 )
11	Voltmeter channel 3 divisor is 1000 rather than 100
12	Voltmeter channel 4 divisor is 1000 rather than 100
13	Voltmeter channel 3 displays fine resolution rather than integers
14	Voltmeter channel 4 displays fine resolution rather than integers

### 3.31.2 The deviation meter scaling factor

This is the parameter previously described in section 3.31 of the overview guide. It scales the deviation meter display. One change however from X-1J. Setting it to zero does not disable the deviation meter - bit 0 of the meter mode word as described above controls whether it is enabled.

When set to a value in the range 1 - 255, the meter is enabled and the value is used as a scaling factor. The ADC is an 8 bit device, so it will give a response in the range 0 - 255, corresponding to an ADC input voltage in the range 0 - 3 volts DC. If optimally configured, this corresponds to the maximum audio level possible for the given receiver discriminator.

The ADC reading ( 0 - 255 ) is multiplied by the meter parameter value ( 1 - 255 ) to give an answer in the range 0 to 65 KHz ( approx. ). This is the value displayed in the mheard list.

Hence, if, for example, a DC voltage of 2 volts at the input to the ADC corresponds to 3.4 KHz deviation, the ADC reading will be 171 ( + a few ) and the Meter parameter will need setting to 20 ( ie to  $3400 / 171$  ).

If the ADC reading is 254 or higher, then in order to indicate an overrange, the symbol '>' will precede the corresponding deviation entry in the heard list.

### 3.31.3 The signal strength meter noise floor value

This parameter sets the 'no signal' offset applied to input readings from the signal strength meter. It is subtracted from the count read from the ADC to give a reading based on a no signal value of zero. If the no signal ( noise ) reading is, for example 0.65V, corresponding to an ADC count of  $256 * 0.65 / 3$  or 54, then 54 is subtracted from each ADC signal strength reading to give an integer from 0 to 201 for an input reading between 54 and 255.

### 3.31.4 The S meter display format multiplier

This parameter operates in a similar manner to the dBm multiplier ( section 3.31.5 ) but having divided the intermediate result by 256, that value is displayed as an integer in the range 0 to 9 preceded by the letter 'S'. If the value exceeds 9, it is displayed as 'S9+'. If the dBm multiplier has been set up correctly, then set this parameter to the dBm multiplier divided by the number of dB per S point.

#### 3.31.5 The dBm meter display format multiplier

This parameter is used to convert an ADC reading for the signal strength meter into a dB count. The ADC reading is converted into a count from 0 to n, according to the description contained in section 3.31.3. It is then multiplied by this parameter and divided by 256. When added to the dBm noise floor value ( section 3.31.6 ), this gives the displayed dBm value in the heard list.

#### 3.31.6 The dBm noise floor value

This is the noise floor ( dBm ) reading that corresponds to the zero count in section 3.31.3. It is entered as a positive count corresponding to a negative value. For example, if the zero point in the example of section 3.31.3 is -113 dBm, the noise floor value entered is 113.

#### 3.31.7 The voltmeter channel 1 multiplier

This is the multiplier that controls voltmeter channel 1 ( ADC channel 3 ). It is set as described in section 3.34 below.

#### 3.31.8 The voltmeter channel 2 multiplier

This is the multiplier that controls voltmeter channel 2 ( ADC channel 4 ). It is set as described in section 3.34 below.

#### 3.31.9 The voltmeter channel 1 offset value

This is the value subtracted from the ADC reading before it is multiplied by the multiplier parameter. It is described more fully in section 3.34 below.

#### 3.31.10 The voltmeter channel 2 offset value

This is the value subtracted from the ADC reading before it is multiplied by the multiplier parameter. It is described more fully in section 3.34 below.

#### 3.31.11 The voltmeter channel 3 multiplier

This is the multiplier that controls voltmeter channel 3 ( ADC channel 1 ). It is set as described in section 3.34 below.

#### 3.31.12 The voltmeter channel 4 multiplier

This is the multiplier that controls voltmeter channel 4 ( ADC channel 2 ). It is set as described in section 3.34 below.

#### 3.31.13 The voltmeter channel 3 offset value

This is the value subtracted from the ADC reading before it is multiplied by the multiplier parameter. It is described more fully in section 3.34 below.

#### 3.31.14 The voltmeter channel 4 offset value

---

This is the value subtracted from the ADC reading before it is multiplied by the multiplier parameter. It is described more fully in section 3.34 below.

### 3.32 PARM, MODE, MTU, METER & IPSTATS command syntax

The syntax of these commands has changed.

All use the same syntax, which may be either of two types, the original TheNet 1.01 syntax ( as used in versions previous to X-1J ) or an 'offset & value' type.

The original syntax was, by way of example,

```
PARM { [ * | new_value ] [ * | new_value ] ..... }
```

so to set the 10th PARM ( the L4 retries ) to 1, the syntax would be :

```
PARM * * * * * * * * * 1
```

The equivalent new syntax command would be :

```
PARM / 10 1
```

The '/' command signifies that what follows is the parameter number followed by the new value. As for the old command syntax, the complete list of parameters is displayed. Setting the parameters may only be done by a Sysop. Note that BOTH command syntaxes are supported - you can use whichever you prefer.

### 3.33 BTEXT, INFO and CTEXT Command Syntax

In Version X-1J, the syntax of these commands changed. In addition, the Info message was doubled in size to 160 bytes.

If someone who is not Sysop uses the command, the current settings are displayed.

If a Sysop uses it without any additional parameters, the current setting is displayed.

If a Sysop enters one of the commands followed by a parameter of '\*', the current message is deleted.

If a Sysop enters a string of text, that text is added to the current message, followed by a newline.

It is therefore possible to build up multiple line messages. If you wish to start a message with a blank line, enter a message with a non display ( or innocuous display ) character such as control-A. It will get entered followed by a newline. On most systems this will not display. On some systems such as PCs running NOS, it will display as a smiley face.

### 3.34 The ADC command

This command is used to read the voltages on the ADC channels 3 and 4 ( referred to as the 'voltmeter channels 1 and 2' in the documentation. It also displays the ADC channels 1 and 2 ( referred to as the 'voltmeter channels 3 and 4' in the documentation ) if these optional channels are enabled.

ADC channels 1 and 2 are normally used for the deviation meter and S meter respectively. If one or other of these functions is not wanted, then the corresponding ADC channels may be used as general purpose inputs just like channels 3 and 4. In this case, the METER Mode Flags are used to enable the channels, the ADC3 and ADC4 commands are used to set the textual descriptions and the ADC command is used to read the channels. If the S meter or deviation meter are enabled and the channels are also in use as ADC command channels then the ADC command readings will read true but the

deviation meter or S meter may occasionally read incorrectly if the ADC command is issued by a user and actioned at the same time as the software is reading on a received packet.

. A typical display is shown below :

```
IPNET:G8KBB-5}
13.2 V DC
-5 deg C
```

Each channel is associated with a scaling factor in the METER command, an offset value, two control flags and each meter may be enabled or disabled by setting or clearing the appropriate bit in the meter control flags word ( see section 3.31 ). Finally, each has a textual string that may be appended to the reading, as shown in the above examples.

The displayed voltage is shown as

$$( \text{meter\_reading} - \text{offset\_value} ) * \text{Scaling\_factor} / n$$

where n is 100 or 1000 depending on the setting of the resolution bit ( see section 3.31.1 ). The calculation is done with 16 bit signed registers, so it is important that the intermediate result of

$$( \text{meter\_reading} - \text{offset\_value} ) * \text{Scaling\_factor}$$

lies in the range -32768 to + 32767, hence the choice of n.

It is shown with a resolution of either 1 or 0.1 depending on the resolution flag, as described in section 3.31.1.

By way of example, with a scaling factor of 71, a resolution of 'fine', an offset of 0 and a divisor of 1000 will give a display ranging from 0 to 18.1V in steps of 100 mV for an input in the range 0 to 3V DC input to the ADC. By setting the scaling factor and divisor accordingly, it is possible to create a meter with full scale readings up to 320V. For the 18 V fsd meter, a resistor from ground to the ADC input of 2.2K with a resistor from the ADC input to the voltage to be measured of 11K will give the correct display with a scaling factor of 71 and an ADC reference of 3V ( with a slight error that can be corrected by tweaking the ADC reference ).

The facility may be used to create a temperature meter by connecting a suitable sensor to the ADC input. This will usually give a value that ranges over a portion of the input range for a range of temperatures. The reason for the offset value and use of signed arithmetic is to allow the input reading to be converted into a direct display of temperature. Full details are contained in the file 'temp.doc', but by way of example, if the input probe gives a range of 0 to 2.6 V for a temperature display of -40 to +90 degrees Celcius, the settings would be

```
offset = 68
scaling factor = 59
divisor = 100 ( control bit cleared )
resolution = coarse ( control bit cleared )
```

and for a corresponding Farenheit display of -40 to +194, the settings would be

```
offset = 38
scaling factor = 106
divisor = 100 ( control bit cleared )
resolution = coarse ( control bit cleared )
```

Note that there is little point using the fine resolution mode for these displays.

The ADC channels are read whenever the ADC command is issued, interrupts being disabled for 100 microseconds for each reading on a 4.9 MHz TNC2. The ADC needs 40 microseconds to convert the data, and the 100 microseconds allows for a 10 MHz TNC2 to be used.

### 3.35 ADC1, ADC2, ADC3 and ADC4 commands

These commands are used to set a textual string that follows the ADC readings. The format and parsing is the same as for the INFO and similar commands ( see section 3.31 ), but the length is only 8 characters. The same routines were used to save space but this is not really an ideal way to do it. To set a string, the previous must be cleared with the '\*' option, and a new string entered. If the string is less than 12 characters long it will be followed by a newline. These strings are not preset in the EPROM and will need to be entered after a coldstart. All other meter parameters may have ROM defaults.

### 3.36 L3MHEARD

The L3 Heard list records the identity of nodes that use this node for gatewaying their Net/Rom frames. Every time that the node receives a Net/Rom frame from another node it is either destined for the node or it is for gatewaying to another node. When a frame is received for gatewaying, its passage is recorded in the L3 Heard list. It functions and is managed in a manner similar to the main heard list, with the callsign of the sending node, time since it was last heard, total number of frames gatewayed and level 2 port number from which it was received being recorded. In addition, each entry records the callsign of the last destination node. Each Net/Rom frame contains an originating node callsign and a destination node callsign. The entry in the table is associated with the originating node callsign and the final entry on each line shows the destination node of the last frame gatewayed from each origination node.

The L3 heard list is managed in a manner identical to the main heard list. It has a length set by the sysop and will drop old entries from the end of the list when the list is full or when a node hasn't gatewayed through the node for more than 12 hours.

The initial size of the L3 heard list is the same as the size of the main heard list with both set to the ROM default size.

### 3.37 BRATE

This command only works with the PK96.

The PK96 hardware does not have the TNC2 hardware switched baud rates. Instead, it has software selected baud rates and a 9600 baud modem as well as the 1200 baud modem. The node software supports this by the inclusion in the ROM default values of initial baud rates and by a sysop only switch command to allow those to be changed during operation.

Each baud rate may be set in the range 300 to 19200 baud to one of 7 standard rates. This does not imply the code will reliably handle data at 19200 but it seemed fun to include it.

The rom defaults may be overridden by the BRATE command. The syntax of this is similar to the PARMS, MODE etc. commands, and has two parameters. The first is the radio port rate, the second the RS232 rate. Values are in the range 0 to 6 and have the following meanings :

Setting	Baud Rate
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	19200

When the BRATE command is used to change a setting, it DOES NOT take immediate effect. It will take effect after a warmstart. Coldstart will, of course, restore the ROM defaults. A warmstart may be effected by the RESET command ( as sysop ) or by powering off and on again the PK96.

The radio baud rate setting also controls the modem selection. If the selected baud rate is 1200, then the 1200 baud modem is selected. If any other rate is selected, the 9600 baud modem is selected, but with the baud rate requested.

When a CWID is sent ( if enabled ), the code will reselect the 1200 baud modem regardless of which modem was in use in order to send a valid ID in tone. It also keys the tone on and off rather than switching between tones as the TNC2 version does. If the 9600 baud modem was in use, the 9600 baud modem will be reselected after the CWID.

### 3.38 ARPTIMER

The ARP timer is managed as for other parameter based commands such as PARMS, MODE etc. and is used to control the ARP protocol machine.

The ARP table has, until now, been managed manually. With this version, the node will automatically update its ARP table by issuing ARP requests and by interpreting ARP requests and ARP replies.

The ARPtimer command has two parameters. The first enables (1) or disables (0) the ARP service routine. The second is the ARP timer in minutes. When the node receives an ARP request or reply, it will if appropriate make an entry in the table and set the timer to the ARPtimer value. It will then decrement once a minute. When it reaches 0, the ARP entry is deleted. The ARP timer may be set to any value between 15 and 1440 ( i.e. quarter of an hour and 24 hours ).

ARP requests/replies DO NOT apply to Net/Rom encapsulation. Entries will only be made automatically for AX.25 connections. For Net/Rom encapsulation, make the entry manually.

If the ARP function is disabled, the system does two things. Firstly, it will stop processing received ARP frames. Secondly, it will stop decrementing the ARP table entry timers thereby freezing the ARP table entries. As a warning to any attentive sysop, it will still generate ARP requests if it lacks a table entry but will ignore any response.

Note that any entries in the ARP table made by the SYSOP using the ARP command will not be overwritten by received frames and will not age.

One final difference between this code and NOS. When NOS receives a frame for transmission for which it needs to request an ARP entry, it will queue the frame pending address resolution. It will also send an ICMP source quench if further frames are received whilst the request is pending. This code

will not. The IP frame that causes the ARP request to be sent will be lost. So, if you PING it and it sends an ARP request you will get a response to the PING the next time you ping it.

#### 4. Other Changes

This section covers the other miscellaneous changes to the software.

##### 4.1 Command Processor

The command processor has been altered. In general, but not in all cases, commands only appear on the 'help' menu when they are enabled, so for example the 'BBS' command will not be shown unless it has been enabled with the 'BBS +' command. The exception is the sysop commands, like MODE, LINKS and PARMS, which are never shown to users but are of interest to them. If the appropriate bit is set however in the MODE command ( see 3.5.11 ), then for the sysop or manager, all commands appear in the help prompt - EVEN IF DISABLED.

The help screen now shows commands in a combination of upper and lower case characters.

Some commands however are never displayed in the command lists, even when enabled. An example of this is the PARMS command. This is controlled by a flag associated with each command in the same way that the command enable / disable bit is associated with each command.

An addition to the command syntax controls this. A command is displayed to a user only when it is enabled and its display bit is set. The display bit is set by using the '+' command with the letter 'D' following the '+'. Conversely it is cleared by using the '-' option followed by 'D'. Hence to display the PARMS command in the user command list, the sysop enters

```
parms + d
```

or to hide the STATS command whilst still leaving it open to users to use, the sysop enters

```
stats - D
```

The 'D' may be in upper or lower case.  
The full syntax of the command modifier is therefore

```
command [ - | + [ D ] ]
```

If no letter follows the '+' or '-', or if it is not 'D', then the command enable / disable flag is modified. The letter 'D' may be in upper or lower case.

It is common practice to create a configuration file that can be uploaded to a node following reset. It is also helpful to have comments in such files. In order to allow this, the node will ignore any command given to the switch where the first character on the line is '#'.

##### 4.2 Beacon digi

It is possible to set a digi in the address used for beacon packets. Details of how to do this are contained in the configuration guide. Note that this is provided for those rare occasions when there is a genuine need. This is rarely the case and should not be done unless really necessary.

##### 4.3 Nodes Broadcasts after power-up

The node will now broadcast its node table 60 seconds after power-up. This is to ensure that the network is back to an operational state as soon as possible following a node reset. The reason for the short delay is to cater for the situation where the Sysop switches on the node before the radio.

#### 4.4 TexNet interface handling

The TexNet interface is controlled by bit 7 of the user help messages word ( MODE parameter number 12 ). If set to a value between 128 and 255, the TexNet interface is enabled. With values between 0 and 127 it is disabled.

When disabled, operation is as normal.

When enabled, the node will react to the TexNet "\*\*\* LINKED to.." messages. When a level 2 connection to the node is made, the node will examine the first line of text received by the node. It must be a line of text terminated by a return character, and it must be the first line of text received. If any other line is received first, the node will issue an error message.

The line is checked to see if it starts with the text

\*\*\* LINKED to

It must start in the first character of the line and match exactly the string shown. If it does, the text that follows the string is scanned for a valid callsign. If one is found, the user call is remapped so that any subsequent actions of that user are attributed to that new callsign. Any text that follows the callsign ( including any TexNet SSID ) is ignored.

So, for example, if the string

\*\*\* LINKED to WB4DDP E something else

is received by a node ANODE:G8KBB from a station G9BF, then the session, otherwise attributable to G9BF will be remapped to WB4DDP. The 'E' that followed was an SSID of 14, which is ignored.

The heard list will show the following

ANODE:G8KBB } TheNet X-1J4 (644)  
TexNet(G9BF WB4DDP)

Any actions by that user, such as conferencing or onwards connection to other nodes will use the callsign WB4DDP. If a remote disconnect causes reconnection to the node, the remapped callsign will continue to be used.

Note that only one station at a time may use a connection between a TexNet node and a TheNet node as it references a remapped level 2 connection. It is however possible to connect to the callsign or alias of the node and invoke the remapping function.

When the TexNet remapping is done, then the connect text message will be sent to the user if the connect message is set and enabled. If the node sees a packet with a TexNet PID, it will note that station as TexNet in the heard list.

#### 4.5 The USER command and CHOKE status

The USER command shows stations connected to the node. When a user connects on to another node or downlinks, an entry is formed showing the two links on either side of a "<->" or "<.>" symbol. In this version, the '-' characters will be replaced by another character if one or other link is

choked. The first '-' character is associated with the left hand connection ( circuit or uplink ). The second with the other circuit or downlink.

For a level 2 uplink or downlink, the "-" is replaced by 'C' if the connection is choked. For Level 4 connections, the "-" is replaced by one of 3 characters, "R" if it is choked from the other end, "L" if it is choked from this end or "C" if it is choked from both ends !.

#### 5. CWID keyer.

The CWID keyer sends the station callsign in CW by alternating between the two modem tones ( TNC2 ) or keying a single tone ( PK96 ). This is nominally sent at 20 wpm once every 30 minutes, but the speed and period can be changed remotely.

After a delay of 30 minutes, the callsign is sent appended to the end of the next data packet that is sent over the air. There is a 500 ms delay after the end of the data packet before the call is sent.

The program prefers to send CWIDs appended to ordinary data packets. However, if one minute after the CWID has supposed to be sent it is still pending because no data packets have been sent, it will key up the transmitter anyway. Persist, TxDelay and other parameters are honoured, but the process involves changing the SIO mode and this will have an annoying effect on any packets being received in full duplex mode.

#### 6. Version X-2.

X-1 was the first release of this code. The objective is to get some practical feedback and test the code before the full release, version X-2, which I hope will be very similar to this release ( X-1J2 ). I have been saying this for some time now, but things keep getting added.

Version X1A added the escape-N command and the change to the connect, nodes and reset commands. The timers were also added to the stats command.

Version X-1B removed all the escape commands apart from C, D and P. It also added the MODE command and extended the + and - command qualifiers to all commands.

Version X-1C added TALK, MANAGER and AUDIT. The SYSOP command was enhanced and the INFO command was altered to limit the length of a message ( a bug in the original version of TheNet ). The help screen was changed to display commands in a combination of upper and lower case.

Version X-1D extended the auditing and statistics to cover auditing everything but level 3, and statistics of the CPU, Level 1, Level 2 and timers.

Version X-1E added beacon timer control, the connect redirector, the nodes dump facility, level 3 & 4 statistics and the LINKS and CALIBRATE commands.

Version X-1F added the CLOSEDOWN, DXCLUSTER, ACL, CTEXT, HELP and BTEXT commands. Another parameter was added to the MODE command to control textual messages. The mod suggested by DF2AU to correct the DCD latchup was included. Additional statistics were added covering CRC errors, receiver overrun, transmitter underrun and framing errors.

Version X-1G added mainly the IP router, with the following commands to control it - IPRROUTE, ARP, IPSTATS, PADDRESS, IPBROADCAST. In addition, the ALIAS, BBSALIAS, HOSTALIAS and DXCALIAS commands crept in, as did QUIT as an alternative to BYE. The help messages extended to enable nodes in the routes list to appear as alias:callsign, and an extra byte on the MODE command allowed '#' nodes to be selectively NOT broadcast. The order of HELP and HOST commands changed

---

so that 'h' on its own gave help not host. The code was optimised with some time critical parts being recoded in assembler and a peephole optimiser being used for additional improvements.

Version X-1H fixed 3 bugs in X-1G.

Version X-1J added the deviation meter support with the Meter command and Mheard changes. In addition, parameters were added to the MODE command for slime trail control, control of digipeating and reconnection to node. The command syntax of Info, Btext and Ctext was changed to support multiple lines and the Info message space was doubled to 160 bytes. Nodes broadcasts now occur 60 seconds after power up and the ARP Digi bug fix was included. The level 4 minimum retries was dropped to 1 and the PARM, MODE, IPSTATS, METER and MTU command syntax was extended to support 'offset & value' type operation. An MTU command was added to allow IP MTU limits to be changed under software control. The node alias case sensitivity bit and TALK 8 bit data bits were added.

Version X-1J Release 2 fixed a couple of bugs, added the S meter function, voltage reading functions, TexNet handling and extended the command enable / disable switches with the 'command - D' syntax.

Version X-1J release 3 was an interim version for testing. If you have it trade up to release 4 !

Version X-1J Release 4 fixed the buffer loss problem and implemented the automatic update of the ARP table by ARP REQUEST and ARP REPLY. The node also maintains a second heard list of L3 Netrom nodes the gateway through the node, and a bug in the copying of digipeated frames in KISS mode 2 has been corrected. A defensive port flush function has been added and link list integrity checking has been extended to check reverse links. A bug in handling of BBSALIAS, HOSTALIAS and DXCALIAS has been fixed as has the clearing of BBS, HOST and DXCLUSTER on cold start. The USER list shows circuit choke status for patchcord connections.

If you read this and say 'Pah. it doesn't do XXXXX' or 'It still doesn't do YYYYYY' or anything of a similar nature, don't keep it to yourself. Tell me. I may well do it. An example of this are the many changes introduced into X1-J as a result of suggestions mainly by KA2DEW .

## 7. The IP router

The IP router co-exists in the node with the other software. It is connected to the L2 and L3(Net/Rom) protocol machines, and is managed from the L7 switch. It will accept data from L2 Datagrams, L2 Virtual Circuits or NOS protocol extended Net/Rom frames. It will output to these 3 depending on the setting of the IProute and ARP tables.

The router supports the IP options of NOS and also does IP fragmentation. Level 2 segmentation is not supported. In addition, ICMP is implemented in so far as it is needed to respond to errors or PINGs. No higher layer support is provided, i.e. TCP is not implemented, ip\_send() and ip\_receive() are only implemented in so far as they are needed for ICMP. You can therefore PING it but anything else will solicit an ICMP error message.

The default MTU is 256 for AX.25 and 236 for Net/Rom. It will accept longer datagrams than this and fragment the output but it is not recommended as it merely wastes RAM. The MTU command may be used to change this.

It is possible to be creative in the use of L2 datagram and virtual circuits by use of the port default settings and the ARP table. The algorithm used is :

When a frame is to be sent, the ARP table is scanned for the appropriate entry. The entry tells it what callign to use. For Net/Rom encapsulation, it is sent to the Net/Rom protocol handler. For AX.25 encapsulation the following applies. The ARP table may indicate DG or

VC. In this case, that mode is taken. If there is no DG or VC entry, the TOS bits are examined. If the delay bit is set, a datagram mode is selected. If not, and the reliability bit is set a virtual circuit is selected. If neither bit is set, the default mode for that port is used to select a mode ( see IPstats command, first parameter ).

Port addressing is not supported at the moment. When a node stack is being used, it is possible to set all nodes or a subset of them to the same IP address. For more information, see the file 'IPXLINK.DOC'.

The IP router is manually controlled - no rspf or rip. The node will manage its ARP table automatically now. See the section on the ARPTimer for more details.

## 8. MISC

Anyone interested in a copy of the program, drop me a message on GB7MXM.#36.GBR.EU Also, any suggestions for change gratefully received.

## 9. The AEA PK96

This section describes the implementation of TheNet X-1J release 4 for the PK96. The PK96 version will not work in a TNC2 and vica versa.

It is important to read this section as it describes the differences from the TNC2 version and how to install it in the PK96. The main documentation set reflects the TNC2 version !.

In summary :

- The PK96 version does not have the ADC support and associated commands such as METER, ADC, ADC1 and ADC2. It does not include meter displays in the heard list.
- A command has been added to set the radio and RS232 port baud rates ( BRATE )
- The software supports the hardware of the PK96 ( sic )
- The RS232 DCDB ( pin 23 ) signal to control the crosslink protocol selection has been implemented using a hardware change to the PCB unless the PCB of the PK96 is version C or later.
- The MAIL LED gives a health check signal by flashing once a second
- The STA LED flashes whenever the node routes ( as opposed to sources or sinks ) a Net/Rom L3 or IP frame
- The rear panel push switch may be used to perform a coldstart

### 9.1. Rear panel push switch

Pressing the rear panel switch will perform a coldstart. This may be done at any time other than straight after a reset when interrupts are ignored. The easy way to perform a warmstart is to switch the power off and on again. Be careful not to accidentally push the reset switch when the power is on !

### 9.2. ROM differences

All traces of the ADC have been removed. This also extends to the ROM parameters, and the default baud rates are stored in address offsets 0x151 and 0x152 in THENET1.X1J respectively.

Please be sure to use the new patcher and the correct version of SETHELP. Using the PK96 commands on the TNC2 files or vica versa will corrupt the code.

---

Also, note that bank switching is different with the PK96. No need for wires, just plug the EPROM in. The EPROM is the same 27C512 as is needed for the TNC2 version and it needs both files THENET1.X1J and THENET2.X1J in just the same way. Only THENET1.X1J needs to be patched with the ROM defaults however, and THENET2.X1J is 4K shorter as a consequence. The HELP text in THENET2.X1J may not however be 4 K longer. It may only extend to 0x6fff.

The patcher therefore only modifies THENET1.X1J.

### 9.3. CROSSLINK selection

The PK96 prior to version C of the PCB does not support the pin 23 RS232 mode selection signal for crosslink or dumb terminal mode. If you have version C or later, insert JP5 to connect the signal to RS232 pin 23.

**IMPORTANT - the signal goes directly to a CMOS gate input, so it must either be left open circuit ( for terminal mode ) or connected to GROUND ( for crosslink or KISS protocol mode ). Do not connect it to any other voltage.**

The easy way to ground it is to connect JP4 in the TNC which puts ground on pin 10. Connect pin 10 to pin 23 for crosslink, and remove the connection, JP4 or JP5 for terminal mode.

If you have version B of the PCB it can be modified with a bit of care. To see if it is version B, take the cover off and look at the type number next to the power on / off switch. It should read something like

013-135 B

The 'B' signifies revision B. If you want to modify it, proceed as follows.

**IMPORTANT - I suspect that your supplier will take a dim view of this with respect to the warranty. Your decision.**

1. Find yourself an anti-static work area and use it. You will be soldering to CMOS inputs.
2. Switch off. Remove power cord and other cables. Remove the case by removing the four case screws (two down each side), the RS232 connector mounting pillars on the back of the unit and the fixing ring for the audio connector jack socket, also on the rear.
3. Remove the battery link connector ( next to the battery ) and subsequently take care not to short circuit the battery.
4. Remove the 4 PCB mounting screws ( two at the front on each side, one near the DIN connector behind the reset switch, one at the site of the 5V regulator behind the fuse ).
5. Slide the PCB backwards to clear the LEDs and power switch from the front panel and lift it clear.
6. Locate pin 52 of the CPU. The CPU is the 64 pin device ( 0.07 inch pin spacing ) located between the 40 pin 8530 chip, the ROM and the RAM. It is denoted U1 on the PCB silk screen. This pin is connected to ground by tracks on both the top and the bottom of the board. Use desoldering braid or a desoldering tool to completely desolder it and carefully bend it up and out of the hole so that it can have a wire soldered to it. If you cannot completely desolder it, get a very fine screwdriver behind the pin on the top of the board and gently bend it out whilst applying minimum heat to the pin with a soldering iron from below. Do it carefully and the pin will come out without damaging the PCB tracks.
7. Locate pin 12 of U15. This is a 74HCT04 or similar located between the CPU, the 8530 and the modem disconnect header. Connect a short fine wire on the top of the board from pin 12 of U15 to pin 52 ( the one you bent out ) of the CPU.
8. Turn the board over. There is a short piece of track between pins 13 and 14 of U15. Cut it with a scalpel.

9. Connect a 10K resistor between pins 13 and 14. Keep it close to the board as you will need to replace the board in the case soon.
10. Locate pin 23 of the RS232 connector on the bottom of the board. ( remember that RS232 connectors number their pins along each row, so pin 23 will be third from the end nearest the power connector of the shorter row of PCB pins ). Connect a short piece of wire on the bottom of the board from RS232 pin 23 to pin 13 of U15.
11. Remove the EPROM and replace it with the TheNet EPROM.
12. Connect a short piece of wire from pin 10 of the RS232 connector to ground. A convenient point to use is one of the connector mounting points where the metal parts of the connector body are soldered to ground.
13. Replace the PCB in the case and refit the four mounting screws. Check that there are no shorts from the wiring to the case.
14. Connect a voltmeter to pin 52 of the CPU with the negative lead to ground. Apply power the TNC and switch on. All LEDs apart from the XMT LED should light and extinguish again if the node is working. The MAIL LED will then start to flash once per 2 seconds. The voltmeter should read low ( i.e. about 0 Volts ). Link pin 10 to pin 23 of the RS232 connector. The voltmeter should read 5V ( approx. ). If this does not work correctly, switch off and recheck it.
15. Switch off and remove the power lead.
16. Replace the battery link connector.
17. Check the PCB mounting screws are tight and replace the case. Replace the RS232 mounting posts, audio connector ring and all 4 case screws.
18. Reconnect the cables, switch on, and check it all works.
19. Go get a beer, wine, cup of tea, coffee or whatever else tickles your fancy.

#### 9.4. Crosslink connections and other RS232 matters

The RS232 connections differ to those of a TNC2. The RTS input controls whether the node can send data on the RS232 interface, and as there is no input on pin 20, it must be used for the crosslink handshake. Unfortunately, the crosslink handshake signals work the 'wrong way up' for the input, so it will be necessary to invert the signal into pin 4. This only applies if you have a diode matrix or similar active coupler. To crosslink 2 nodes just ignore the handshake signals.

Finally, pin 4 will stop the node sending data to the RS232 port if it is grounded. It can be grounded by switching off a computer that drives this input. The effect is that data would be backed up waiting to be sent indefinitely. The node would eventually run out of buffers. To prevent this, a modification has been done to the code. Every ten minutes, the node checks to see if there was data to be sent and whether any was sent. If there was, and no data was sent, and if this also happened in the previous 10 minute period, it will purge the send list of all data waiting to be sent. It will also increment the 'port flush' counter.

Just in case DCD gets stuck on ( this is not likely for the PK96 but can occur with some TNC2 clones and the code is also implemented in the TNC2 version ) the same process will apply to the radio port.

The 'port flush' counters have been added to the STATS, at the end of the list. Two numbers are shown, the first being the counter for the radio port, the second for the RS232 port. Under normal operations, neither should increment. If a port gets stuck, then the counter for that port will increment every ten minutes for which the port has data to send and can't send it.

Finally, the 64180 CPU in the PK96 does not have the capability to interrupt the CPU when the CTS line changes state. The code normally uses this to implement the crosslink CSMA protocol where a number of nodes share a common diode matrix connection. Instead the code polls the line status every 10 milliseconds. This means that it will not be very aggressive in seizing the channel and other nodes will tend to get a bigger share of the channel. How well this will perform is not known - so if it gives

problems let me know and I'll increase the polling speed - at the expense of processing power. A better alternative is to use one of the many active crosslinks that poll nodes in sequence.

Dave G8KBB

USER GUIDE FOR TheNet X-1J release 4

This brief note is intended for users of TheNet X-1J, and explains the basic commands. Configuration and sysop features are not covered fully.

TheNet X-1J is an extension of TheNet 1, and provides a number of new features.

The switch provides the following user commands :

- Connect
- Info
- Nodes
- Routes
- Users
- Talk
- CQ
- BBS
- Host
- MHeard
- Bye
- DXcluster
- IProute
- ARP
- QUIT
- ADC

Not all commands may be available on every node as certain commands might have been disabled. If a command has been enabled, it will be displayed when you type an invalid command such as '?'. In addition, there are some commands that are available but are not usually displayed. The main ones of interest are :

- Links
- Mode
- Parms
- Stats
- IPAddress
- DXCAlias
- BBSAlias
- HostAlias
- MTU
- L3MHeard

The Sysop may have customised the command list according to local needs.

In general, commands take parameters, for example to connect to GB7MXM, the command to be entered is CONNECT GB7MXM followed by return. Usually, commands are not case sensitive, ( although node aliases may be ) and commands may be abbreviated.

#### Connect

If the connect command is given on its own, then assuming that the sysop has set it up correctly, you will get connected to the local BBS.

If you give another callsign, either of a local station or a node, the node will attempt to connect you to that station either by a level 4 connection or by downlinking. If you are downlinking, you may also specify digipeaters.

In either case, you get either a connected message or a message telling you of the failure to connect. If you enter any other command at this stage, the connection attempt will be aborted.

If you attempt to downlink with digipeating, or attempt to downlink to an invalid callsign such as a node alias with an SSID, you may get an error message depending on how the Sysop has configured the node.

Finally, if you connect to the node, then connect to another station, and that station disconnects you ( e.g. connect to a BBS and use the 'Bye' command of the BBS ), you will either get reconnected to the node or disconnected completely depending on the configuration of the node.

If you enter a node alias and get an 'invalid call' message, it can either be because the node is not recognised or it can be because the Sysop has made node aliases case sensitive.

#### Info

This command gives information about the node as a combination of a message stored in the EPROM and a message entered by the Sysop.

#### Nodes

This command gives information about the distant nodes that this node thinks it can get to. With no parameter, it shows the alias and callsign of all the nodes except those starting with a '#' character. If a parameter of '\*' is given, those 'hidden' nodes will also be shown.

If a callsign or alias is given that the node does not know, it gives an error message. If the callsign or alias of a known node is given, the node gives details of the routes it knows about that lead to that destination. The display shows one option per line, each of which consists of the path quality, obsolescence count and port followed by the callsign of the neighbour. If any route is in use, a chevron is shown against the appropriate entry.

If so configured by the Sysop, 'Slime trails' i.e. nodes without aliases that have not been the subject of a valid node broadcast, may be omitted from the nodes list.

#### Routes

This command gives information about the neighbouring nodes that can be heard. For each neighbour, the display shows the port number, the callsign, the path quality and the number of nodes accessible through this neighbour. If a route has been 'locked' by the sysop, then a '!' character is shown after an entry. The sysop may have configured the node to display nodes as callsign or as alias:callsign. If so configured, then if a node is shown as a callsign alone it means that it is not currently reachable as its node broadcasts are not being received.

#### Users

This shows who is using the node. It does not show other nodes that are using the node as a level 3 relay, nor does it show those users who have connected to the node but otherwise have done nothing.

The display shows the through connections, followed by those users who are connected to the switch and 'idle'. It also shows those users who are connected to the conferencing facility. The latter stations are shown connected to a destination called 'Talk', whilst in the case of connections, the two endpoints are shown. For connections, two symbols are used, '<->' and '<.>'. The former is used for established connections whilst the latter is used for connections being established.

---

When a through connection is established, the "<->" symbol may be modified if one or other circuit is choked. In this version, the '-' characters will be replaced by another character if one or other link is choked. The first '-' character is associated with the left hand connection ( circuit or uplink ). The second with the other circuit or downlink. A link is 'choked' if one end or the other has signalled that it cannot at the moment accept more data and wishes the sender to pause. It is designed to be used by the sysop in locating network congestion problems.

For a level 2 uplink or downlink, the "-" is replaced by 'C' if the connection is choked. For Level 4 connections, the "-" is replaced by one of 3 characters, "R" if it is choked from the other end, "L" if it is choked from this end or "C" if it is choked from both ends !.

## Talk

The Talk command allows a group of users to hold a conference call. It also allows a user to send a message to another user of the node provided that user is connected to the switch but is not patched through to another station and is not currently trying to connect to another station.

A user enters the conference by giving the command 'talk'. He/she gets a message informing them of this and reminding them that the command to escape from the talk command is '/exit'. Any other users currently in the conference get a message from the node telling them of the callsign of the user who has joined them. At this point, every line sent by a user in the conference is copied to all other users in the conference, preceded by their callsign.

To exit from the conference, the command '/exit' is used. This causes a response message to be sent to the user, and at the same time all of those left in the conference get a message from the node telling them of the station who has left the conference. If you force a disconnect, the other stations are not told of your departure.

A string of text may be entered on the same line as the talk command when the command is given. If this is done, before the user is connected to the conference, that string of text is sent to all the other users of the node who appear in the 'user' list but are not connected to anything else. For example if GxABC were to type :

```
TALK GyXYZ, Hello fred can I have a chat - type TALK
```

then other users of the node ( including presumably Fred, would get the message :

```
GxABC>> GyXYZ. Hello fred can I have a chat - type TALK
```

on their screens. The only exception to this is that sysops are not sent the message.

Note that the TALK command will pass 8 bit data if set to by the Sysop.

## CQ

This command is used to broadcast a CQ message. In addition, the fact that you are calling CQ is indicated in the USER list. The callsign will be your own with a different SSID, and anyone else can connect to you by connecting to the callsign with the appropriate SSID.

The CQ remains 'primed' for a while, and if any other command is given to the node the CQ will be cancelled.

## BBS

When you issue the BBS command, assuming that the sysop has configured it, you will be connected to the local BBS.

If you enter the command 'BBS ?', then the current setting of the BBS will be displayed.

#### Host

The HOST command operates just like the BBS command. It may have been disabled by the sysop, it may have been set to connect to the same station as the BBS, or it may have been set to connect to another host system.

If you enter the command 'HOST ?', then the current setting of the HOST will be displayed.

#### MHeard

If enabled, the heard list shows the last few stations heard. The number of entries is limited and set by the sysop so any stations not heard for a while may get pushed out of the list by others heard. Assuming that a station is not pushed out in this manner, the display shows the number of packets heard from that station since it appeared in the list and the time since it was last heard. The time is hours, minutes and seconds. The list also shows the port on which the station was heard ( port 0 is the radio port ), and if it hears IP frames or Net/Rom frames, it adds a note to show that the station is a node and/or a TCP/IP station.

If the list is long enough so that a station is not heard for 12 hours, it will get deleted anyway.

The list may also show a column headed 'Dev.'. This will only be present where the sysop has added to the node a small hardware board that measures the received signal audio level. Specifically, it gives an indication of the peak audio level. By means of a software configuration control and prior calibration, this gets converted into an indication of the transmitting station's signal deviation. It does this by sampling the audio level after every valid packet.

Care must be taken over its interpretation. It does not measure independently the two tone levels - it is assumed that whatever local standards that relate to pre-emphasis ( i.e. use it or not ) have been implemented.

Often, packet stations are set up, and the audio level tweaked until it appears to work reasonably error free. The idea of this add-on is that, having done that, you then connect to the node and display the heard list to see an indication of your actual deviation. It may then be fine tuned to set it correctly. Local advice must be taken over the correct setting as it depends on the channel spacing being used ( e.g. 12.5, 25 or other KHz ).

The meter will give the wrong answer on the following conditions :

- A badly distorted audio signal
- Badly off frequency
- Incorrect adherence to local pre-emphasis standards
- A noisy signal

If you connect, then correct your deviation to the correct display then find performance has deteriorated, it indicates one of the above problems. It is not that the meter doesn't work, it is an indication of a fault elsewhere.

It is in your own interest for those around you to use the correct deviation. The list also allows you to see the deviation of others - so apply peer pressure if someone over or under deviates. Remember it is NOT a case of the higher or the lower the better - it is having the setting RIGHT.

---

The system may also be used to migrate users towards a lower deviation in advance of moving to narrower channel spacing.

The heard list may also show received signal strength. It shows how strong each station is at the receiver. The display will either be in dBm format or in the familiar 'S1 to S9' format ( or 'S9+' for big signals ). To understand this properly will need some guidance from the sysop. By looking at the METER parameters you can find the noise floor in dBm, and if the display is in dBm can subtract one from the other to find out how much stronger your signal is than it needs to be. How accurate this is depends on the radio and its calibration. It could be as close as +- 1 dB or it could be wildly wrong.

### Links

The LINKS command shows the level 2 connections to the node. This is usually of academic interest, but I use it in testing. The display shows the links, one per line, with the two callsigns, the link state, the port number and the current number of retries.

### Mode

The MODE command is a bit like the PARMS command. It shows a number of additional parameters. These are as follows as shown by example :

```

MODE
THENET:G8KBB-5> 0 1800 6 3 2 20 0 600 2 900 1 31 0 1 1 0 0

```

with the following meanings :

0	Host mode protocol ( 0 = standard, 1 = DCD mode )
1800	CWID period. Delay in seconds between CWID
6	CWID speed 10's of msec per dot. 6 equals 20 wpm
3	Enable / disable nodes broadcasts mask.
2	RS232 protocol, 0 = crosslink, 1,2 or 3 are KISS
20	TxDelay in 10's of milliseconds ( Centiseconds ?? )
0	Full duplex control. 0 equals simplex
600	RS232 port nodes broadcast interval in seconds
2	Nodes broadcast algorithm port mask
900	Beacon period in seconds
1	'connect' redirector. 0 is to HOST, 1 is to BBS, 2 is to DXCluster
31	Each bit controls one of the 'user' help messages, 8 bit TALK, case sensitivity & TexNet i/f
0	This byte controls the broadcasting of 'hash' nodes
1	This byte enables / disables the extra alias operation
1	If set to '1', a remote disconnect on a circuit will cause a node reconnection
0	The bits of this controls the operation of the node with regard to 'slime trails'
0	The bits of this control whether digi uplinks and downlinks are permitted

If you want additional details, ask the sysop for a copy of the overview guide.

### Parms

This shows the node parameters as per TheNet 1.01 ( I am not going to list them again here. Sorry ).

### Bye and Quit

These commands disconnects you from the node, closing the link. It says goodbye before disconnecting you if it has been so configured by the sysop. Quit does just the same as Bye does.

---

## DXcluster

If there is a local DXcluster, this command may have been configured by the sysop to connect you to it. It therefore operates in a manner very similar to the BBS command.

## Stats

The stats command gives lots of data about the node operation. A full description of the information is contained in the overview document.

## IProute

This command is used by the sysop to configure the IP route table. It may also be used to display the router table.

## Arp

This command is similar to the IProute command, but shows the Arp table. The Arp table provides a translation from IP address to callsign. The final column, if present, contains the ARP timer values for each entry. This is for entries made automatically from ARP protocol messages. The timer is in minutes and the entry will be deleted when it reaches zero. Note that the node does not queue frames for transmission whilst awaiting ARP resolution. If you PING the node and it generates an ARP request, you will get a response the NEXT time you PING the node as it will have deleted the first request.

## IPaddress

This command is used to set or display the current node IP address.

## BBSAlias HostAlias DXCAlias

These commands are used to set additional aliases for the node. It can be configured by the sysop to accept connect requests ( uplinks ) to the node callsign, the node alias, or the 3 aliases shown by these commands. When the node accepts a connection to one of these aliases, it will immediately invoke the BBS, DXC or HOST commands for you. The way this would normally be used is as follows. Suppose your local ( for example ) BBS was not accessible on the frequency that the node operates on. The BBS alias can be configured to provide easy access across other nodes to the BBS. Hence in the case of the Ipswich nodes, GB7MXM does not have a port on 144.650, but the node IPS2 on 144.650 can get to it by means of another node and a 9600 baud link. If IPS2 is set to accept the extra aliases, and if BBSAlias is set to MXMBBS, then anyone who tries to uplink to MXMBBS in the Ipswich area would be automatically connected to GB7MXM. It goes without saying that if GB7MXM had a port on 144.650 itself, then chaos would ensue.

## MTU

This command allows configuration of the MTUs for IP users. The parameters have the following meanings :

Parameter	Default	Controls
1	256	The MTU for the radio port, AX.25 encapsulation
2	256	The MTU for the RS232 port, AX.25 encapsulation
3	236	The MTU for the Net/Rom encapsulation
4	257	The maximum number of data bytes in a received L2 frame

5                    328                    The maximum number of bytes in a received L2 frame

For more details on the IP router, consult the 'overview' document.

#### ADC

This command is used to read up to 4 channels of DC voltage. Whether they are enabled, and what they read is a local decision by the Sysop.

#### L3MHEARD

This is another heard list. It records the identities of nodes that gateway Net/Rom frames through this node, together with data such as the number of frames sent, port, time since last heard and last destination node.

---

 THENET X1-J4 QUICK REFERENCE GUIDE
 

---

 =====  
 SWITCH COMMANDS
 

---

```

ACL          [ {CALLSIGN + VALUE} | {CALLSIGN - VALUE} |
              { * VALUE } | { & value } ]

ADC
ADC1         [ * | Units_for_Voltmeter_channel_1 ] (see note 6)
ADC2         [ * | Units_for_voltmeter_channel_2 ] (see note 6)
ADC3         [ * | Units_for_Voltmeter_channel_3 ] (see note 6)
ADC4         [ * | Units_for_voltmeter_channel_4 ] (see note 6)
ALIAS        [ * | NEW_ALIAS ]
ARP          [ ipaddr [ { - ptcl } | [+ [P] ptcl callsign [ DG | VC ] ] ]
ARPTIMER     [ * | enable_flag [ * | timer_rate ] ]
AUDIT        [ NUMBER_FROM_0_TO_255 ]
BBS          [ CALLSIGN | * | ? ]
BBSALIAS     [ * | NEW_ALIAS ]
BRATE        [ * | radio_brat [ * | rs232_baud_rate ] ] ( PK96 only )
BTEXT        [ * | BEACON_MESSAGE_TEXT ] (see note 6)
BYE
CALIBRATE    [ PERIOD_VALUE_FROM_1_TO_60 [ TOGGLE_VALUE_1_TO_PERIOD ] ]
CLOSEDOWN A
CONNECT      [ CALLSIGN [ [V] DIGILIST ... ] ]
CQ           [ MESSAGE_FOR_CQ_PACKET ]
CTEXT        [ * | CONNECT_MESSAGE_TEXT ] (see note 6)
DXCLUSTER    [ CALLSIGN | * | ? ]
DXCALIAS     [ * | NEW_ALIAS ]
HELP
HOST         [ CALLSIGN | * | ? ]
HOSTALIAS    [ * | NEW_ALIAS ]
INFO         [ SYSOP_SET_MESSAGE ] (see note 6)
IPADDRESS    [ NEW_IP_ADDRESS ]
IPBROADCAST  [ NEW_IP_ADDRESS ]
IPROUTE      [ ipaddr [ /bits ] [ - | { + port [ ipgateway [ metric ] ] ] ]
IPSTATS      [ { NEWPARAM | * } { NEWPARAM | * } ..... ] (see note 5)
L3MHEARD     [ NUMBER_FROM_1_TO_100 ]
LINKS
MANAGER
METER        [ { new_value | * } ... ] (see note 5)
MHEARD       [ NUMBER_FROM_1_TO_100 ]
MODE         [ { NEWPARAM | * } { NEWPARAM | * } ..... ] (see note 5)
MTU          [ list_of_parameters ] (see note 5)
NODES        [ * [*] | NODECALL {+|-} IDENT QUAL COUNT PORT NEIGHBOUR
              [DIGIS]]
PARMS        [ { NEWPARAM | * } { NEWPARAM | * } ..... ] (see note 5)
QUIT
RESET        [ ANY_CHARACTER ]
ROUTES       [ PORT NODECALL [ DIGILIST ... ] { + | - } PATHQUALITY ]
STATS
SYSOP
TALK         [ STRING ]
UI           DEST   STRING_OF_TEXT_TO_BE_SENT_IN_UI_FRAME
USERS
  
```

---

Note 1 Any command may be enabled or disabled by the use of the '+' or '-' modifier, as shown below :

ANY\_COMMAND [ + | - | THAT\_COMMANDS\_PARAMETERS ]

Commands may be caused to be displayed or not in the prompt screen if the '-' or '+' is followed by 'D' ( or 'd' )

Note 2 IP addresses are of the form nnn.nnn.nnn.nnn where nnn is a number 0..255

Note 3 IProute port parameter is 0 for radio or 1 for rs232

Note 4 ARP ptcl parameter is AX.25 or Net/Rom (may be abbreviated)

Note 5 The commands will accept the 'old' syntax of '\* \* \* value...' or the new offset & value syntax of '/ parameter\_number value'

Note 6 Voltmeter channels 1,2,3,4 correspond to ADC channels 3,4,1,2 respectively. The ADC command reads them.

HOST 'ESCAPE' COMMANDS

<escape> C  
 <escape> D  
 <escape> P [ NEW\_PASSWORD ]

=====

UNDERSTANDING ACL VALUES

UNDERSTANDING AUDIT BITS

Bit	Function	Bit	Function
0	Bar all incoming L2 connects	0	Issue L1 stats every 10 mins
1	Bar outgoing L2 downlinks	1	Audit L2 connects & discs
2	Ignore nodes broadcasts	2	unused
3	Bar gatewaying at level 3	3	Audit L4 connects & discs
4	Bar incoming L4 connects	4	Audit L7 use of sysop cmd
5	Bar outgoing L4 connects	5	Audit all L7 switch commands
6	ignore SSID in searching	6	Issue CPU stats every 10mins
7	unused	7	unused

=====

-----  
PARMS PARAMETERS

Number	Min	Max	Function
1	1	400	Maximum number of destination nodes
2	0	255	Minimum quality for auto update
3	0	255	HDLC ( radio, port 0 ) default quality
4	0	255	RS232 ( crosslink, port 1 ) default quality
5	0	255	Initial value for obsolescence counter
6	1	255	Minimum obsolescence for node broadcast
7	0	65535	Auto update broadcast interval (seconds)
8	0	255	Level 3 ( network ) Time To Live Initialiser
9	5	600	Level 4 ( transport ) timeout (seconds)
10	1	127	Level 4 ( transport ) retries
11	1	60	Level 4 ( transport ) ack delay (seconds)
12	1	1000	Level 4 ( transport ) busy delay (seconds)
13	1	127	Level 4 ( transport ) window size (frames)
14	1	127	Level 4 ( transport ) congestion ctrl threshold
15	0	65535	Level 7 ( switch ) inactivity timeout (seconds)
16	0	255	Persistence for transmit delay
17	0	127	Persistence slottime delay (10's of msec)
18	1	15	Level 2 ( link ) T1 timeout, ie FRACK (seconds)
19	1	7	Level 2 ( link ) window size (packets)
20	0	127	Level 2 ( link ) retries
21	0	6000	Level 2 ( link ) T2 timeout (10's of msec)
22	0	65535	Level 2 ( link ) T3 timeout (10's of msec)
23	0	1	Level 2 ( link ) digipeat enable flag
24	0	1	Callsign validation flag
25	0	2	Node beacon ctrl (0=off, 1=if active, 2=always)
26	0	1	CQ broadcasts enable flag

=====

MODE PARAMETERS			
Number	Min	Max	Function
1	0	1	Hardware handshake host control mode flag
2	0	3600	CWID repeat period ( seconds )
3	4	10	CWID speed ( 10's msec per dot )
4	0	3	Nodes broadcast channel enable flags where 0=none, 1=HDLC only, 2=RS232 only, 3=Both ports
5	0	3	Crosslink protocol selection 0=crosslink, 1=KISS, 2=KISS+selective copy, 3=KISS+allcopy
6	0	255	TX keyup delay ( 10's of milliseconds )
7	0	1	Full duplex enable flag
8	0	65535	RS232 ( port 1 ) node broadcast period ( secs )
9	0	3	Node broadcast algorithm control flags 0=off, 2=RS232 port, 1 & 3 not normally used
10	600	3600	Beacon interval ( seconds )
11	0	2	Connect redirection to BBS flag
12	0	255	Help messages flags, 8 bit TALK, case & TEXNET
13	0	3	Hash node broadcast disable (one bit per port)
14	0	1	Enable extra aliases monitoring if set
15	0	1	Enable auto reconnect to node after remote dis
16	0	3	Slime trail control. Each bit controls... Bit 0 if set hides slime trails in nodes list Bit 1 if set causes slime trails to be ignored
17	0	3	Digipeat control. Each bit controls ... Bit 0 set makes node to refuse digi'd L2 uplink Bit 1 set, node refuses to allow digi downlinks

IPSTAT PARAMETERS			
Number	Min	Max	Function ( Those marked '*' are not used )
1	0	3	ip L2 AX.25 Modes ( 1 bit per port, 1=DG )
2	0	1	ip Forwarding, 1=enable router, 0=disable
3	2	255	ip Default TTL
4	0	0	ip In Receives
5	0	0	ip In Header Errors
6 *	0	0	ip In Address Errors
7	0	0	ip Forwarded Datagrams
8	0	0	ip In Unknown Protocols
9 *	0	0	ip In Discards ( TTL exceeded )
10 *	0	0	ip In Delivers
11	0	0	ip Output Requests
12 *	0	0	ip Output Discards
13	0	0	ip Output No Routes
14 *	1	30	ip Reasm Timeout
15 *	0	0	ip Reasm Requireds
16 *	0	0	ip Reasm OKs
17 *	0	0	ip Reasm Fails
18	0	0	ip Frag OKs
19 *	0	0	ip Frag Fails
20	0	0	ip Frag Creates

MTU PARAMETERS			
Number	Min	Max	Function

```

-----
1      64   1024  IP Port 0 ( Radio Port ) Level 2 AX.25 MTU
2      64   1024  IP Port 1 ( RS232 Port ) Level 2 AX.25 MTU
3      64    236   IP Net/Rom port MTU
4     257   1025  Level 2 Max data bytes in a packet before error
5     328   1096  Level 2 Total max packet size in bytes
-----

```

=====

METER PARAMETERS

Number	Min	Max	Function
1	0	32767	The meter mode flags
2	0	255	The deviation meter scaling factor
3	0	255	The signal strength meter noise floor value
4	0	255	The S meter display format multiplier
5	0	255	The dBm meter display format multiplier
6	0	255	The dBm noise floor value
7	0	255	The voltmeter channel 1 multiplier
8	0	255	The voltmeter channel 2 multiplier
9	0	255	Voltmeter Channel 1 offset value
10	0	255	Voltmeter Channel 2 offset value
11	0	255	The voltmeter channel 3 multiplier
12	0	255	The voltmeter channel 4 multiplier
13	0	255	Voltmeter Channel 3 offset value
14	0	255	Voltmeter Channel 4 offset value

=====

THE METER MODE FLAGS ( METER parameter 1 )

BIT	If set, then ....
0	The deviation meter is enabled
1	The signal strength meter is enabled
2	The signal strength is shown as S points rather than dBm
3	ADC channel 3 is enabled ( voltmeter channel 1 )
4	ADC channel 4 is enabled ( voltmeter channel 2 )
5	Voltmeter channel 1 divisor is 1000 rather than 100
6	Voltmeter channel 2 divisor is 1000 rather than 100
7	Voltmeter channel 1 display resolution is 0.1 not 1.0
8	Voltmeter channel 2 display resolution is 0.1 not 1.0
9	ADC channel 1 is enabled ( voltmeter channel 3 )
10	ADC channel 2 is enabled ( voltmeter channel 4 )
11	Voltmeter channel 3 divisor is 1000 rather than 100
12	Voltmeter channel 4 divisor is 1000 rather than 100
13	Voltmeter channel 3 display resolution is 0.1 not 1.0
14	Voltmeter channel 4 display resolution is 0.1 not 1.0

=====

HELP MESSAGES FLAGS ( MODE Parameter 12 )

BIT	FUNCTION
-----	----------

- 
- 0 If set, the 'please wait, trying xxxx' operates
  - 1 If set, all commands appear in help for sysop
  - 2 If set, the 'goodbye' message is given
  - 3 If set, a welcome message is enabled ( CTEXT )
  - 4 If set, nodes are shown as 'alias:callsign'
  - 5 If set, TALK data is passed as 8 bit data
  - 6 If set, node aliases are deemed to be case sensitive
  - 7 If set, enables the TexNet "\*\*\* LINKED to" interface

=====

ARPTIMER PARAMETERS

Number	Min	Max	Function
1	0	1	0 disables ARP table mgmt, 1 enables
2	15	1440	ARP entry expiry time in minutes

---

---

 TheNet X-1J Release 4

## Configuration instructions

## 1. Introduction

This document describes the build process for creating a ROM image for TheNet X-1J release 4. This process differs from the early versions of TheNet-X in that it is delivered as two files rather than three. This is in response to a number of requests for a simpler process. In addition the patcher has been considerably changed and utilities for hex conversion are included. A routine to print out passwords in a 'user friendly' format is included together with a utility to change the help text in THENET2.X1J.

BEWARE ! The files carry the same names as TheNet X-1J ( original release ). Be sure to use the correct set of files !. The files also carry the same names for the PK96 and TNC2 versions but the files are not interchangeable ( patch.\* and sethelp.\* are also version specific ).

## 2. Files.

The ROM image comes as two files,

```

THENET1.X1J
THENET2.X1J

```

These two files are loaded into memory as described below.

Before loading them however, both should be configured as described in section 3.

In addition, the following files are also provided :

```

PATCH.EXE
INTEL.EXE
MOTOROLA.EXE
SETHELP.EXE
PRINTPWD.EXE
INTEL.C
MOTOROLA.C
SETHELP.C
PRINTPWD.C
X1JCONV.C
X1JCONV.DOC
X1JCONV.EXE

```

PATCH.EXE is the windowing patch utility for the ROM images. INTEL.EXE and MOTOROLA.EXE are utilities that are designed to convert binary files into hex notation, in the Intel Intellec and Motorola S formats. SETHELP.EXE changes the help screen and PRINTPWD.EXE does the pretty printing by extracting the password from a parameter save file created by PATCH.EXE.

The ROM image consists of two halves, one for the lower half of a 512K EPROM, and one for the upper half. The files are loaded as shown :

<b>FILENAME</b>	<b>Load starting at hex</b>
=====	
<b>THENET1.X1J</b>	<b>0000</b>
<b>THENET2.X1J</b>	<b>8000</b>

No information on how to load the files into a programmer is presented as all are different. Typical scenarios are however given in section 5.

### 3. Configuration

Each of the two halves of the ROM image contains two different parts, a common set of drivers & interrupt routines and part of the functionality of the node. Part 1 contains the level 2, 3 and 4 software plus the IP router. Part 2 contains the switch. For the TNC2 each must be patched in an identical way to reflect the desired operation as each part contains an identical section at the start of the file for configuration data. For the PK96 only the first file needs to be patched. This patching may be done manually or it may be done with the patcher.

The first part of the ROM images is identical to TheNet 1.01 in its configuration. These parameters are followed by additional ones for the extended version as shown :

HEX LOCN	FIELD SIZE	DESCRIPTION
003B	6 BYTES	CALLSIGN OF THE NODE
0041	BYTE	SSID OF THE NODE CALLSIGN
0042	6 BYTES	ALIAS OF THE NODE
004A	WORD	MIN AUTO UPDATE QUALITY
004C	WORD	HDLC DEFAULT ROUTE QUALITY
004E	WORD	DEFAULT RS232 ROUTE QUALITY
0050	WORD	INITIAL NODE LIFETIME
0052	WORD	MIN LIFETIME TO BROADCAST
0054	WORD	BROADCAST INTERVAL IN SECONDS
0056	WORD	LEVEL 3 TIME-TO-LIVE INITIALISER
0058	WORD	LEVEL 4 TIMEOUT IN SECONDS
005A	WORD	LEVEL 4 RETRIES
005C	WORD	LEVEL 4 ACK DELAY IN SECONDS
005E	WORD	Level 4 CHOKE TIMEOUT IN SECONDS
0060	WORD	LEVEL 4 WINDOW SIZE
0062	WORD	NUMBER OF BUFFERED FRAMES PER CONNECTION
0064	WORD	NO ACTIVITY TIMEOUT IN SECONDS
0066	WORD	PERSISTENCE
0068	WORD	SLOT TIME IN TENS OF MILLISECONDS
006A	WORD	LEVEL 2 INITIAL T1 COUNTER IN SECONDS
006C	WORD	LEVEL 2 WINDOW SIZE
006E	WORD	LEVEL 2 RETRIES
0070	WORD	LEVEL 2 INITIAL T2 COUNTER
0072	WORD	LEVEL 2 INITIAL T3 COUNTER
0074	WORD	LEVEL 2 DIGIPEAT ENABLE FLAG 0 = DISABLED, 1 = ENABLED
0076	WORD	CALLSIGN VALIDATION, 0 = OFF, 1 = ON
0078	WORD	BEACON MODE, 0 = OFF, 1 = AFTER TRAFFIC, 2 = ALWAYS
007A	WORD	CQ ENABLE FLAG, 0 = DISABLED, 1 = ENABLED
007C	BYTE	FULL DUPLEX FLAG, 0=SIMPLEX, 1 = DUPLEX
007D	BYTE	SEND FLAGS IF NO DATA NEEDED, 1 = YES
007E	BYTE	COMMAND LEAD-IN CHARACTER ( ESCAPE )
007F	BYTE	TRANSMIT KEY-UP DELAY, 10's OF MILLISECS
0080	80 BYTES	DEFAULT PASSWORD
00D0	BYTE	NULL BYTE AT END OF PASSWORD
00D1	80 BYTES	INFORMATION MESSAGE

---

0121	BYTE	NULL AT END OF INFORMATION STRING
0122	WORD	CW REPEAT RATE IN SECONDS. 0 DISABLES
0124	BYTE	CW BIT SPEED IN 10's OF MILLISECONDS 6 = 20 WPM
0125	BYTE	DEFAULT HOST MODE. 0 = NORMAL 1 = HARDWARE HANDSHAKE CONNECT CONTROL
0126	BYTE	CROSSLINK PROTOCOL MODE CONTROL 0 = TheNet NORMAL CROSSLINK PROTOCOL 1 = USE KISS INSTEAD OF CROSSLINK 2 = AS PER 1, ALSO COPY NON NODE PKTS 3 = AS PER 2 BUT COPY ALL PACKETS
0127	BYTE	MHEARD LIST LENGTH. 0 = OFF, MAX = 100
0128	BYTE	NODE BROADCAST CONTROL. 0 = NO BCAST 1 = HDLC PORT ONLY, 2 = RS232 PORT ONLY 3 = BOTH PORTS
0129	WORD	RS232 NODE BROADCAST INTERVAL ( SECS ) 0 DISABLES
012B	BYTE	NODE BROADCAST ALGORITHM CONTROL. BIT 0 = HDLC, BIT 1 = RS232 WHEN BIT SET, IMPLEMENT VARIANT ALGO.
012C	8 BYTES	OPTIONAL BEACON DIGI LIST, NULL TERM.
0134	WORD	DEFAULT BEACON INTERVAL IN SECONDS
0136	BYTE	CONNECT REDIRECTION, 0=HOST 1=BBS 2=DXC
0137	BYTE	HASH NODE CONTROL. Each bit controls whether nodes whose alias starts with a '#' are included in node broadcasts on a specific port. Bit 0 determines port 0 ( the radio ), bit 1 controls the RS232 port. If a bit is set, hash nodes are not broadcast.
0138	4 BYTES	THIS IS THE NODE'S AMPRNET ADDRESS. It is a numeric address of 4 bytes. Each byte corresponds to one byte of the address, for example if the address is 44.131.16.31, then the data stored at each of the bytes 138, 139, 13A and 13B respectively will be 1F, 10, 83 and 2C. Contact your local co-ordinator for an address.
013C	4 BYTES	THIS IS THE AMPRNET ADDRESS USED BY THE NODE TO RECOGNISE BROADCASTS. The data is stored in the same way as for the node's address ( as shown above ). A typical address would be 44.131.0.0 for the UK.
0140	BYTE	IP PORT MODE CONTROL. This byte controls the default modes used for AX.25 frames on each port. It is bit mapped, with bit 0 controlling the radio port and bit 1 controlling the RS232 port. If a bit is set, the default mode for that channel is datagram ( UI frame ), if not it is virtual circuit.
0141	BYTE	IP INITIAL TIME TO LIVE
0142	BYTE	IP ENABLE FLAG. If zero, the IP router is disabled. If not zero the IP router is operational.
0143	BYTE	HELP MESSAGES CONTROL BYTE. Each bit enables or disables a different type of help message as follows :

---

---

Bit 0 - 'trying to connect' message  
 Bit 1 - sysop sees all commands in help  
 Bit 2 - give a 'good-bye' message to users  
 Bit 3 - enables the connect text message  
 Bit 4 - show nodes as alias:callsign  
 Bit 5 - pass 8 bit data in TALK if set  
 Bit 6 - Make node alias handling case sensitive  
 Bit 7 - Enable TexNet interface handler

0144      WORD      MTU\_IP0  
 This is the MTU for port 0 Level 2 AX.25 ( the radio port ) for the IP router.

0146      WORD      MTU\_IP1  
 This is the MTU for port 1 Level 2 AX.25 ( the RS232 port ) for the IP router.

0148      WORD      MTU\_IPN  
 This is the MTU for the Net/Rom subnetwork layer. It should NOT exceed 236 for compatibility with Net/Rom

014A      WORD      MTU\_I\_MAX  
 This is the maximum number of data bytes, plus one, that will be accepted in a received L2 AX.25 packet. Above this will cause an error.

014C      WORD      MTU\_L2\_MAX  
 This is the absolute limit on the number of bytes in a received AX.25 packet, counting the data, control, and address information. It is set to 328 usually ( 256 data bytes, 2 control and 70 address ).

014E      BYTE      Auto reconnect to node on remote dis if 1

014F      BYTE      NoSlime -  
 Bit 0, if set, stops 'slime trails' being displayed in the nodes table. Bit 1, if set, stops slime trails being accepted by the node.

0150      BYTE      Bit 0 if set bars digi L2 connects to the node. Bit 1 if set bars digi downlinks from the node.

0151      BYTE      DEVIATION METER DEFAULT.  
 This is the default value for the RX Deviation meter.

0152      WORD      METER FLAGS. Each bit controls a meter function

0154      BYTE      S METER SIGNAL MINIMUM This parameter is the ADC noise floor reading ( 0 to 255 )

0155      BYTE      S METER MULTIPLIER This is used to scale readings to give the S1-S9 style display

0156      BYTE      DBM MULTIPLIER This is used to scale readings to give the dBm style display

0157      BYTE      DBM NOISE FLOOR This is the reading that corresponds to no signal on calibration

0158      BYTE      This is the voltmeter channel 1 scaling factor

0159      BYTE      This is the voltmeter channel 2 scaling factor

015A      BYTE      This is the voltmeter channel 1 offset value

015B      BYTE      This is the voltmeter channel 2 offset value

---

PK96 NOTE - Parameters at addresses 0x151 through 0x15B are for the meter control, which are not used in the PK96. Instead, the default baud rate parameters are stored at addresses 0x151 and 0x152 as shown below :

0151	BYTE	Radio port default baud rate
0152	BYTE	RS232 port default baud rate

The patch utility will not assist in changing the help text. That text is positioned at the end of THENET2.X1J. It is a null terminated string of characters. Newlines are represented by the value 0xd ( decimal 13 ). It can be as long or as short as you like, as long as the length does not exceed the end of the eprom. The easiest way to change the help text is to use the SETHELP.EXE utility described below.

Any problems, give me a ring !

#### 4. The PATCH utility

The patch utility is designed to help configure the two ROM images in a manner that is not as user hostile as hand crafting a binary image ( only one file for PK96 ). It is invoked as follows :

```
PATCH [ file1 file2 ]
```

If no parameters are given, it will look for files THENET1.X1J ( and THENET2.X1J for TNC2 ) in the current directory. It will stop if it cannot load them. If the images are renamed, they may be given as parameters. If this is done, both files must be given for the TNC2 or just file1 for the PK96, with file1 corresponding to part 1 and file2 corresponding to part 2. The program is menu driven, with extensive help provided on the operation of the program and each parameter. It also tries to make sure that only valid data is entered.

The program may also be instructed to load and save textual representations of the parameters. These consist of ASCII files, with one parameter per line. Each parameter consists of the name of the parameter, and equals sign, and the value for that parameter. The values are mainly numeric, with the obvious exceptions of things like the callsign, alias, password, info message etc. To get an example of the format, use the patcher to dump a file and look at it. The idea of this is not simply to load and dump whole images, but to load partial configurations such as passwords & info messages only or parameters only. The file may be edited to remove or add lines as desired. Note that each parameter MUST only occupy one line. For the information message, whitespace before the first printable character is ignored by the program, and if a newline is to be included, it is denoted by the sequence \m ( i.e. backslash followed by the letter m ). Similarly, to include the backslash character itself, a double backslash must be entered, i.e. \\.

#### 5. The SETHelp utility

The SETHelp utility simplifies the editing of the help text which is stored at the end of THENET2.X1J. It is invoked as follows :

```
SEThelp bin2input bin2output newtext
```

If an invalid number of parameters are given, a few lines of help text will be printed on the screen and the program will terminate without making any changes. Likewise, the program will stop if it cannot load the two input files or write the output file.

The command line parameters represent the names of three files complete with extensions. Using the above names, bin2input is the name of the ROM file that contains the help text (THENET2.X1J if it wasn't renamed or copied). The second command line parameter, bin2output, represents a filename that will be created upon completion of SETHelp. This name can not be identical to the bin2input filename. The third command line parameter, newtext, is the name of a file that contains the new help text. This file must be created in advance of running SETHelp and it should only contain ASCII characters (no formatting characters). If a word processor is used to create this file, be sure to save the file as a flat ASCII file. When the file is prepared, a quick verification of the file content can be made by using the DOS TYPE command to display the help text on the screen.

When SETHelp is executed with the proper command line parameters, it will report either the number of unused bytes in the EPROM image or the number of bytes that were in excess. If the new help text exceeded the available space, edit the help text file reducing it's size and re-run SETHelp. When re-running SETHelp, it is not necessary to delete the bin2output file because SETHelp will over-write it automatically. For this reason, please be sure of the command line parameters before running SETHelp.

---

Each version (TNC2 or PK96) of the code has its own associated SETHelp.EXE file. SETHelp is specifically compiled and released with each revision (X1JR2,X1JR4, etc.) and version (TNC2 or PK96) of the firmware. The program has built-in knowledge of the help text starting address within the EPROM. To prevent users from mixing revisions or versions of SETHelp, a built in checksum routine checks the bin2input file for the expected contents. If an attempt is made to run SETHelp with an incorrect release or the PK96 version of SETHelp on the TNC2 code or vice versa, it will refuse to modify the file and report an error message. Executing SETHelp without any command line parameters will report the expected firmware revision and version on the screen.

Since SETHelp may add or delete bytes in the EPROM image, it is normal for the bin2input and bin2output files to be different in size. It does not matter if SETHelp is run before or after the patch utility.

## 6. Programming examples

There are two utilities included to facilitate conversion to hex for use in programming eproms. The source of both is also included if anyone wants a different file type. The programs have been compiled with Turbo C++.

Each has the same method of invocation,

```
INTEL infile outfile [ address ]
```

or

```
MOTOROLA infile outfile [ address ]
```

These create INTELLEC or S1 type records respectively. Each reads an input binary file and outputs a hex version. The starting address assumed for the file will be 0000 unless specified otherwise in the command line.

### 6.1 Intel format, loading as two halves

1. Use the patch program to create the desired image.
2. execute :
 

```
INTEL TheNet1.x1j part1  
INTEL TheNet2.x1j part2
```
3. load part1 into the programmer and program the lower half of the ROM.  
load part2 into the upper half.

### 6.2 Motorola format, loading as one image

1. Use the patch program to create the desired image.
2. Execute :
 

```
MOTOROLA TheNet1.x1j part1  
MOTOROLA TheNet2.x1j part2 8000  
COPY part1+part2 romimage
```
3. Edit romimage with a text editor to remove the intermediate end of file ( S0... ) marker.
4. Load romimage into the EPROM in one go & program it.

## 7. Acknowledgements

Intel and Intellec are trademarks of the Intel corporation

---

Motorola is a trade mark of the Motorola company.

My thanks to KH6ILT for the bug fixes to MOTOROLA.C and to John Bednar WB3ESS for the SETHelp utility and its documentation in this file. N8PTT also deserves credit for X1JCONV.\*.

## Deviation Meter For TheNet X-1J

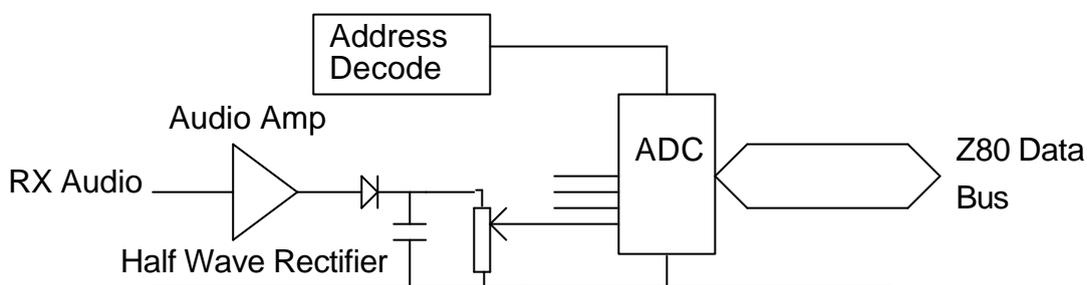
### 1. INTRODUCTION

The deviation meter is actually a peak audio level meter. It is the first of a series of applications built into TheNet X-series firmware that makes use of an Analogue to Digital Converter ( ADC ). The ADC can be (optionally) added quite simply to a TNC2.

This paper describes the circuit, its configuration and its operation, but there is a separate set of files that describes the deviation meter in more detail, containing the artwork, netlist etc. If you can't find the archive of this set of files ( originally named X1JDEV.ZIP ), contact G0JVU.

### 2. CIRCUIT OVERVIEW

The basic structure of the circuit is shown below :



The audio input is amplified and converted to a DC peak representation. This voltage is read by the ADC, which has a range of 0 .. 3 volts, and converted into an 8 bit binary value in the range 0 .. 255.

The software is configured to read the ADC at the end of each valid packet. A packet is considered valid if its CRC is correct. As soon as a valid CRC is detected, the ADC is instructed to start converting on the deviation meter channel, this conversion taking 30 microseconds. Just before the formatted frame is linked into the chain of pending received frames, the ADC is read and the value stored with the frame.

The reading therefore corresponds to the peak audio level at the end of a packet. If there are no non linear elements between the receiver's discriminator and the meter input, this will correspond fairly accurately to the peak deviation.

Care must be taken over its interpretation. It does not measure independently the two tone levels - it is assumed that whatever local standards that relate to pre-emphasis ( i.e. use it or not ) have been implemented. If used for 9600 baud FSK operation, this is not a problem.

The meter will give the wrong answer on the following conditions :

- A badly distorted audio signal
- Badly off frequency
- Incorrect adherence to local pre-emphasis standards if used for AFSK
- A noisy signal

### 3. SOFTWARE

As explained in section 2, the deviation meter is activated on receipt of a valid packet. The software interrupt routine, upon detecting a valid CRC, whether the deviation meter is installed or not, whether it is enabled or not, will instantly write the 'start channel conversion' command to the ADC ( I/O address 0x20 ) as soon as the interrupt is serviced. It will therefore also be signalled when other errors such as framing or invalid CRCs are detected.

If the CRC is correct, then about 50 to 100 microseconds later, the Z80 will read the ADC and will store the corresponding value with the packet. Again, this happens whether or not the function is enabled (it is quicker to do it than to test whether to do it or not ! ).

The value received is transferred to the internal heard list format without conversion. It is still therefore the 8 bit count ( 0 to 255 ).

When a user displays the heard list, the deviation meter parameter ( as set with the METER command ) is used to control the display. If disabled, the deviation details are not displayed in the heard list. If it is enabled, then the 8 bit count value is multiplied by the multiplier value to give a deviation reading in hertz. This is displayed as kilohertz in the heard list. If the multiplier setting is '1' it will have a full scale range of 0.2 KHz ( not very useful ). If set to 10, it will have a full scale range of 2.5 KHz. If set to 255, it will display up to 64 KHz. A typical setting will therefore be in the range 15 to 30, depending on the analogue setting. The objective is to set it such that the maximum possible audio level from the receiver, given its filters, corresponds to just under the 3 volt maximum input to the ADC.

The deviation is displayed to a resolution of 100 Hz, but it should not be assumed to be this accurate.

Small fluctuations in the reading can be expected.

Finally, to guard against an unnoticed overrange, the display in the heard list will precede the deviation reading by a chevron ( '>' ) if the value read from the ADC is 254 or 255.

### 4. Detailed Circuit Description.

All of the signals necessary for this circuit are found on the Z80 CPU chip located on the TNC motherboard, with the exception of a -5V power supply, and the audio signal to be measured.

We therefore decided to base the circuit around the Z80 CPU. We knew from experience that people are reluctant to attack the insides of their TNCs' with a hot soldering iron, and settled on the socket idea as the least intrusive way of gaining access to the required signals.

The circuit was designed with the radio ham junk box in mind, with the only "specialist part" required being the ADC itself.

The input is first amplified (x10) by U2A a quarter part of the LM324 quad op-amp. This circuit is also a precision half-wave rectifier, which is used to reduce the effect of D1's voltage offset. A more linear response is obtained with this configuration. The output is smoothed by C2, and then R3 provides a user adjustable control for the DC representation of the received audio level into the ADC.

The second part of the quad op-amp (U2B) provides a low impedance voltage reference drive for the ADC. R6 should be adjusted such that the DC level on pin 8 (VREF) on the ADC is equal to 3.0 V.

The address decode for the ADC is performed by 2 parts of the 74HC00 (U3). The ADC is mapped to the I/O address 0x20 and is selected by the X-1J software when required.

The other three (customisable) ADC channels will be available for use in later releases, for such items as temperature sensors, pressure sensors, or whatever comes to mind that will be of interest to users etc. There are also two spare op-amps within the LM324 for use with these 'customisable inputs'. Any input signal should be conditioned to provide a 0V...3V full scale signal into the ADC .

#### 4.1 Parts List

QTY	REF	PART
2	C1, C3	1 $\mu$ F @ 10V
1	C4	0.1 $\mu$ F (de coupling capacitor)
1	C2	22 $\mu$ F @ 10V
1	R1	10K
2	R5,R7	1K
1	R4	1K8
2	R3,R6	10K Miniature Potentiometers
1	R2	100K
2	D1,D3	IN4148
1	D2	2V7 Zener (250mW)
1	U1	ADC0844 (National Semiconductor) *
1	U2	LM324
1	U3	74HC00
1	JP1	3 pin HEADER
1	40PIN	40PIN WIRE WRAP SOCKET
1		PCB
1	X-1J	SOFTWARE!

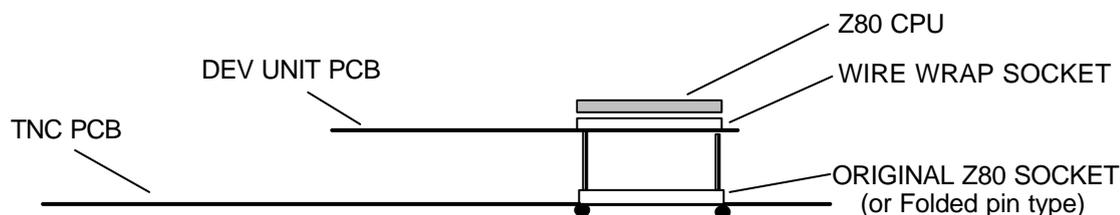
\* The ADC0844 is a four channel ADC available from most large mail-order companies. At the time of the X1J release (August '93), I have had some concern over the price of the device. When originally selected the device was available for approx. £8 in the UK. In the last two months it has risen to £18! It may be cheaper direct from outlets within the USA.

#### 5. Construction

PCB's, PCB patterns, and component locations will be available from G0JVU / G8KBB starting in September, 1993. The PCB will be single sided and approx. 3 inches by 2 1/2 inches.

However the circuit is not complex and for the more adventurous a small circuit can be built up (without the need for a 40 pin socket) and then wired to the solder side of the TNC motherboard. In fact the original prototype built on veroboard measured 1 inch by 2 1/2 inches.

The 40 pin socket on the DEV PCB is intended to be a wire wrap type, soldered to the DEV unit PCB with long legs extending down, cropped so that the PCB can be "plugged" into the 40 pin Z80 socket on the TNC motherboard. The previously removed Z80 CPU is then re-inserted in the socket on the DEV PCB.



If your TNC's Z80 is not socketed, you may like to consider adding one now! Otherwise you will need to work out some other way of getting the necessary signals to the DEV board. I suggest fitting a

"tin-plate" folded pin socket, as the wire wrap sockets will locate into them. Turned pin sockets present extreme difficulty in mating with the square section wire wrap sockets.

If there is not enough clearance in the TNC box for the extra PCB, you could consider using a header plug, with the necessary signals being transferred to the DEV board, mounted elsewhere, via a short cable.

There are three other connections that need to be made from the DEV PCB to the TNC motherboard:

JP1 Pin 1	Cable shield for audio signal (Earth)
JP1 Pin 2	Audio input from rig (5pin DIN inside TNC)
JP1 Pin 3	-5V (from TNC motherboard)

Solder in the components in the usual way, taking care not to make any solder bridges to the tracks running in-between the IC pads. Ensure the chips, and polarised components are the right way around, and take the usual ESD precautions. There are four wire links needed on the PCB to complete the single-sided circuit.

## 6. Alignment

There are four adjustable controls that will need to be set-up for correct operation.

- 1) Radio's audio drive signal into the TNC. (Volume control).
- 2) The deviation signal level control (R3).
- 3) The reference voltage for the ADC. (R6)
- 4) The METER parameter within X-1J.

The best set-up sequence (todate) is:-

- a) Adjust R6 on the DEV board to give 3.0V input into pin 8 of the ADC.
- b) Attach an oscilloscope (if available) to pin 1 of the LM324. Turn the rig's squelch control such that the squelch noise is driving the input of the DEV circuit. Adjust the rig's audio drive signal into the DEV circuit so that the output of the op-amp is just clipping the power rails. This will be seen as a 10V peak to peak signal.
- c) Adjust R3 to give the maximum DC signal into pin 3 of the ADC. This should be 3V maximum.

*Note: The function of R3 is to scale the input voltage into the ADC. The maximum signal available from the circuit is approx. 3V which corresponds to the full range of the ADC. A 3V input into the ADC will enable the circuit to present the widest possible range of values to the X-1J software. However significant "unexplained" variations in the reported deviation (MHEARD list) can be reduced by lowering the input voltage to the ADC. If you do need to adjust this voltage, then you will also have to adjust the METER parameter within the X-1J software again.*

- d) The "METER" parameter within X-1J gives the final adjustment to the circuit. Converting the received binary number into a meaningful deviation in KHz.  
In practice with the unit operating as described above, a value of 47 was found to provide a true representation of the displayed deviation to the measured deviation. To ensure an accurate measurement is given to the user, you will need to calibrate the function using a signal with a known deviation!

**As you can see from the above set up procedure. If you change the setting of the volume on the rig afterwards, all will be lost! It is preferable to take the audio from before the audio amplifier sections of the receiver ( ideally direct from the discriminator ) and to adjust the parameters or op-amp gain accordingly.**

**7. Contacts.**

G8KBB can be contacted via...

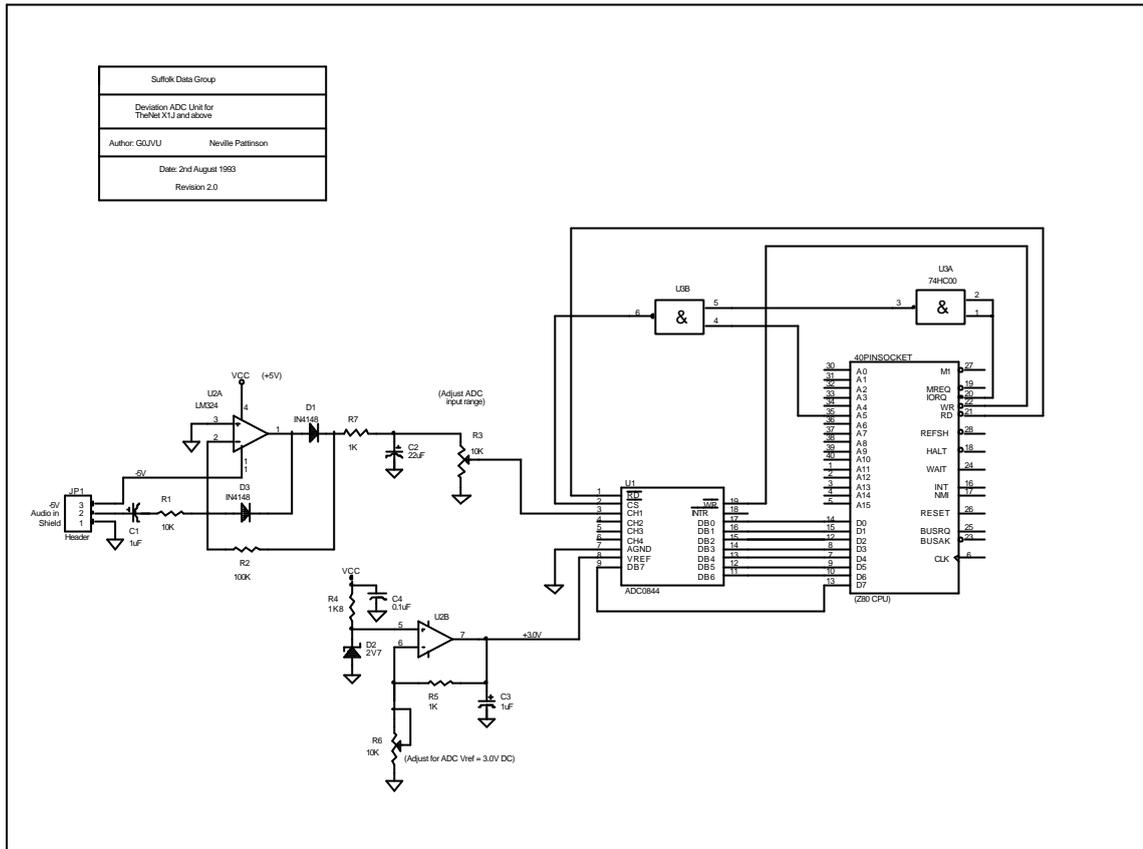
G8KBB @ GB7MXM  
+44 473 682266 ( GMT 19:00 - 2200)

Dave Roberts  
7, Rowanhayes Close  
Ipswich  
IP2 9SX  
ENGLAND

G0JVU can be contacted via....

G0JVU @ GB7MXM  
G0JVU @ G0JVU.ampr.org  
NEVILLE @ SWFMC1.SINET.SLB.COM  
Compuserve: 100021,3307  
+44 394 274795 (GMT 19:00 - 22:00)

Neville Pattinson  
10, High Hall Close  
Trimley St Martin  
Felixstowe  
Suffolk  
IP10 0TJ



---

## Signal Strength Meter for TheNet X-1J Release 4

This file contains a description of the S-meter extensions necessary for TheNet X-1J to display received signal strength.

The software assumes that there is a signal strength meter available that produces a voltage proportional to the logarithm of the input signal strength. If there is no such output available from the receiver, it is often possible to add such a function to it.

If there is such a meter output, the ADC expects an input voltage in the range 0 to 3V. It is not necessary for the voltage to be referenced to zero for no signal, as the software can compensate for this. It must not exceed the ADC reference voltage ( 3V ).

If there is no such meter output, then one may be created by adding a second IF to the receiver. If a device such as the MC3356 or MC3362 is used, it has a logarithmic Received Signal Strength Indicator ( RSSI ) output of surprising accuracy. The first prototype I built had a deviation from linearity of less than 1dB over the main part of its range, with a kink at low signal levels and compression at the high end. If you can print out the Word for Windows version of this file, a graph of the calibration data is appended to the file. If not, the raw data is contained in the file 'smeter.csv' in comma separated spread sheet format. The next one built had 2 dB variation in its linearity over the operating range.

The prototype circuit is contained in the Word for Windows version of this file. It consists of an FET input buffer ( so that the receiver is not unduly loaded ) followed by a low pass filter. The filter has a cut-off of 1 MHz. This is connected to the IF input of the receiver chip, and the output of the RSSI taken from pin 14.

The circuit is also shown in the file 'smeter.ljt'. This is an HP PCL printout file. Copy it ( a binary file with the '/B' switch if using DOS COPY ) to an HP Laserjet or compatible printer.

You must consider the circuit as a design idea that will need to be modified for your radio. My prototype was fitted to the 455 KHz IF signal from the second conversion mixer, and the low pass was needed as there was a significant component of the 10.245 MHz second conversion oscillator in the signal. The IF strip of the MC3356 will operate from 200 KHz to 50 MHz, so without the low pass it can be driven by a 10.7 or 21.4 MHz IF. What is important is that the signal is taken after the main receiver selectivity, usually its crystal filter, and before any limiting IF amplifier stages. It is also important that the signal levels are correct, so that a signal that is just detectable on the receiver just starts to increase the DC output of the RSSI. It may be necessary to adjust the signal level, for example by adding an amplifier stage before the MC3356 input.

Note that there are many devices with RSSI outputs - use any of them that are handy but remember you need one with an accurate and large range. The operational range of the MC3356 is between 50 and 60 dB, and I am told that more modern cellular radio IFs have up to 90 dB range !.

To calibrate the meter, you need a known signal, for example a signal generator of known output, and a switched attenuator with at least 5dB steps and preferably 2 dB steps. Connect a DC voltmeter to the output of the MC3356, and connect the signal generator to the receiver input operating frequency ( 144.625 for the prototype ) via the attenuator. The signal should be increased in 2 dB steps and the voltage noted for each step. The results need to be plotted as a graph. In calibrating the prototype, slight errors were noted in the calibration of the switched attenuator. These need to be subtracted out from the data.

On the graph, draw a straight line through the curve as a 'best fit' ignoring the end of range effects of noise floor, hysteresis or overload. Where the line crosses the noise floor, note the DC voltage and dBm level at this point. Calculate the slope of the curve in units of dB per volt. You should then have the following data items :

- The noise floor DC reading

- The slope of the best fit calibration curve
- The dBm point that corresponds to the crossover of the noise floor and the best fit calibration line.

The dB multiplier is calculated as :

$$\text{dB\_multiplier} = X \cdot V_{\text{ref}} / V$$

where X dB change in input caused V volts DC change ( i.e. the slope of the best fit line from the graph ), and Vref is the ADC reference voltage.

The data are input as follows :

The signal strength meter noise floor is entered as an integer in the range 0 to 255 ( hopefully a small number about 50 ish ) calculated from the DC noise floor reading from the graph ( V ) and the ADC reference voltage ( Vref ) as

$$256 * V / V_{\text{ref}}$$

The dBm meter display format multiplier is entered as calculated above from the graph. In my prototype, 54 dB change caused 2V DC change in output with a 3V reference voltage, so the multiplier was 81.

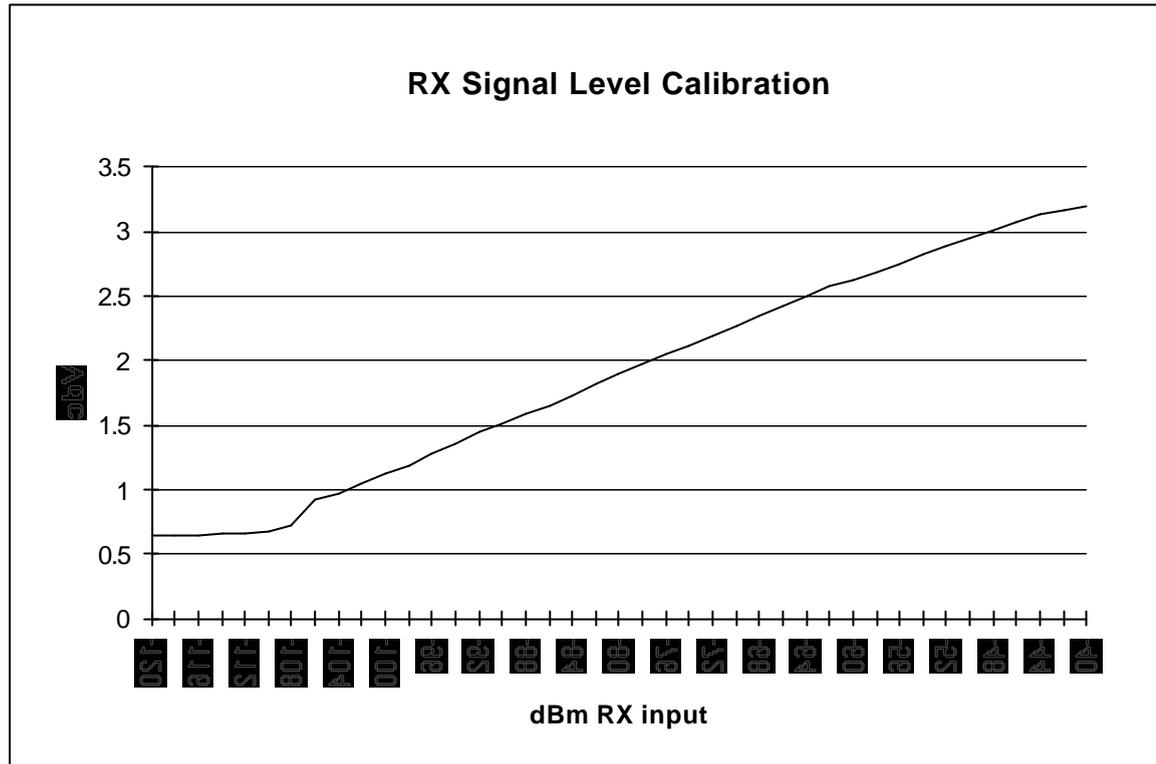
The dBm noise floor is entered at a positive integer corresponding to the complement of the dBm zero point from the graph. For example, 0.65 V DC was the noise floor reading for my prototype and the calibration line crossed this noise floor level at a dBm reading of -113 dBm. The dBm noise floor is entered at 113 ( i.e. drop the '-' ).

The S meter multiplier is set by trial and error depending on your perception of what constitutes an S9 signal !. Alternatively, it is set to the dB\_multiplier divided by the number of dB per S point, so in the previous example, if you want 4 dB per S point, set it to 20. Note that there are several 'standards' for the number of dB per S point, all vociferously defended and justified. It is better to use the dBm scale.

The output of the RSSI needs to be connected to the ADC in the TNC. The easiest way to do this is to use the squelch line in the standard TNC2 5 pin DIN connector ( pin 5 ). This signal is frequently unused in nodes. The RSSI output is connected to pin 5 in the radio, and in the TNC the signal is disconnected from the squelch circuits and connected instead to channel 2 of the ADC ( one of the unused pads on the ADC ). In TNCs such as the BSX2, the squelch signal is connected into the TNC circuits via a diode that forms a logical AND gate with the modem DCD. The easiest way to disconnect pin 5 from these circuits is to lift one end of that diode.

The lead from radio to TNC must be reasonably short as the output impedance of the RSSI is not low. If problems are found, an op-amp buffer may need to be added to give a low impedance drive.

When exploring the innards of radios looking for suitable tap point, a degree of care and ingenuity will be needed. Finding one with about the right signal level, prior to a limiter, after the main bandpass filter and without undue loading on the radio circuits is not always easy.



Example Node heard list showing dBm format

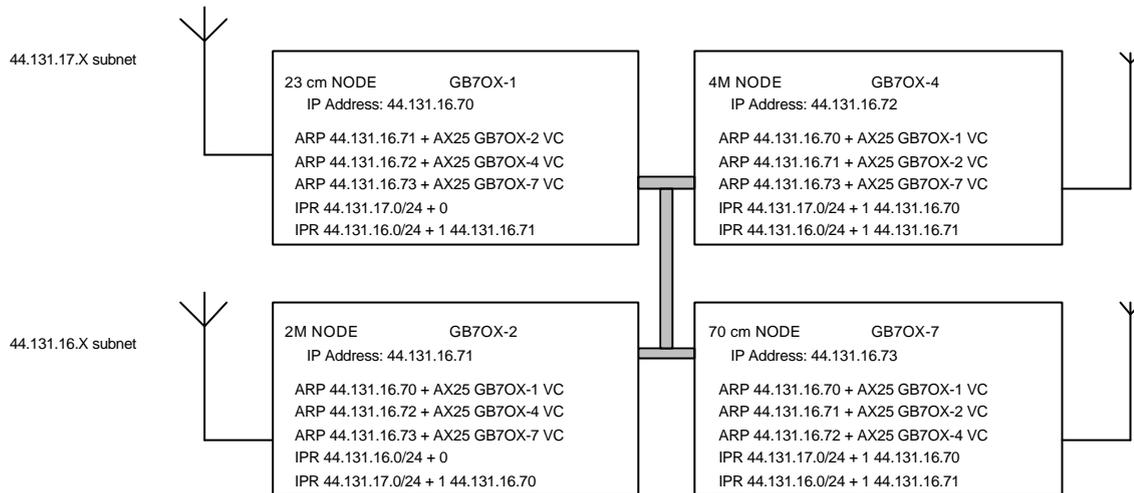
```

IPNET:G8KBB-5}
Callsign   Pkts   Port   Time      Dev.   dBm    Type
G8KBB-2    1129   1       0:0:0     5.7    -79    Node TCP/IP
FELIX      869    0       0:0:6     5.7    -79    Node TCP/IP
G0JVU-2    4285   0       0:0:40    5.9    -78    Node TCP/IP
G7MNS      368    0       0:1:17    4.1    -89    Node TCP/IP
G8STW-5    6227   0       0:4:54    5.0    -102   TCP/IP
G1YRE      61     0       0:5:27    6.2    -82    Node TCP/IP
GB7MXM     326    0       0:7:6     5.8    -78    Node TCP/IP
FB1ICL     1       0       0:13:40   6.9    -104   TCP/IP
G0TMH-5    1       0       0:13:57   6.1    -107   TCP/IP
G00EY-5    2288   0       0:14:10   6.1    -93    Node TCP/IP
G1DVU-5    1       0       0:18:39   7.6    -107   TCP/IP
G8HUE      90     0       0:21:50   5.5    -92    Node TCP/IP
G7BKO      1       0       2:0:14    7.0    -96    Node TCP/IP
G4ZEK-14   13     0       3:39:22   5.7    -79    Node TCP/IP
G0NJA      29     0       4:8:54    6.6    -91    Node TCP/IP
G7JVE-5    259    0       5:23:33   4.3    -105   TCP/IP
G8INE      5       0       8:11:28   6.3    -112   TCP/IP
G4IZC-5    69     0       8:26:29   6.8    -112   TCP/IP

```

### Using IP addresses on cross connected nodes.

Consider four X-1J nodes connected via a crosslink . The more obvious way of configuring IP connectivity between them is to allocate each an IP address, and then use relevant ARP and IPRROUTE commands to set up IP routing between them. A simplified example is shown below.



If there are several nodes in an IP subnet, they will use up a lot of addresses. It is possible to use "dummy" IP addresses to route between adjacent nodes on a crosslink. This means that you do not allocate an IP address for each node, but only one for the whole node "stack".

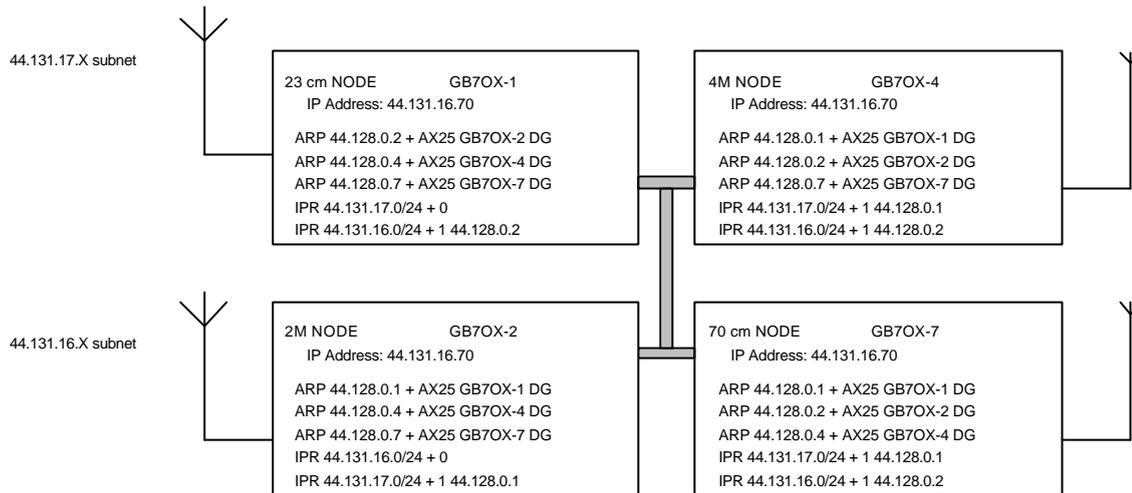
The downside of this is that IP users can only ping their local node, they cannot ping another node in the stack.

Note that these "dummy" addresses are NEVER actually used. We need to allocate addresses in order to make the correct entries in the various tables inside the nodes; but because of the way in which the protocol is implemented, the IP addresses are never used, instead the nodes are addressed by their callsign over the crosslink. Depending on how desperate for IP addresses you are, this technique could feasibly be used on point to point backbone links too. In the example I have chosen to use the ampr.org test addresses in the 44.128 series. In fact you can use any address in your node that you know won't appear in your ARP or IPRROUTE tables for real.

The way to do this may not be immediately obvious, and a worked example is given below.

The reason it works is because of the way the address translation occurs.

Suppose a packet arrives at GB7OX-4 addressed to 44.131.16.179. First the routing table on GB7OX-4 is scanned, and a match is found. This packet must go to 44.128.0.2. A scan of the ARP table shows that 44.128.0.2 is known, and we must use the MAC address (or in the amateur radio world, the "callsign") rather than an IP number to address the station directly; so the packet is addressed, and sent down the crosslink to GB7OX-2. GB7OX-2 receives the packet, and finds a match in its routing table, and so sends the packet out of its radio port according to the entries in its IPRUTE and ARP tables.



At no time in the above example does the address 44.128.0.2 appear in any protocol headers. Its use is solely to fool the node into passing the frame on to GB7OX-2. The only time it might get used if the route tracing header option of IP were used ( it is implemented in TheNet X-1J but seldom used ) when it would appear that the frame had been routed through two routers with the same IP address.

If TCP layer services are ever implemented, we may need to think again about this.

Fiona Pattinson  
G7ANH @ GB7MXM.#36.GBR.EU