

Kantronics

KPC-3XE

with USB computer interface

User Guide

**Introduction,
Getting Started,
Modes of Operation,
Command Reference, and
Hardware Specifications**

Contact Us:

Kantronics

14830 W. 117th St.
Olathe, Kansas 66062

Orders / Inquiries

(913) 839-1470
FAX (913) 839-8231
E-mail sales@kantronics.com
Website: www.kantronics.com

Service / Technical Support

(913) 839-8143 (8 AM to 12PM and 1 to 5 PM Central Time, M-F)
FAX (913) 839-8231
E-mail service@kantronics.com

REVISIONS

Revision	Date	Description
A	2026	Initial release version.

The KPC-3XE is a Kantronics hardware and software design incorporating the AX.25 Level 2 Version 2 Packet protocol as adopted by the American Radio Relay League, and FX.25 which was created by the Stensat Group in 2005, and was presented as a technical paper at the 2006 TAPR Digital Communications Conference in [Tucson](#).

We have attempted to make this manual technically and typographically correct as of the date of the current printing. Production changes to the TNC may add errata or addendum sheets. Comments and/or suggested corrections can be sent to us (see our [contact information](#)).

This document was compiled in the U.S.A.

The KPC-3XE is manufactured in the U.S.A.

© Copyright 2007-2026 by Kantronics. All Rights Reserved.

Contents of this publication or the firmware within the KPC-3XE may not be reproduced in any form without the written permission of the copyright owner.

KPC-3 Plus is a registered trademark of Kantronics Co., Inc.

KPC-9612 Plus is a registered trademark of Kantronics Co., Inc.

KAM XL is a registered trademark of Kantronics Co., Inc.

NET/ROM is a registered trademark for SOFTWARE 2000.

APRS is a registered trademark of Bob Bruninga, WB4APR.

Warranty Registration

A registration link is available on the Kantronics web page.

IMPORTANT

READ THIS PAGE BEFORE INSTALLING THIS KANTRONICS PRODUCT

This product contains SOFTWARE stored in Programmable flash Memory which is protected by both United States copyright law and international treaty provisions.

The user of this product will be deemed to be bound by the terms of the SOFTWARE license shown below. To not wish to be bound by such license, return such product and all associated documentation unused to the supplier for refund of the amount paid.

License Agreement

1. **License.** In consideration of payment of the License Fee, which is included in the price of the product, the Licensee (you) is granted by the Licensor (Kantronics Company, Inc. - Kantronics) a non-exclusive right to use the SOFTWARE and associated documentation. No ownership rights to the SOFTWARE and its Documentation are transferred from Kantronics to you.
2. **Term.** This License Agreement is effective until terminated. You may terminate this Agreement by destroying the PROM or diskette or CD and documentation. You may not rent or lease the SOFTWARE, but you may transfer the SOFTWARE and accompanying written materials on a permanent basis provided you retain no copies, and the recipient agrees to the terms of this Agreement. Kantronics may terminate this Agreement without notice if you violate any terms or conditions of the Agreement. In the event of termination of the Agreement, provisions relating to Kantronics' disclaimers of warranties, limitation of liability, remedies, or damages and Kantronics' proprietary rights shall survive.
3. **Object Code.** The SOFTWARE is delivered in object code only. The user shall not reverse compile or otherwise reverse engineer the SOFTWARE.
4. **Limited Warranty.** This product is covered by the standard Kantronics Limited Warranty, which is enclosed.
5. **General.** This License Agreement constitutes the complete Agreement between you and Kantronics.

The SOFTWARE and/or Documentation may not be exported or re-exported in violation of any export laws or regulations of the United States of America or any other applicable jurisdiction.

This Agreement shall be governed by and interpreted under the laws of the State of Kansas, United States of America.

Use, duplication, or disclosure by the Government of the United States is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer SOFTWARE clause of DFARS 252.227-7013.

Kantronics may, in its sole discretion, provide you with upgrades of the SOFTWARE and/or Documentation if you have provided Kantronics your completed Warranty registration with a copy of your receipt showing the amount you paid.

LICENSEE ACKNOWLEDGES HAVING READ AND UNDERSTOOD THIS AGREEMENT AND AGREES TO BE BOUND BY ITS TERMS. LICENSEE FURTHER AGREES THAT THIS AGREEMENT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN LICENSEE AND LICENSOR AND SUPERSEDES ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

Any questions concerning this Agreement or any other matter relating to Kantronics, Kantronics products, or business practices, may be sent to us by any of the means on our [contact information](#) page.

TABLE OF CONTENTS

User Guide	1
Contact Us:.....	2
REVISIONS	3
Warranty Registration.....	4
IMPORTANT	4
License Agreement	4
LIMITED WARRANTY	9
Return/Repair Procedures	12
Radio Frequency Interference Statement	15
RFI Suppression	16
FCC Declaration of Conformity:	16
Introduction.....	18
Welcome	18
Front Panel	18
Back Panel	18
Overview of This “User’s Guide” Manual	19
Major Uses of the KPC-3XE.....	19
Package Contents	20
Additional Parts of a Packet Radio Station	20
Our Assumptions about You	21
Documentation Conventions	21
Overview of Packet Radio.....	22
Inside a TNC – the KPC-3XE	32
Installing the KPC-3XE;.....	34
The Major Components of a Packet Station	35
Back Panel	35
Connect the KPC-3XE to a Power Source	37
Connect the KPC-3XE to a Computer	39
Connect to a Transceiver	43
Connecting a GPS Device (Optional)	55
Getting Started	56
The Front Panel of the KPC-3XE	57
Beginning a Packet Session.....	58
Giving Commands and Transmitting Data.....	58
A Sample Connect	59
Monitor Communications From Nearby Stations	60
Communicate Directly with a Nearby Station	61
Modes of Operation	63
Packet Mode of Operation	63
Remote Access	78

PBBS (Personal Mailbox)	80
GPS NMEA Interfacing	100
KA-Node	113
Introduction to Basic Packet Networking	120
K-Net Network node	123
K-Net Node Commands	124
OTHER MODES OF OPERATION	152
LSUB (“wire” Line Substitution) Mode	152
Poll Mode	153
TUP (Transparent Unproto Packet) Mode	157
Modem Mode	159
Modemk Mode	160
Kantronics Host Mode Operation	162
KISS Mode	163
DAMA (Slave Mode) Capacity	165
Command Reference	168
Introduction	168
Commands	174
Appendix A: Advanced Installation	241
Precautions	241
Cable Wiring	241
The TNC Radio port	242
Appendix B: Advanced Information	246
Assembly and Disassembly	246
Hard Reset/Self-Test	246
Calibration/Equalization	248
PTT (Push-to-Talk) Watchdog Timer	249
Microprocessor Watchdog Timer	249
A/D Converter	249
Jumpers	250
Test Points	251
Board Layout (jumpers/test points)	252
Appendix C: Options	253
Low Power Operation	253
Replacing the Lithium Battery	253
Appendix D: In Case of Difficulty	254
Does Not “Sign-On” to Computer	254
Unable to Make a “Connect”	254
Cannot Transmit	254
Cannot Return to Command Mode	255
Appendix E: Additional Information	256
Specifications	256

Messages from the TNC 257

ASCII Chart 264

LIMITED WARRANTY

KANTRONICS LIMITED WARRANTY

Effective January 1, 2009

For prompt warranty service, please register your Kantronics unit at <https://kantronics.com/>. To receive notice of future updates and new product information, sign up for the optional newsletter.

NOTE: Completion of the Warranty Registration and proof of purchase is a pre-condition to warranty coverage.

1. **WARRANTY.** Kantronics warrants to the first consumer purchaser (you”), for the Applicable Warranty Period (as described below), that the Applicable Product (as described below) will be free from defects in material and workmanship.
2. **REMEDY.** Kantronics agrees that, for any Applicable Product found by Kantronics to be in violation of the warranty of Section 1 hereof within the Applicable Warranty Period, it will, at its option, repair or replace the defective Applicable Product at no charge to you, excluding in-bound shipping charges.
3. **EXCLUSIVE REMEDY.** Repair or replacement of the Applicable Product, as provided herein, is the sole remedy available to you against Kantronics, and in no event will Kantronics be responsible for any other liability or damages or for incidental, special, or consequential damages, regardless of whether purported liability is predicated upon negligence, strict tort, contract, or other products liability theory and whether or not Kantronics is warned about the possibility of such liability or damages. **SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.**
4. **DISCLAIMER.** This Limited Warranty is in lieu of all other warranties expressed or implied and no representative or person is authorized to assume for Kantronics any other liability in connection with the sale of its products. KANTRONICS SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTY OF MERCHANTABILITY AND IMPLIED WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE FOR ANY APPLICABLE PRODUCT. IF, HOWEVER, YOU ARE A CONSUMER WITHIN THE MEANING OF 15 U.S.C. 2301(3), THE ABOVE DISCLAIMER OF IMPLIED WARRANTIES IS EFFECTIVE ONLY FOR PERIODS

OUTSIDE THE APPLICABLE WARRANTY PERIOD. **SOME STATES DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU.**

5. **APPLICABLE PRODUCTS AND PERIODS.** Kantronics products are of two types - (1) hardware units and (2) firmware and software for operation of these units, whether incorporated into the units themselves or separate from the units as adjuncts or accessories to the units. Hardware units and the media containing firmware, software and documentation are sold to the consumer purchaser and become property of the purchaser. Firmware and software are licensed for use by the consumer purchaser in return for a fee included in the purchase price of the units and do not become the property of the consumer. (See separate License Agreement provided with these products). The products to which the warranty of Section 1 hereof applies (herein "Applicable Products") and the periods during which the warranty shall apply to such products (herein, "Applicable Warranty Period") are as follows:

Applicable Products:

UNITS:

KPC-3XE, KPC-3 Plus, KPC-9612 Plus, 9612XE, KAM XL, MT1200, MT1200G

Applicable Warranty Period: One (1) year from date of purchase.

MEDIA:

EPROMS, CDs, manuals (however bound), specification and other supplemental pages or any other media on which firmware, software or documentation are supplied.

Applicable Warranty Period: Thirty (30) days from date of purchase.

6. **EXCLUSIONS.** This Limited Warranty does not apply to the cosmetic appearance of the Applicable Product; to broken or cracked cabinets; to any accessory not supplied by Kantronics which is used with the Applicable Product; to any product that has been subject to misuse abuse or overvoltage; to any product that has been modified by non-Kantronics personnel unless specifically authorized **in writing** by Kantronics; or to any product damaged or impaired by shipping (whether or not caused by poor packaging), neglect, accident, wiring not installed by Kantronics, improper parameter settings which are cleared by performing a hard reset, or use in violation of instructions furnished by Kantronics or of generally accepted industry

practice. Kantronics does not warrant that the functions contained in any software will meet your requirements or achieve your intended results; or that operation of any software will be uninterrupted or error-free or without effect upon other software used with it. *Responsibility for the selection of the hardware and software program to achieve your intended results rests with you.*

7. **REMEDY PROCEDURE.** Should you need to make a warranty claim, first contact the dealer from whom you purchased the product. If the dealer is unable to assist you, [contact us](#) prior to returning an Applicable Product to receive a Return Authorization Number. (As a practical matter, problems can often be solved in such a manner without the product having to be returned to Kantronics for repair or replacement.)

Return of any Applicable Product for the enforcement of rights under this Limited Warranty shall be at your expense. Any product returned for warranty service, which Kantronics determines to be without defect or not covered by this Limited Warranty shall be subject to the minimum charge for labor and the product will be returned to you at your sole expense. Please note, no warranty service will be provided until Kantronics has been furnished with your Warranty Registration card and copy of proof of purchase establishing purchase date.

8. **NON-ASSIGNMENT.** This Limited Warranty is not assignable by the user. Any attempt to assign or transfer any of the rights, duties, or obligations hereof is void.
9. **OTHER RIGHTS.** This Limited Warranty gives you specific legal rights and you may also have other rights, which vary from jurisdiction to jurisdiction.

Return/Repair Procedures

Important: Our repair statistics show that over 70 percent of the units returned for service do not, in fact, require any service. Therefore, we advise you to please double-check the following list of common, user-solvable, sources of difficulty before contacting Kantronics about returning your unit for service.

Checklist for Possible Problems

Should you encounter difficulty in getting your equipment to “talk” to your computer, please perform at least the following limited checks before calling or writing:

Carefully check your connections to the computer serial port, or the USB connection and operating system configuration for that port.

If you purchased third-party cables, double-check to be sure that they conform to serial port connection wiring.

Verify the terminal baud.

It may be useful to perform a “Hard Reset”. (See Hard Reset section.)

If service or repair still appears to be necessary after you have checked the items listed above, it may be wise to call, fax, e-mail, or write Kantronics to determine if the problem can be solved without returning the unit.

Verify that appropriate voltage is being supplied to the unit.

Return Procedures

When calling, report the product name and ask for the Service Department. Please have the following information available:

The unit name and serial number (the serial number is found on the bottom of the unit).

The firmware version number (the version number is displayed when you give the Version command).

If possible, you should have the unit and your computer available to perform troubleshooting operations when you call.

The Service Department telephone hours are 8:00 AM to 12:00 Noon and 1:00 PM to 5:00 PM Central Time, Monday through Friday. If you call outside these hours, the phone will just ring. If phone lines are busy, it may be faster to contact service by e-

mail. Service e-mail is checked twice per day. Before contacting us, please take the time to list the problem fully and carefully.

When writing, faxing, or e-mailing Kantronics, include a clear description of the problem, unit name, firmware version, computer type, computer software used and if possible, a list of current parameter values for your unit (as shown in a DISPLAY listing). Be sure to include a return fax number and/or e-mail address.

Returns to the factory for refund or exchange are strictly regulated. Any return for refund or exchange must be approved by the service department.

Charges

Consult the limited warranty policy in this manual for the service provisions offered by Kantronics at no charge. This warranty is considered to be in force only when the customer has submitted his completed warranty registration within 10 days of purchase, and when the stipulations of the warranty have been met.

Violations of warranty clauses will automatically void the warranty and service, or repairs will be charged to the owner.

Service outside the warranty will be charged at the cost of parts, labor, and return shipping. Units returned for service without a Return Authorization number will be subject to a minimum charge of ½ h labor plus shipping and handling.

Notice of repair cost will be sent via email message, which will include a “click-to-pay” button. When your payment information has been received, we will be notified of the payment. Your unit will be shipped when we have received this payment notification.

If payment has not been previously arranged, repaired (or unrepairable) units may be returned via C.O.D.

International Returns

- This section applies to international returns only, not to domestic returns.

In case of unit problems, first contact the dealer from whom you purchased the product. If you must return a Kantronics product to us, please observe the steps outlined below. It will save you, the customer, and Kantronics unnecessary difficulties and expenses.

- All returns must be shipped to the factory.

- All expenses for returning items to Kantronics must be paid by you, including any duty/entry fees, whether the return is for warranty or non-warranty repair.
- Usually, the best way to return items to us is by mail. However, if you wish to use one of the courier services such as DHL, UPS Expedited, Federal Express, etc., be sure to use **DOOR-TO-DOOR** service. If you use one of these services, a commercial invoice may be required. Please check with your carrier before shipping.
- Include in the description of the items on the paperwork (whether postal or courier) the words:

“U.S. GOODS RETURNED FOR REPAIR/REPLACEMENT.”

Step 1. An additional description of “Amateur radio peripheral equipment”, or “Data communications equipment”, would be helpful. It would also be helpful (but not required) to include the code number 9801.00.1035 which tells U.S. Customs agents that the package contains “U.S. goods returned without improvement/enhancement”. However, if the words “U.S. goods returned for repair/replacement” are on the paperwork, the number is not necessary.

- Provide a monetary value for customs purposes. This is usually the value of the item(s) in their current condition. A \$0 value is not acceptable for U.S. Customs.
- Inside the package, with the item(s), include;
 - a mailing or e-mail address (if available) if contact is required
 - a correct and full address for return
 - method of payment to be used for any charges (if MasterCard or VISA, include expiration date)
 - a brief description of the problem
 - a reference to any conversations with the technical/sales staff about the problem
 - and the Return Authorization number assigned
- For warranty repairs, we will pay the shipping charges to return the item(s) to you via air parcel post. If you wish to return by courier service, include an account number. To be eligible for repair under warranty, we must have a record that you sent your Warranty

Registration and proof of purchase to Kantronics, and the item(s) must still be within the warranty period at the time the return is authorized.

- For non-warranty repairs, you must pay the return shipping charges.

Radio Frequency Interference Statement

Note 1: This equipment has been tested and found to comply with the limits for a Class B digital Device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications.

There is no guarantee that the interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced Radio/TV technician for help.

The user is cautioned that any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment. The user is also cautioned that any peripheral device installed with this equipment must be connected with a high-quality shielded cable to ensure compliance with FCC limits.

Note 2: The shield of the cable, whether foil, braid, braid over foil, or double braid, must be properly terminated (connected) 360° to the connector. This is usually accomplished using a metal or metalized plastic back shell, but may be implemented by direct contact, including soldering, with metal portion of connector. Experience has indicated that cable assemblies (with connectors) advertised as "shielded" are not necessarily terminated properly, if terminated at all. Check cable construction to be sure.

RFI Suppression

In moving to the world of digital communications via computers, a new dimension of RFI may be encountered. In spite of the equipment manufacturers' diligence, each new piece of electronic equipment will react differently in each separate environment. Every amateur station will have its own unique layout, equipment variation, and antenna installations. Experience has shown that these differences are related to the total RF environment and may be causative factors in RFI induced problems. The suggestions given here may assist in resolving RFI problems encountered in any "unique" station.

- Use shielded cable for all connections between equipment.
- Make all interconnecting cables as short as is practical. A balance should be maintained between cable length and equipment proximity. At times simply moving the video monitor one foot further from an interface or other device will solve a "screen hash" problem.
- Antenna runs should be kept away from equipment control lines and/or interconnecting cables. If it is necessary for such lines to cross each other, they should do so at 90-degree angles.
- Ground leads should be as short as possible and go to a GOOD EARTH GROUND.
- Interconnecting cables appearing to act as radiators or antennas should be looped through a toroid. Be certain toroids, if used, are designed for the frequency in use.

FCC Declaration of Conformity:

NOTE: This equipment, Kantronics' KPC-3 XE, has been tested and found to comply with the essential emission and immunity requirements of the EMC Directive FCC Title 47, Part 15, Subpart B. The test results are on file at the corporate offices of Kantronics.

Type of Equipment:
Class of Equipment:

Information Technology Equipment
Class B

CE Marking Considerations

The following cautions pertain to CE Marking of this product:

- All cables connecting to DC IN, PORT 1 (VHF), and COMPUTER must be < 3 m in length
- A cable for GPS or telemetry (control output or A/D input) connected to the COMPUTER port may be ≥ 3 m in length.
- All cables, except for the DC IN port, must be shielded with the shield properly terminated 360° to the connector. See note 2 in the RF Interference section
- The nominal 12 V dc power must be supplied from a CE marked or third party approved power brick (wall wart) or ac to dc power supply. If a homemade power supply is used the components making up the supply must meet IEC/EN standards for such components.

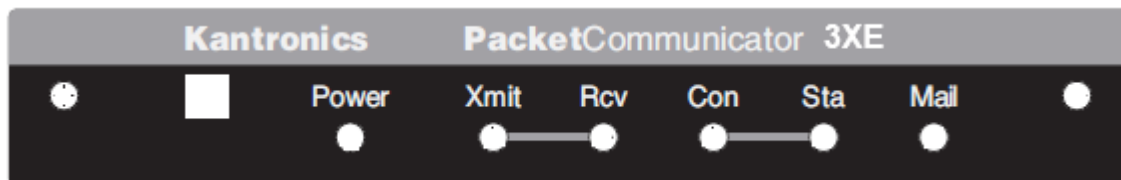
Introduction

Welcome

Welcome to the Kantronics KPC-3XE, a pathway to amateur radio packet communication.

Please review this chapter before installing this KPC-3XE as part of your packet radio station.

Front Panel



This is the front panel of the KPC-3XE.

Back Panel



This is the back panel of the KPC-3XE.

Overview of This “User’s Guide” Manual

This user’s guide provides documentation on the KPC-3XE and packet radio, including the following:

- What equipment will be needed for your packet radio station.
- A brief introduction to packet radio.
- Installing and configuring a packet radio station and making first connections.
- Getting started.
- Documentation for each mode of operation
- A full “Command Reference”.
- Details of user configurable jumpers,
- and other technical specifications.

Additional documentation and supporting material are available on the Kantronics web pages [KPC-3XE](#).

Major Uses of the KPC-3XE

By adding the KPC3XE and a computer to a ham radio station, you can send and receive packets of digital information. This allows the following:

- use a packet station to carry on real-time digital conversations with other packet stations
- send, receive, store and forward mail using the internal personal mailbox
- exchange messages to and from a community bulletin board (BBS)
- send and receive files with a file exchange app or a terminal program in a computer
- capture and beacon location data from Global Positioning System devices, or from weather station devices
- serve as a digipeater for packets from other stations

- serve as a network node point for other stations, using KA-NODE or the K-Net network node feature

And much more.

Package Contents

Check to see that these items listed below are at hand:

- the KPC-3XE unit
 - 2.1 mm dc power connector
 - A USB interface cable (type A computer end, to type B modem end)
 - The Quick-Start page

Additional Parts of a Packet Radio Station

- KPC-3XE
- An FM transceiver
- A microphone (Mic) plug and/or mating accessory plug or external audio plug for the radio (or an assembled interface cable, with connectors matching those on the transceiver)
- A computer with an available USB port
- An interface cable, to connect the TNC to the computer
- A 12 V dc power supply or AC to DC power adapter

Our Assumptions about You

We assume that you are familiar with the following or that help is available for these topics if necessary:

- general familiarity with the radio equipment, its operation, and its intended uses
- basic use of your computer and its operating system
- basic electronics knowledge needed to build or upgrade hardware (e.g., making cables)

Documentation Conventions

The following conventions are used in this KPC-3XE documentation:

To indicate a particular key, the name of the key is given in capitals. For example, press the ENTER key.

Sometimes combination of characters is needed by holding down one key on the computer while pressing another key. This is indicated by giving the name of the first character then a plus '+' character, then the second character. For example, "Ctrl+C" means "press the key labeled 'CTRL' or 'Ctrl' (i.e., the "control" key) and, while continuing to hold it down, press the 'C' key". Then release all keys, these multiple-key combinations that generate a single character, are shown in angle brackets, like this: <Ctrl+C>.

Conventions for KPC-3XE commands are covered in the "[Command Reference](#)" section of this manual.

Overview of Packet Radio

This section gives a brief overview of packet radio, for those who are new to packet radio and those who want to review the topic.

Topics covered are:

- The three basic components of a packet radio station
- Sending a message from one station to another
- Details about packets
- A brief discussion of protocols
- An overview of the internals of the KPC-3XE

If already familiar with packet radio, rest of this chapter may be skipped or just give it a quick review.

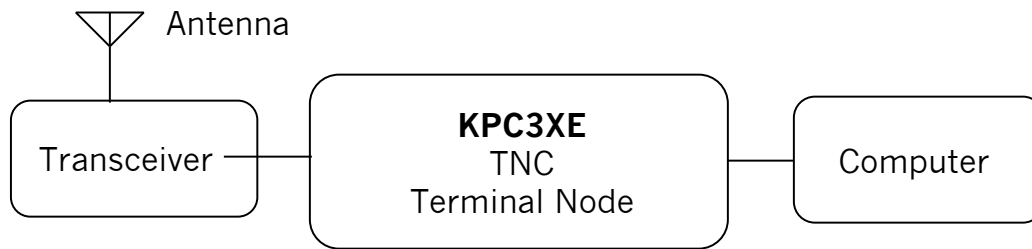
Packet radio provides the ability to send error-free digital communications to other packet radio stations. As a sender or receiver, just the messages, or files, being sent and received may be seen. But there is much going on beneath the surface. At least a basic understanding of what is involved in packet radio communication is needed.

- Do not be concerned here with *how* to do things —that will come later.

Three Basic Components of a Packet Radio Station

A packet radio station has three basic parts:

- a transceiver, with an antenna,
- a device called a TNC (i.e., Terminal Node Controller), which is a combination modem and special-purpose micro-computer, and
- a general-purpose computer (or a terminal).



The three parts of a packet radio station work together as follows:

- The **transceiver**: (1) sends and receives radio signals to and from an antenna and (2) passes audio signals back and forth between itself and the TNC.
- The **TNC** (Terminal Node Controller): (1) translates audio signals into digital information and vice versa, (2) performs a number of control and information storage functions, and (3) communicates digitally with a computer.
- The **computer** communicates digitally with the TNC, to: (1) view messages received from the transceiver or stored in a mailbox (i.e., PBBS), (2) use the computer to send data to, and receive data from, other stations, via the TNC and a transceiver, and (3) control the operation of the TNC.

Sending a Message to Another Station

A quick way to see the basic components of packet radio in action is to follow a simple message as it goes from one station to another, via an intermediate station.

Let us assume that an originating station, WØXI, wants to send the message “HELLO” to a destination station, KBØNYK. Also, assume that WØXI knows that his signal might not reach KBØNYK’s station directly, but that it can be picked up and repeated by an intermediate station, NØGRG, and then received by KBØNYK.

The following are the basic steps of a station-to-station packet communication, as seen from the point of view of the message, “HELLO,” sent from source station WØXI to destination station KBØNYK, through intermediate station NØGRG. Step 1 is one-time configuration task, Steps 2-3 establish a connection between stations, and Steps 4-13 follow the “HELLO” message from the time it is created by WØXI to when it is read by KBØNYK.

- Step 1. Before this message can be transmitted from a source station to a destination station, via an intermediate station, all the stations involved must have **callsigns**, which are used by the TNCs involved to identify and process information (e.g., source, destination, and routing of information).

So, we start by assuming each operator involved in this example has already performed this one-time configuration task, creating the callsigns given above. Stations are identified by reference to their callsigns. Note that a callsign can be for a TNC (e.g., KPC-3XE) or for a Personal Bulletin Board System (PBBS) “mailbox” that resides in RAM inside the TNC.

- Step 2. This communication begins when the operator of station WØXI turns on his computer, TNC (e.g., a KPC-3XE), and transceiver of his station and uses a computer software program to tell the TNC that he wants to establish a line of communication, called a “connection,” with a “destination” station KBØNYK using an intermediate station NØGRG as a repeater station to carry the information beyond the range of the originating station and into the range of the destination station.

- Step 3. The “connect” command with address given to the TNC causes the TNC to formulate a “request to connect” message that is sent from WØXI. If this message is detected by the specified intermediate station (i.e., NØGRG), repeated, and then detected by the destination station (i.e., KBØNYK), a connection may be established.

To confirm a connection, the destination station automatically sends a special message back through the reverse path, to the originating TNC.

When this “confirmation of connection” is processed by WØXI’s TNC, the TNC assumes that the operator wants to begin sending messages. So the TNC: (1) puts a message on the computer monitor confirming that WØXI is now connected, through intermediate station NØGRG, to KBØNYK, (2) turns ON the “connected” indicator on the TNC, which stays ON as long as the connection is in effect, and (3) switches itself from taking commands from the computer (i.e., TNC in “command mode”) to serving as a message-processor, sending and receiving messages to and from the connected stations (i.e., TNC in “conversation mode”).

- Step 4. Now that WØXI is in “conversation” mode, the operator types in the message, “HELLO”, and presses the ENTER key on the computer to

indicate the end of the message and the beginning of the message's journey.

- Step 5. The computer sends the message, "HELLO," to the TNC for processing.
- Step 6. The TNC receives the message, "HELLO", and takes charge of the situation, performing a number of complex operations: it uses the "connection" information about where the message is to go and the path it is to take and combines this with the message itself to assemble a "packet" of digital information for transmission from the transceiver.

The details of what is in the packet and how it is organized are fully defined by agreed-upon standards, so any machine that knows the "rules" can make use of the information.

The standard for amateur packet radio at this time is AX.25, so let us assume the packet assembled in the TNC is an AX.25 packet.

Technical note: All amateur radio packets use the AX.25 standard, but it is possible to use another standard or protocol (e.g., TCP/IP) "on top" of AX.25, in which case the packet is referred to by the name of the highest-level protocol.

- Step 7. After the TNC assembles the "HELLO" message into a packet of digital information in the TNC, it passes the packet on to a modem, which is also in the TNC, which transforms the information into a series of audio signals for serial transmission to station WØXI's transceiver.
- Step 8. The TNC orders the transceiver to send a message (i.e., keys push-to-talk) and then sends the audio version of the "packet" through the radio port of the TNC, up a cable to the microphone connector on station WØXI's transceiver, and to the transmitter.
- Step 9. The packet is transmitted by WØXI's transceiver as a burst of radio-frequency signals.
- Step 10. The "HELLO" packet, including address information (i.e., to, from, via) and other control information, is now on the air, where it may be correctly interpreted by all appropriately configured (i.e., AX.25 packet) stations that receive the signal.
- Step 11. KBØNYK is on the air and is running AX.25 packet, but it is too far away from WØXI to pick up the signal, so does not get the message as it is sent from WØXI.

Step 12. As we know from the “connection,” already established, intermediate station NØGRG, an AX.25 packet radio station, is on the air, and is close enough to receive the radio burst sent by WØXI. It receives the burst of radio signals from WØXI and passes the audio signals on to its TNC for processing. The TNC (e.g., a KPC-3XE) transforms the signal from audio to digital, analyzes the series of 1’s and 0’s in the incoming bit-stream, and identifies the burst as an AX.25 packet of information containing the message, “HELLO.”

Also, and most significantly, NØGRG registers that the packet designates NØGRG as the first (and only) intermediate station on a path between a source, WØXI, and a destination, KBØNYK. As a fully functioning AX.25 packet station, NØGRG knows what to do: it sends the message back out over the air (i.e., digipeats it), where it may or may not be received by its intended destination, KBØNYK. Before doing so, it marked its call in the packet to indicate to others that it had seen the packet and had sent it on its way.

Note: Since a “connection” has been established, we expect the message to get through, but each packet is processed separately, and each must make it on its own.

Step 13. As we know from the “connection,” already established, KBØNYK, an AX.25 packet radio station, is on the air, close enough to receive the RF burst sent by NØGRG, and currently set to display messages on a computer screen. So, the “HELLO” message re-transmitted by NØGRG arrives at KBØNYK’s TNC (e.g., a KPC-3XE), where it is processed and recognized as an AX.25 packet addressed to itself. Then KBØNYK’s TNC sends the message, “HELLO,” to the computer screen for KBØNYK to read. At the same time, KBØNYK’s TNC creates an acknowledgment packet to send back to WØXI’s TNC, reversing the address path it found in the “HELLO” packet. When KBØNYK’s TNC detects that the radio channel is clear, it keys the transmitter and sends the packet back over the air.

Step 14. NØGRG’s TNC now recognizes another packet which has its callsign in the address path, requesting it to be digipeated. NØGRG’s TNC does so as soon as it can.

Step 15. WØXI’s TNC now hears a packet addressed to it. Since the digipeated packet originally came from KBØNYK, to whom he is connected, and since the packet contains an acknowledgment for data that he had

recently sent, WØXI's TNC can rest easy knowing that the data he sent was properly received.

This concludes our close-up look at a single packet communication event, showing how the various parts of packet radio work with each other. To simplify and focus this example, various details and complications were ignored, but this still shows the basic steps involved in establishing a connection and sending the message "HELLO" from one person/station to another person/station.

This example covers one of many uses of packet radio. As covered in other sections of this manual, there are many ways to communicate with other stations.

Now, after having followed one message from one station to another, it is useful to learn about the "packets" in packet radio.

Protocols: Rules for Working Together

Packet radio, like any communication, is only possible if all parties involved (i.e., sender, receiver, and intermediate points on the route between them) agree on the rules for organizing information and the rules for dealing with all steps of the transmission. These rules are called “protocols,” or standards.

AX.25

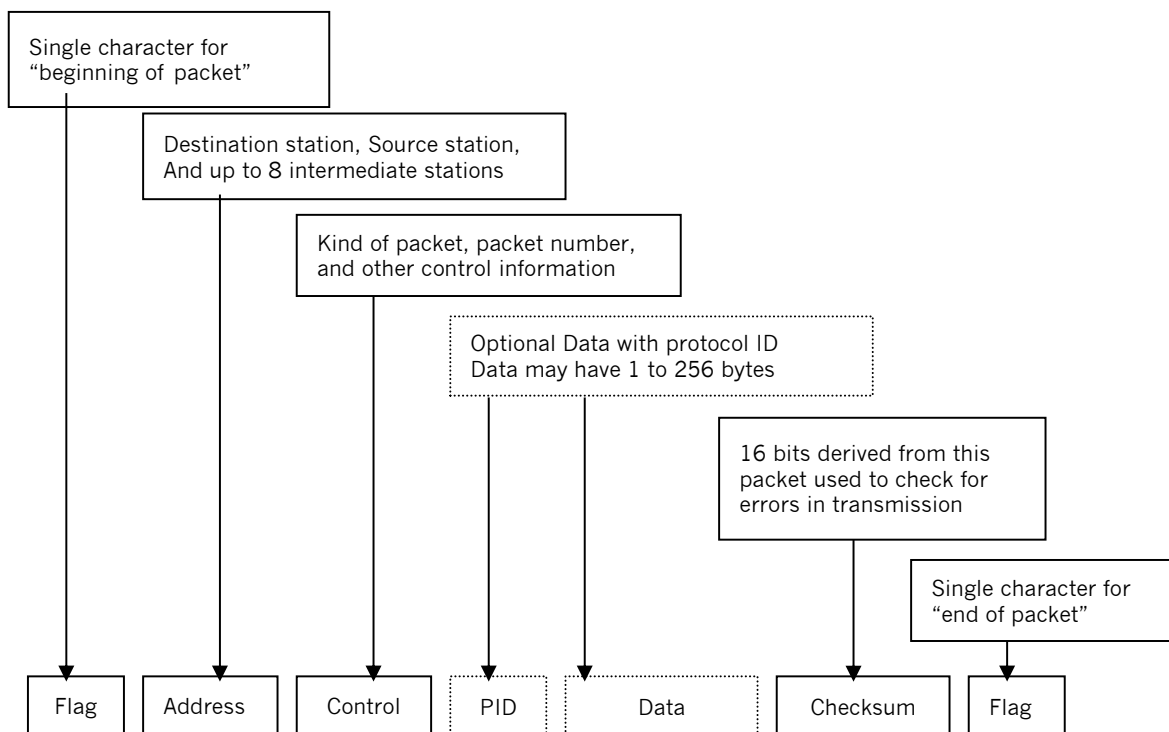
The standard protocol for amateur packet radio communication is AX.25 (level 2, version 2). For details on AX.25, see the ARRL publication, **AX.25 Amateur Packet-Radio Link-Layer Protocol**.

How an AX.25 Packet is Organized

A packet radio station takes care of all the details of working with packets, so it is not necessary to see the “inside” of packets. But it is useful to have at least a basic idea of how they are organized.

All amateur radio packets (also called “frames”) are defined by the AX.25 protocol, which is discussed in the next section. Packets used to carry messages, or chunks of messages, as in the “HELLO” example, are called “information packets.”

The following diagram shows the basic building blocks of “connected information” packets used in amateur packet radio:



Kinds of Packets

As shown in the above diagram, the "control" field in an AX.25 packet includes a code telling what kind of packet the current packet is. In addition to connected information packets, the AX.25 protocol also defines two other kinds of packets, each of which is identical in structure to the connected information packet: control packets (of which there are several kinds) and unnumbered packets. To learn more about these, see the documentation for the KPC-3XE ' MCOM command.

As noted at the beginning of this section, the organization of amateur radio packets is defined by the AX.25 protocol. Now it is useful to give a brief discussion of this protocol, and another that fits "within" it.

Packets: Dividing Messages into Segments

An efficient way to send a message from one place to another is to break the information into small segments, called packets or frames, and send each packet separately.

Note: The correct technical term is “data frame”, not “packet” or “frame,” but we will use the term “packet,” since that is familiar to most people and is common in the field.

Your packet radio station automatically divides your communication (e.g., a message or file) into small segments (see the PACLEN command for details) and puts each segment into its own “packet”. Each packet includes information used for addressing and error correction.

Unconnected Packets

Packets do not need to be addressed to any station. Packets transmitted without any established connection between sender and receiver are “unconnected” packets. “Unconnected” packets are just sent out without any back-and-forth coordination with any receiving station. Stations receiving these packets can check to see if the packets are correct (i.e., the same bit pattern is received as when they were sent), but they cannot request any retransmission of faulty packets. Nor can they request re-transmission of any packets in a series that are not received at all. This is because “unconnected” packets do not include sequence information that could be used to track them in order and because a TNC sending or receiving “unconnected” packets is not set up to do any communicating about the success or failure of its packet transmissions.

- **Note:** Packets sent as “unconnected” are also called UNPROTO (non-protocol), because the sender expects no response from the receiving station.

Connected Packets

Packet radio communication is often done between two stations that are connected. “Connected” communication using packet radio assures transmission with virtually 100% accuracy by having the sender and receiver follow a set of rules for communicating. These rules are specified in the AX.25 protocol.

The following is an overview of what happens during a packet radio communication between two connected stations.

After a connection has been established between two stations, information packets are transmitted one-by-one (or in small batches), with a sequence number as part of each packet, from the sending station to the receiving station. As the communication proceeds, the receiving station keeps track of which packets it has received error-free (as determined by using the check-sum that is part of each packet) and uses this to tell the sending station what packet(s) to send next.

Using these rules of communication, a packet radio connection between a sender and a receiver provides virtually error-free transmission of information, until the communication is ended by the parties, or until the sender “times out” after trying a number of times to get feedback and not succeeding.

- **Hint:** In most cases, communication by packet radio is like mail, not the telephone, in that it is not done in real-time, but via messages that are routed from one station, or “node,” to another.
- **Alternatives to AX.25**
- **FX25:** This is slight modification of the AX.25 protocol. It adds forward error correction to transmitted and received packets
- **TCP/IP and the KISS protocol:** The TCP/IP (Transmission Control Protocol/ Internet Protocol) suite of protocols require functionality not found in native AX.25 packets, so Phil Karn KA9Q defined a protocol called KISS that implements TCP/IP functionality for amateur radio. For further information on KISS see the KISS Mode section of the “Modes of Operation” chapter.
- **XKISS** is an extension of the KISS protocol, defined by John Wiseman G8BPQ.
- For further information on XKISS, see the XKISS Mode section of the “Modes of Operation” chapter.

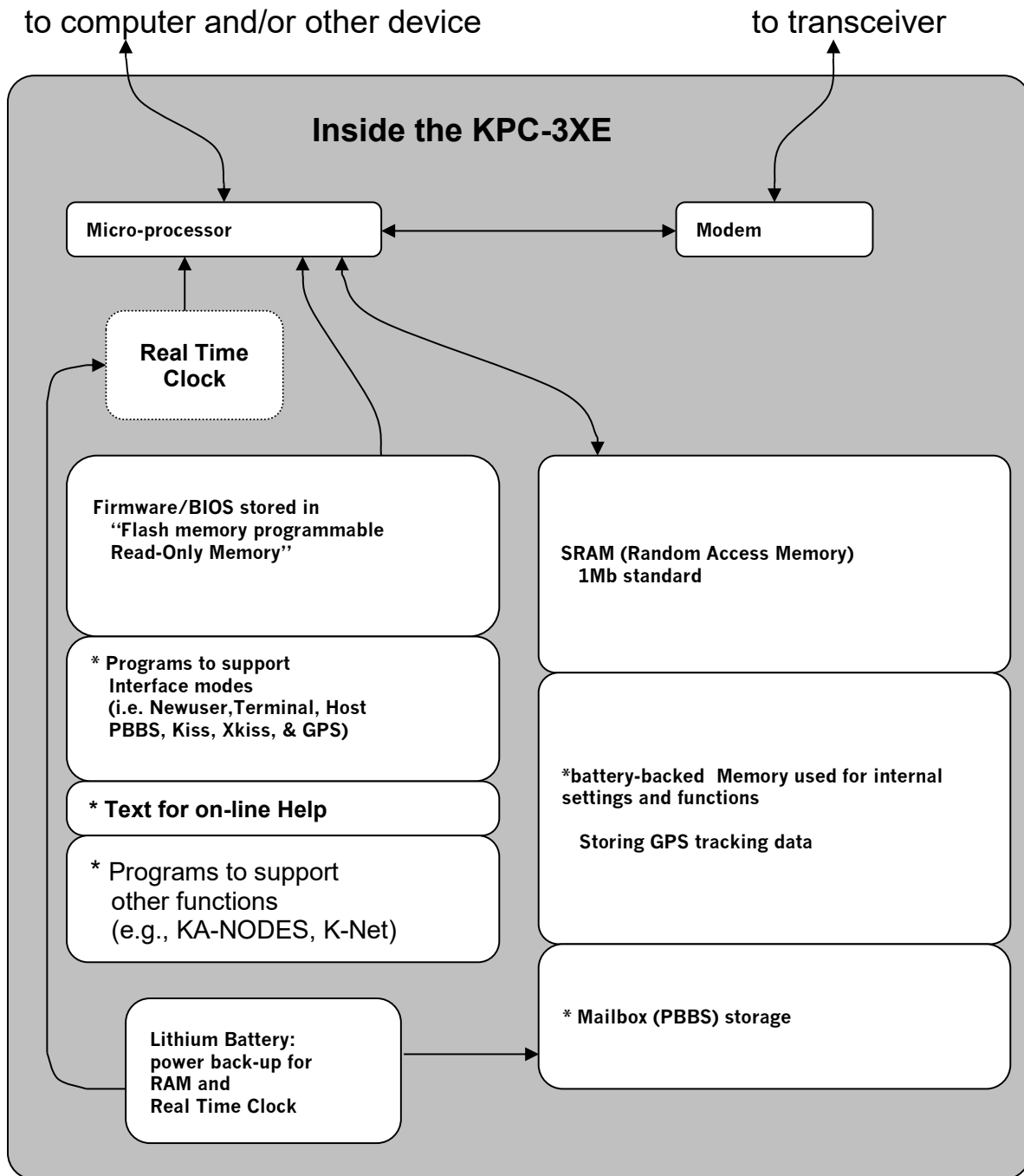
Inside a TNC – the KPC-3XE

To better understand the workings of a packet radio station, it is helpful to have an overview of the major components of the TNC device at the center of the packet station.

In the early days of packet radio, TNC's consisted primarily of a modem for communicating with a transceiver and a special-purpose microprocessor called a PAD (i.e., packet assembler/disassembler) used to process packets of information and communicates with a computer or terminal.

- As packet radio has developed, more and more features have been added, such as memory and software dedicated to a “Personal Bulletin Board System” within the TNC. The following diagram shows the most important internal components of a TNC (the KPC-3XE). The diagram is not to scale. For a detailed example, see the parts list and layout diagram for the KPC-3XE in the appendices of this manual.

Note: Other TNC models may have different features than the KPC-3XE. For example, the ability to operate two radios at once.



This concludes the introduction to packet radio. The next chapter steps through the installation of a KPC-3XE.

Installing the KPC-3XE;

The goal of this chapter is to be a quick guide through the steps of setting up a packet radio station.

The major steps involved in installing (i.e., connecting the parts) a packet radio station are as follows:

- connect KPC-3XE to a power source and confirm that the unit powers on when it should,
- connect the “Computer” port on the unit KPC-3XE to an appropriate port on a computer, install terminal communication software in the computer if necessary, and configure the terminal software to communicate properly with the unit KPC-3XE,
- connect the “Radio” port on the TNC to a transceiver and adjust the volume of the transceiver, so the TNC can receive signals from the transceiver for processing and can send signals to the transceiver for transmission,
- (optionally) use the RS232 radio port on the TNC to connect to a GPS device (or weather station device), to capture and beacon information. .

We start with a brief section on equipment needs for a packet station and an overview diagram that shows how the parts will be connected.

The Major Components of a Packet Station

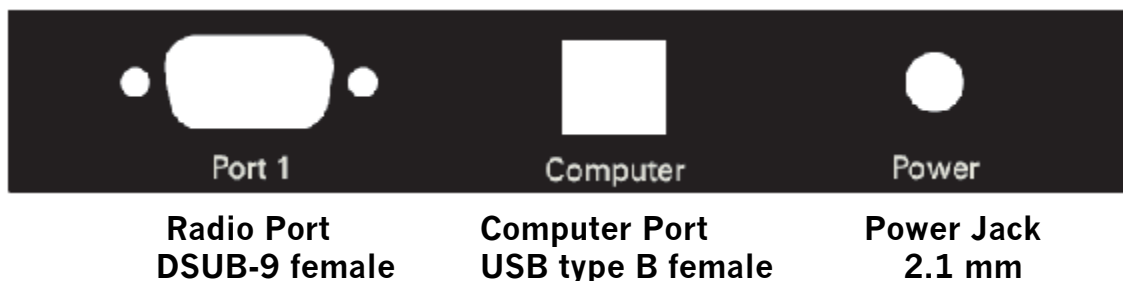
This section describes the basic components needed for a packet radio station and the connections between components that will be involved.

The KPC-3XE

setting up A packet radio station involves connecting other units (devices) to the TNC.

As shown below, the back of the unit has connectors to connect the KPC-3XE to your transceiver, your computer (or a GPS device), and a power source:

Back Panel



Radio Port	This DSUB-9 female connector accepts the cable from a radio, for 1200 baud packet operation.
Computer	This USB type-B female connector accepts the USB cable from a computer USB port.
Power jack (2.1 mm)	This connector is provided to apply external power (5 to 25 V dc). The center post is the positive connection, and the sleeve is negative (ground).

Note: All Kantronics' TNC's can operate without the computer or terminal being connected, once their commands and parameter settings have been configured. For example, the internal personal mailbox (PBBS) can receive and store messages and it can serve as a relay station (in digipeater mode) for other stations.

This independence from the computer is possible because the TNC contains the intelligence necessary to carry out these functions, once it is appropriately configured and attached to a transceiver.

The Transceiver

Most FM transceivers can be used for packet operations.

There are currently hundreds of models of FM transceivers. **Refer to the documentation for the transceiver to determine how it must be connected to external devices such as a TNC.**

Specifically, the following information is needed:

- what type of microphone connector to use with the transceiver, and
- which pins connect to which in the cabling between the TNC and the transceiver's microphone jack

Sample wiring diagrams for interfacing to some common transceivers are included later in this chapter.

The Computer

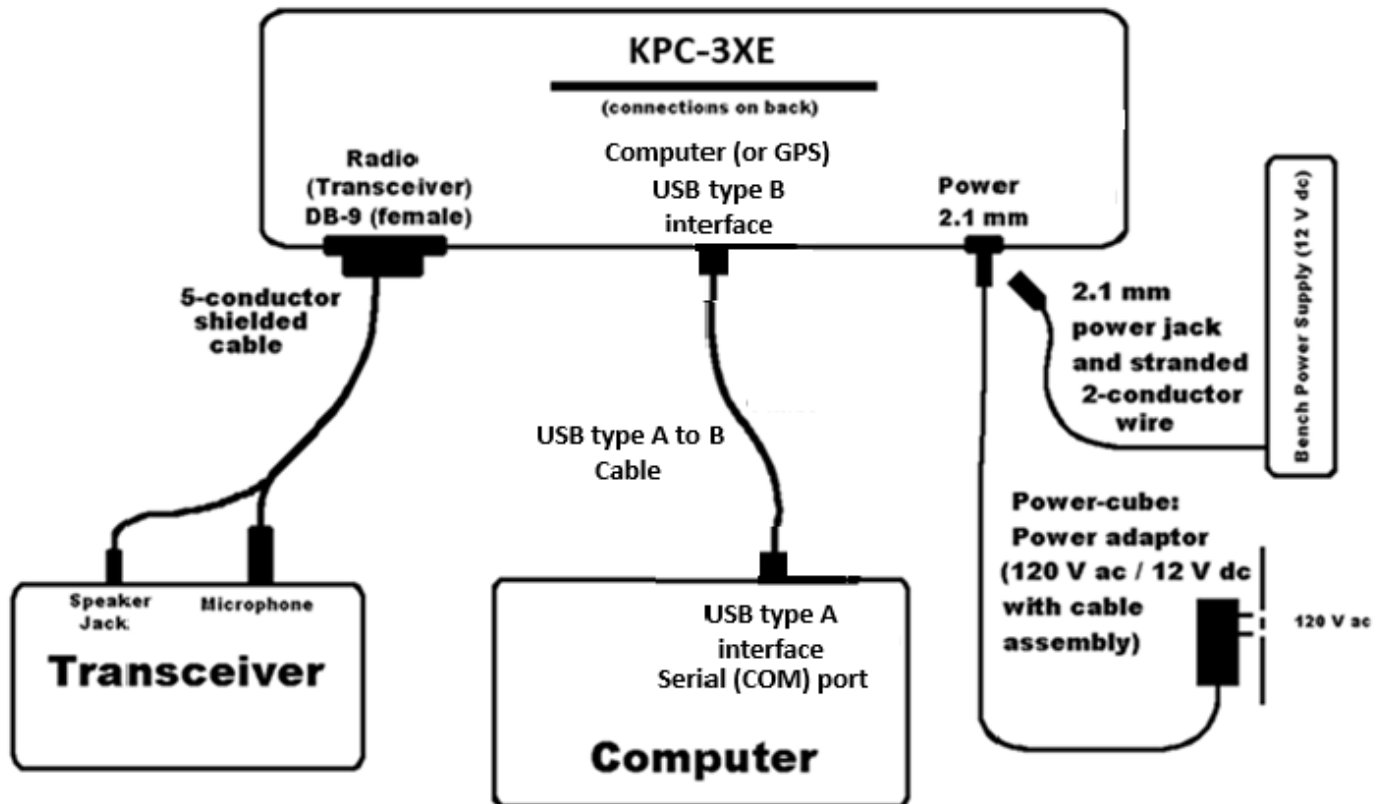
Any computer that has an available USB port can be used. The computer operating system will need to have (or can install) an appropriate driver for an FTDI USB interface chip. It will also need a terminal program that can be configured to communicate with USB connected devices.

GPS Device (Optional)

The KPC-3XE works with Global Positioning System (GPS) devices that (1) support the NMEA 0183 standard and (2) have a data port through which they communicate. The data output of a GPS device connects to the KPC-3XE through a pin on its "Radio" port.

How the Parts are Connected

The following diagram shows how the components of a packet radio station will be connected and the kinds of connectors that will be used. As the station is assembled, refer to this diagram.



Follow these step-by-step instructions for installing the packet station:

- Power input
- The interface cable assembly connecting to the external speaker and the microphone jacks, (or accessory jack) of a transceiver to the TNC
- (Optional) The connection between a GPS device, and an input pin on the TNC radio port.

Connect the KPC-3XE to a Power Source

As illustrated on the right side of the station cabling diagram, , supply power to the TNC through its power jack;

- connect to a regulated 12 Vdc power supply

- use external power from an AC to DC adapter
- See the Appendix for the allowed range of voltage input.

External Power

Use a regulated 12Vdc power supply:

If making a power cable, attach the supplied 2.1-mm power plug to a user-supplied 18-22 gauge stranded 2-conductor cable connect (by soldering) one wire of the two conductors in that cable, to the center (PLUS) lug of the plug. Connect (by soldering) the other wire in the cable to the barrel or sleeve lug of the power plug.

Then, with the power supply OFF attach the wire from the PLUS terminal of the 2.1mm power plug to the power supply PLUS (or +) terminal of the that power supply, and the wire from the ground (-) lug from the 2.1mm plug, to the “-“ (ground) terminal on the power supply. Turn the power supply ON.

Then, press the power switch on the front panel of the TNC to turn it ON.

Confirm that the TNC power indicator LED is lighted then press the TNCs power switch again to turn it OFF.

- If the power indicator (LED) does not go on (or OFF) as expected, check to be sure the wiring is correct. If the wiring has been carefully checked, and the TNC power indicator LED still does not light as expected, check or replace any fuses in line between the power supply and the TNC.

Caution: Do not exceed the TNCs maximum input voltage (see specifications).

If using a fuse in the positive lead between the supply and the TNC, do not use a fuse rated for 200-250 mA or less.

External Power from an adapter

(AC Mains input to DC output)

This adapter should plug in to an AC main and should provide a DC output voltage. That output should be close to 12 V dc. It should be able to supply more than 50mA. Be sure to verify the polarity of the power plug at the TNC (center positive), barrel negative (ground).

The steps are:

Connect the 2.1-mm plug from the adapter to the power jack of the TNC, and plug the power adapter into the AC mains. Press the power switch on the front panel of the TNC to turn it ON, and confirm that the power LED lights and then press the power switch to verify that the power LED goes OFF.

- If the power indicator (LED) does not go ON (or OFF) as expected, the power adapter be defective.

A suitable AC/DC adapter can be ordered through the SHOP section of Kantronics' web pages. Note that this adapter is for 120-V ac to 12-V dc 300-mA output. If the local mains AC voltage is different, an appropriate adapter will need to be acquired from a different source.

Connect the KPC-3XE to a Computer

The KPC communicates with a computer via a USB cable. This cable connects between the TNCs "Computer" port and a USB port on a computer. This is shown in the computer connection diagram.

The USB Cable

The computer interface cable needed to connect the TNC to a computer is included in the TNCs package. It is a common type, USB type A on its computer end, and USB type B on the modem (TNC) end. If a spare cable is needed, get the same type USB cable.

Terminal software

To communicate with this TNC, a basic or generic terminal communication program is required. The simple function of a terminal program, is to take characters entered on a keyboard and pass them through the configured communication port in the computer to a connected device (the TNC in this case, and display characters received from that device on the screen for user to view.

There are several terminal programs, which can be used in computers with various computer operating systems. Configuring a terminal program is usually

simple, such as setting the computer port to communicate through, setting its interface baud, and enabling hardware flow control (or RTS/CTS).

In the SHOP section of Kantronics' web pages, a simple (and very generic) terminal program for Windows systems, called K-Term, can be ordered (at no charge). When ordered, a download link will be provided (sent to the email address of the "purchaser"). It is an installable windows program. It may require the user to be an admin type, to allow its installation. Other terminal programs (most can be freely downloaded), are;

Terminalbpp (on a google sites user page)

RealTerm, Cutecom, and TerraTerm (can be found on the sourceforge website)

AUTOBAUD

The first time this TNC is powered on, it starts an AUTOBAUD routine. This is to help set it to match the interface baud that is set in the terminal program.

The AUTOBAUD routine in the TNC works as follows:

- Step 1. The TNC sends the message “PRESS (*) TO SET BAUD” over and over, stepping through each of its capable bauds.
- Step 2. Any time while this message is being sent, whether it is readable or garbled on the terminal screen, press the “*” key (or SHIFT+8 key combination), or the * on the number keypad on the keyboard.
- Step 3. When the TNC receives, the “*” character from the terminal, it uses that character to establish its interface baud to match the terminal program baud.

Technical note: The KPC will store that determined baud as the value of its ABAUD parameter setting. That setting will be retained in battery-backed internal memory as will all other changes to settings. So, whenever that TNC is powered on, it will start with that that interface baud. Later, that setting can be changed, if desired.

- Step 4. Finally, the TNC sends a message asking for a CALLSIGN, which will also be stored in the MYCALL setting and used until changed. At this point, the TNC is ready for user commands.

- **Note:** Do not confuse the computer interface baud (as stored in ABAUD) with the radio port packet baud (which is in its HBAUD setting).

Setting Basic Communication Parameters

There are several settings in addition to ABAUD (baud) and MYCALL (CALLSIGN) to which changes may be desired. Understanding these parameters and knowing how to reconfigure them fine-tune the system, to increase efficiency and performance.

Required TNC Parameter Settings

The following TNC parameter settings are needed (which are factory defaults):

- COMMAND = \$03 (CTRL+C) (default = \$03).

- CANLINE default = CTRL+X (HEX \$18)
- CANPAC default = CTRL+Y (HEX \$19)
- PASS default = CTRL+V (HEX \$16)

Optional Parameter Settings

The following TNC parameters can affect how data is displayed on terminal programs. These are default. If they have been changed to non-default values, data displayed on the terminal screen may be affected.

- ECHO = ON (default)
- FLOW = ON (default)

***Technical note:** Usually, ECHO and FLOW go together, both ON or both OFF. When using software that supports split screen displays, for example, set both ECHO and FLOW to OFF.*

- XFLOW = ON (default)

***Technical note:** XFLOW sets the use of whether software flow control. Its default setting is ON. If XFLOW is changed to OFF, the following parameters should also be set to 0 (zero): XON, XOFF, START, and STOP. When XFLOW is OFF, the TNC will not respond to software flow control characters from a terminal.*

- FILTER = OFF (default)
- 8BITCONV = ON (default)
- **Hint:** Also, the **MONITOR** parameter may be ON or OFF (its default is ON). Set Monitor to ON while using a terminal program, to view heard packets. But to prevent a build-up of data in an internal buffer in the TNC, MONITOR to OFF before exiting the terminal program **and** leaving the TNC ON. This is because when the TNC's receive buffer is full, it will return a **BUSY** message to other stations that try to connect.

Trouble-Shooting Difficulties in Communicating

When having difficulty establishing and maintaining communication between the terminal and the TNC, here are some points to consider.

Problem: Nothing Happens on the Terminal Screen

One possibility is that the current COM port that the terminal is set to use is not where the TNC is connected. The device on the selected port might be a mouse or an internal phone modem, for example. To see if it is a mouse, go to the TERMINAL Screen, start moving the mouse and see if sequences of characters appear as it is moved. If so, try a different COM port setting in the terminal program.

Another possibility may be that the TNC has already been programmed with a different baud than the one set in the terminal program. Try changing the interface baud in the terminal program.

If all attempts fail, it may be necessary to clear all the setting changes that have been made in the TNC. Either remove its internal memory backup battery for short period of time (with the unit powered OFF), OR perform the Hard Reset/self-test procedure. This will return the TNC to a “fresh out-of-the-box” state, where it can be initialized as a new unit.

Problem: the terminal displays bad data

If only garbled data is shown on the terminal, the most likely cause is a mismatch in baud settings between the terminal and the TNC.

Try other interface baud settings in the terminal or reset the TNC so it can be re-initialized.

Problem: the TNC Stops Behaving Normally

A terminal program is intended for use with a TNC in its TERMINAL Interface Mode (or, less often, in BBS, HOST, KISS, or NET Interface Mode), but the TNC can still output data in any Interface Mode. Garbled data, or partially garbled data may indicate the TNC has been switched to a different interface mode other than regular TERMINAL, each interface mode has a specific exit code. If a different program was used to communicate with the TNC (such as Winlink, or APRS) they may have the TNCs display settings, or its interface mode.

Again, if normal communication cannot be regained, a TNC Hard Reset may be required.

Connect to a Transceiver

The last connection needed is between this TNC and a transceiver. An interface cable can be user assembled or a ready-made cable can be purchased for

specific applications See the end of this chapter for sources. The SHOP section of Kantronics' web pages has a few of the common type TNC to radio interface cables.

This TNC to radio interface cable will connect transmit audio and PTT from the TNC to the microphone jack on the radio and will connect the receive audio signal from the speaker jack of the radio to the TNC.

As illustrated in the connection diagrams, this interface cable connects between the TNCs DB9 "Radio" port and radio interface jacks.

This section goes step-by-step, dealing with the following questions:

- What parts will be needed for the assembly?
- What are the exact wiring requirements for connecting a transceiver to the TNC ***and also*** by requirements that a specific make and model of transceiver may require (as determined by the documentation for that transceiver)?
- What are the steps used in constructing and connecting the cable assembly?

Parts required

- DSUB-9 connector (including hardware), to make the TNC end of the interface cable,
- A length of 5-conductor shielded cable (a length of approximately 3ft is adequate in most cases),
- A microphone connector plug as required for the transceiver (and) or external speaker plug (with attached wiring), depending on the receive audio output of the transceiver.

Preparing the Cable Assembly

This section gives instructions for how to assemble the cable.

Wiring Directions

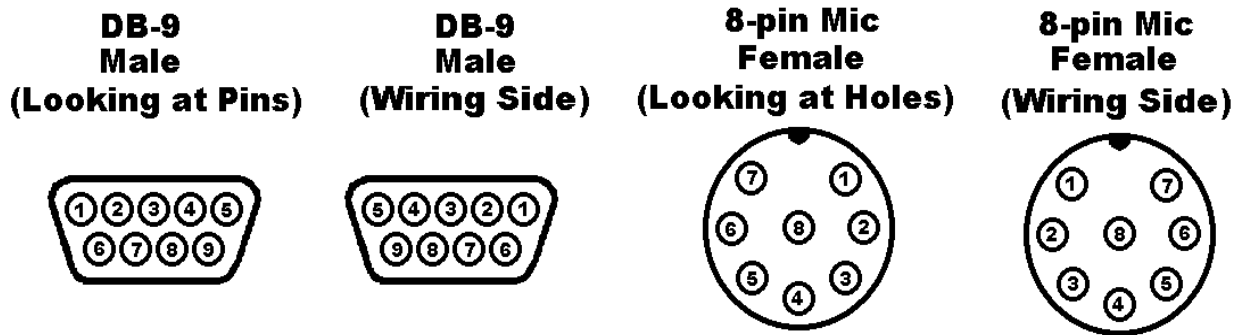
Determine which pins on the DSUB-9 connector plug will be wired to which connections on the transceiver's microphone connector and speaker jack.

- There are many different models of transceivers, each with their own exact requirements for how they are to be connected to other devices, such as TNCs. Radios from individual manufacturers typically have the same or similar jacks and pinouts between their models. But each different manufacturer typically will have different connectors and pinouts.

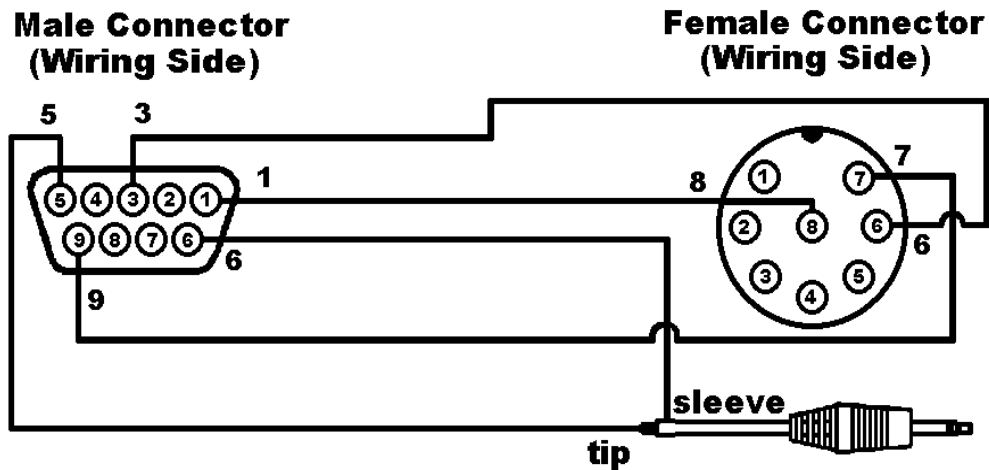
Note: If further assistance is needed about the radio end cabling, consult the radio manual, contact the radio manufacturer or the dealer where it was the radio was purchased, or a local amateur who has already made a similar interface cable.

The following diagrams, used for example only, show wiring connections between the TNC end of the cable, and some common transceivers (including HTs) from three major manufacturers: Yaesu, Icom, and Kenwood.

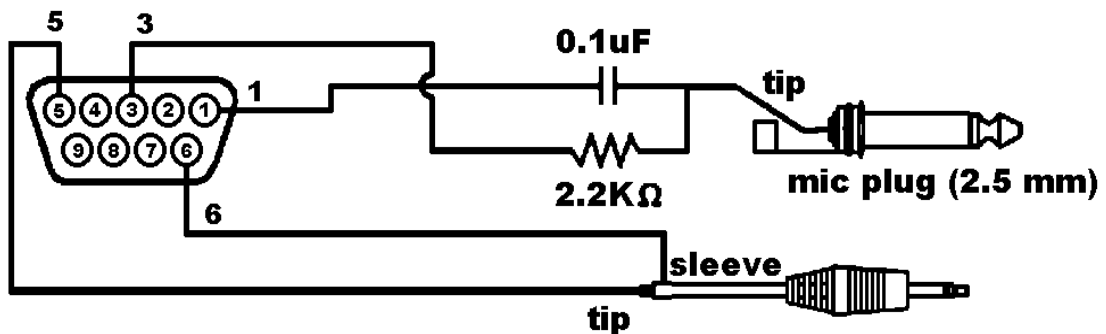
- These diagrams may also apply to transceivers from other manufacturers. See the section on “Interfacing Hand-Held Radios” in the “Advanced Installation” appendix for further information. For the pinout information for other types of radio connectors (such as accessory jacks), refer to the radio manual.



Yaesu Wiring



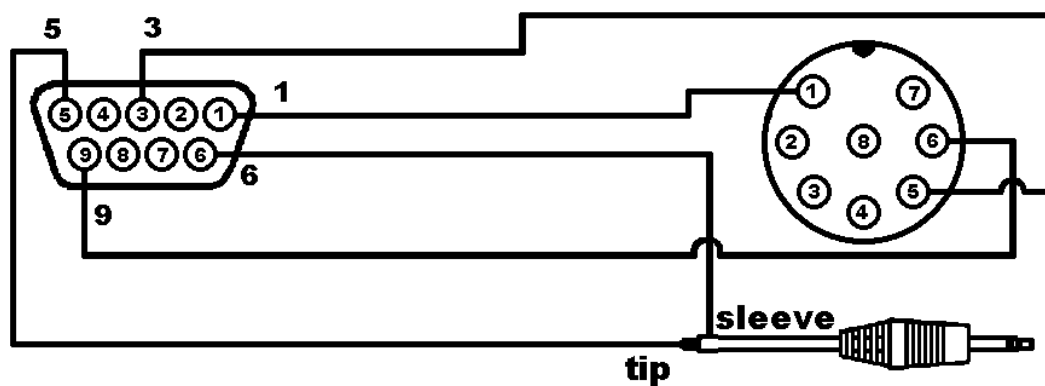
Yaesu Hand-Held Radios



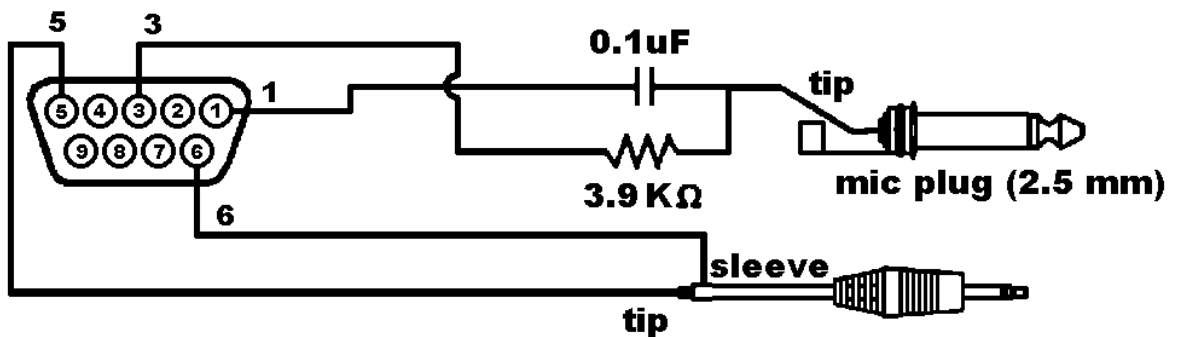
Icom Wiring

**Male Connector
(Wiring Side)**

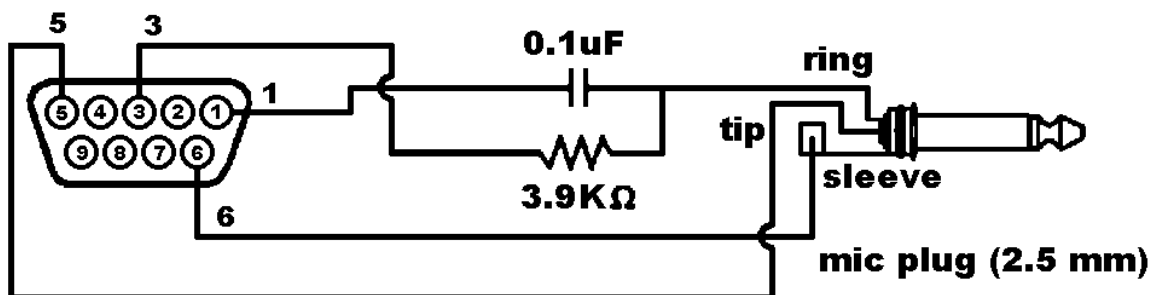
**Female Connector
(Wiring Side)**



2 AT Style Hand-Held Radios



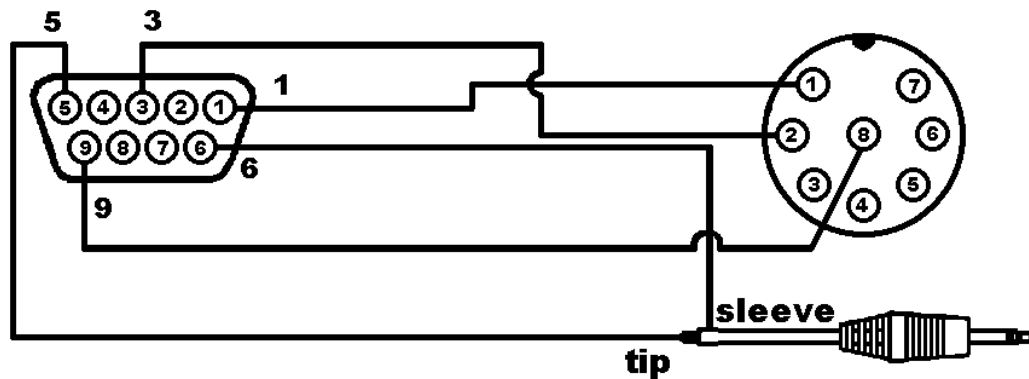
W2A Style Hand-Held Radios



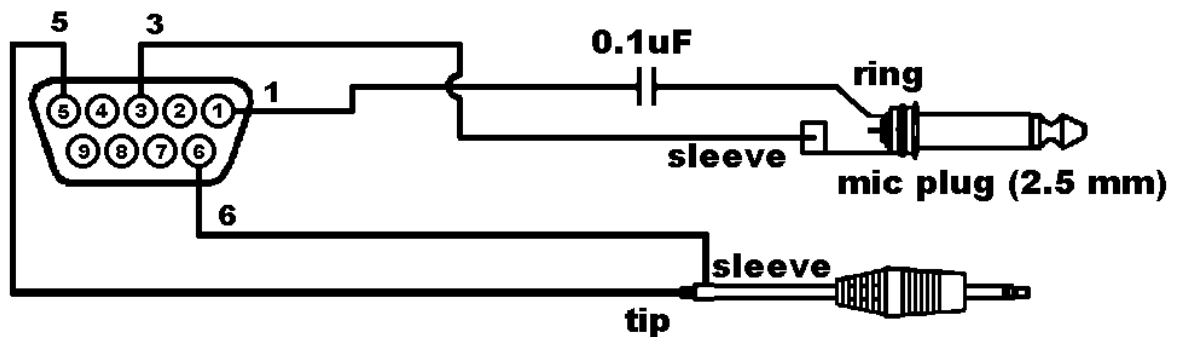
Kenwood Wiring

**Male Connector
(Wiring Side)**

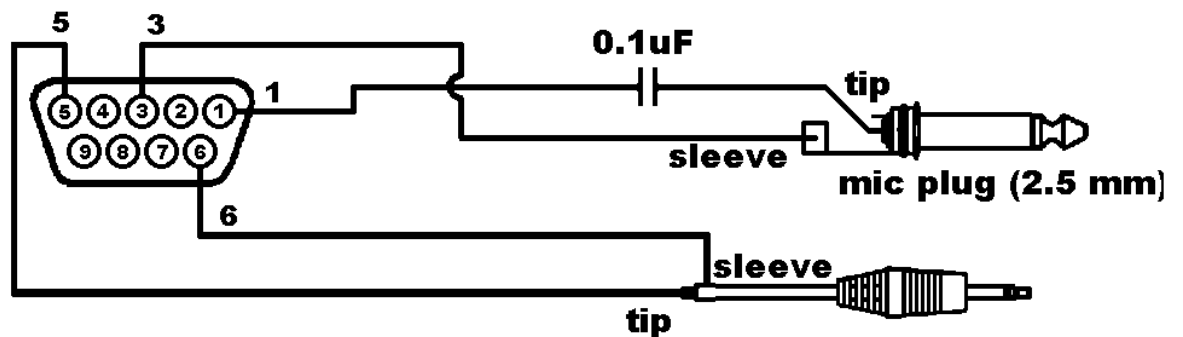
**Female Connector
(Wiring Side)**



TR-2600 and later Hand-Held Radios



TR-2500 Hand-Held Radios



A Transceiver's Microphone Connection

For the microphone on a transceiver, make notes of the following connections, which will be used, for wiring the cable assembly:

- identify and make a note of the microphone (transmit) audio input (which will be wired to pin 1, **transmit audio output** on the TNCs DSUB-9 port,
- identify and make a note of the radios PTT connection (which will be wired to pin 3, **Push-to-Talk output** on the TNCs DSUB-9 port,
- for base or mobile (but not hand-held) transceivers, identify and make a note of the radios ground connection which will be wired to pin 6 on the TNCs DSUB-9 port.

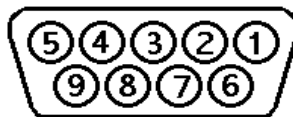
A Transceiver's External Speaker Jack

The external speaker jack on most radios is “mono” (2 conductor). The TIP of a plug for that jack is usually the receiver output audio to a speaker. The barrel of that plug is ground.

- The tip of the external speaker plug), will be wired to pin 5, **receive audio input** on the TNCs DSUB-9 port,
- The ground connection on the external speaker plug will be wired to pin 6 (or pin 9 if that TNC pin is configured as ground) on the TNCs DSUB-9 port.

Constructing the Cable Assembly

The diagram below, shows the wiring (solder) side of the male DSUB-9 connector, the pins are numbered from **upper right** (pin1) to **lower left** (pin 9).



This is the same as when looking at the DSUB-9 jack on the back panel of the TNC.

Similarly, when working with a typical 8-pin Mic connector where the pin numbers are known, consider which way that connector is being viewed (looking at

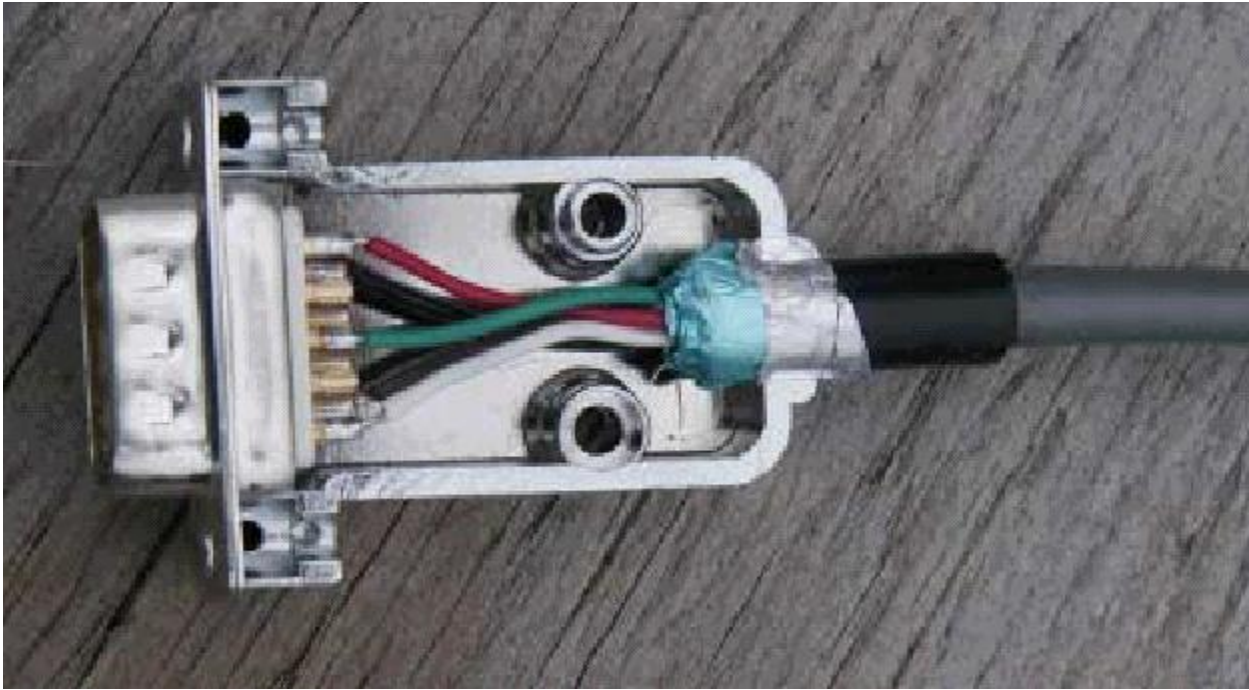
pins/sockets, or looking at the wiring side of the plug, where the wires are connected).

- **Hint:** Make notes of which wires (colors, cables) are used for each connection, to keep track of which connection is which.
- It may be easier to solder wires to the male DSUB-9 connector if it is first plugged into the TNCs DSUB-9 jack. This may also keep the male pins straight if too much heat is applied.

A shielded cable must be used with the shield properly terminated 360° to the connector. This helps to keep electrical noise and RF interference out of the cable.

Instructions for making a cable assembly using foil shielded cable and metalized plastic back shell follow:

1. Strip cable outer covering back enough to expose enough metalized plastic foil. Do not cut or rip foil.
2. Carefully loosen the drain wire and foil from around the insulated wires.
3. The diameter of the cable may need extra thickness where it enters the back shell, with any type of tape or sleeve, so that when the metalized plastic shell halves are screwed (clamped) together they will hold the cable as strain relief and press against the metal foil.
4. Connect wires as needed to terminals of connector. Keep track of which wire is connected to which terminal.
5. Carefully fold the metalized foil back over the cable and Z-fold it so the metal side of the foil is exposed to the connector back shell at the entrance area. Spread the foil over the cable to get as much coverage as possible (this may not result in 360° coverage but should be sufficient).
6. Place the drain wire in the gap where there is no foil coverage.
7. Place the metalized plastic shell halves together over the connector and cable, with the (2) two long half-threaded anchor screws and two shaped washers,
8. Screw the two halves together. There should be enough compression to hold the cable in place and there should be solid contact between the metal side of the foil and the metalized back shell. See the figure below.



Wiring to a Transceiver Microphone:

Follow the notes taken above and wire the cable assembly in the following order:

Step 1. Using a microphone connector appropriate for the transceiver and the notes taken above, attach wires as follows:

- Attach the wire from pin 1 (transmit audio) on the DSUB-9 to the microphone connector's mic audio input pin,
- Attach the wire from pin 3 (push-to-talk) on the DSUB-9 to the microphone connector's PTT or STBY pin,
- (Optional) for base or mobile units, attach the wire from pin 9 (one path to ground) on the DSUB-9 to the ground pin on the microphone connector. Assuming the jumper inside the TNC has been set so that pin is a ground pin.

Connecting the Transceiver's Speaker Plug

Use an appropriate third-party speaker plug and the notes taken earlier to do the following wiring to the DSUB-9 cable assembly:

- Step 1. Locate the wire that connects to the receiver audio of the external speaker plug (usually the plug tip) and attach it to the wire from pin 5 on the DSUB-9 (Receive Audio)
- Step 2. Locate the wire that connects to the ground of the external speaker jack (usually the plug barrel) and attach it to the wire from pin 6 on the DSUB-9 (Ground)

This is all that is needed to do to wire the speaker plug part of the assembly.

Important: Keep the wiring notes for future reference.

This completes the construction of the TNC to transceiver cable assembly.

Connecting the Transceiver Cable Assembly

Use this cable assembly to finish assembling the packet radio station.

- Step 1. Check to be sure all the parts of the completed cable assembly are properly fastened and ready for connection.
- Step 2. Turn OFF all power.
- Step 3. Plug the male DSUB-9 connector on the cable into the female DSUB-9 connector on the TNCs "Radio" port,
- Step 4. Secure the male DSUB-9 connector on the cable to the female DSUB-9 connector on the TNC, with the screws in the db9 shell, to the threaded nuts on each side of the TNCs "Radio" port so that it is securely fastened
- Step 5. Plug both the microphone connector and the external speaker jack into the transceiver.

Adjusting the Radio Receive Volume

Adjust the receive volume of the transceiver to an adequate output level (or to confirm that it is already set at the proper level), do the following:

- Step 1. Turn ON the computer (with your transceiver OFF),
- Step 2. Start the terminal program,
- Step 3. Turn ON the TNC,

- Step 4. If not already done, initialize the TNC (setting its baud and callsign).
- Step 5. Turn ON the transceiver and open its squelch control (turn it fully counterclockwise, or wide-open),
- Step 6. Slowly turn the transceiver's receive volume up until the RCV LED (receive indicator) on the TNC is lighted.
- Step 7. Increase the radios receive volume control slightly above that point.
- Step 8. Slowly turn the transceiver's squelch control clockwise until the TNC's RCV LED is no longer lighted.

This is a proper adjustment, to allow the TNC to "hear" packets and be able to transmit when the radio channel is not busy.

The TNC is now installed and ready to use. See the "Getting Started" chapter, to learn how to do the basic operations with this TNC.

- If the TNC receive LED does not light, check the cabling between the radio and the TNC. Also, until the TNC has been initialized from the terminal, it will not show ANY indication of receiving.
- If receiver audio is connected to a fixed level (un-squelched) audio from the radio, the TNCs receive LED may remain on all the time. In this case, set the TNC parameter CD to SOFTWARE. Then, the receive LED will not light until valid packets are being heard. The TNC cannot transmit while it's receive LED is lighted. (see CD in the commands section)

Transmit level adjustment

The 1200-baud data drive level (transmit audio voltage, TXA) from this TNC to a radio is adjusted digitally, using the terminal keyboard and the transmit level XMITLVL command or the calibrate (CAL) command.

The TNCs transmit audio drive level should be adjusted to result in about 2.5 to 3.5 kHz of deviation of the radios transmit signal. A deviation meter can be used to get an accurate level adjustment but is not necessary. The level can be adjusted so that this TNCs transmitted packet audio (as heard through another receiver) sounds about the same as packets heard from other stations.

The XMITLVL command has a parameter default value of 100; this corresponds to a drive voltage of about 50 mV. It may be changed by manually entering a higher or lower number in the TNCs XMITLVL command.

- For details, see the description of the XMITLVL command in the “Command Reference.”

Alternatively, the calibrate function (the CAL command) can be used. While in Calibrate mode, the XMITLVL can be manually adjusted with the ‘-’ or ‘+’ key on the terminal keyboard. During the transmit level adjustments, numbers below 256 increase or decrease in 0.5 mV steps and for numbers above 256, the drive increases or decreases in roughly 15 mV per step. See the commands section for more detail on these commands, and the Calibration/Equalization section for the Calibrate function.

Sources for “ready-made” TNC to radio interface cables;

The SHOP section on Kantronics’ web pages
(www.shop.kantronics.com/cables/),

Associated Radio in Overland Park, Kansas (www.associatedradio.com),
This dealer can make some custom cable assemblies to order.

Connecting a GPS Device (Optional)

This TNC has a command called GPSPORT. When this command has been enabled, Pin 2 on the TNCs radio port is used as an input for data from a GPS device (or a weather station).

Since interface cabling may vary from one GPS vendor to another, the correct data output cable may need to be acquired from the GPS device manufacturer, or from another source.

To connect the GPS receiver and this TNC as a position reporting system, 2 wires will need to be connected for data flow from the GPS receiver to the TNC. Connect the GPS receiver “DATA OUTPUT” to the TNCs DSUB-9 radio port XCD (pin 2). Connect the GPS receiver “SIGNAL GROUND” to ground (pin 6 or 9) on the TNC. Some GPS receivers may require other pins to be connected, check its documentation.

The GPS data input on this TNC is enabled with its TNC command GPSPORT.

See the GPS chapter in this manual for details on using this TNC with a GPS (or weather station) device.

- Note: some weather station devices have data output similar to that of GPS devices. The configuration of this TNC and connection to the device is the same.
- For more information on installation topics, see the appendices.

Getting Started

This chapter supplies important information, and examples to get started using a packet radio station.

We assume that the packet station has been installed and operating, And the user is familiar with its use and capabilities.

If the station has not yet been installed, that should be done before continuing with this chapter.

Material covered here includes diagrams, explanations, and mini-tutorials about:

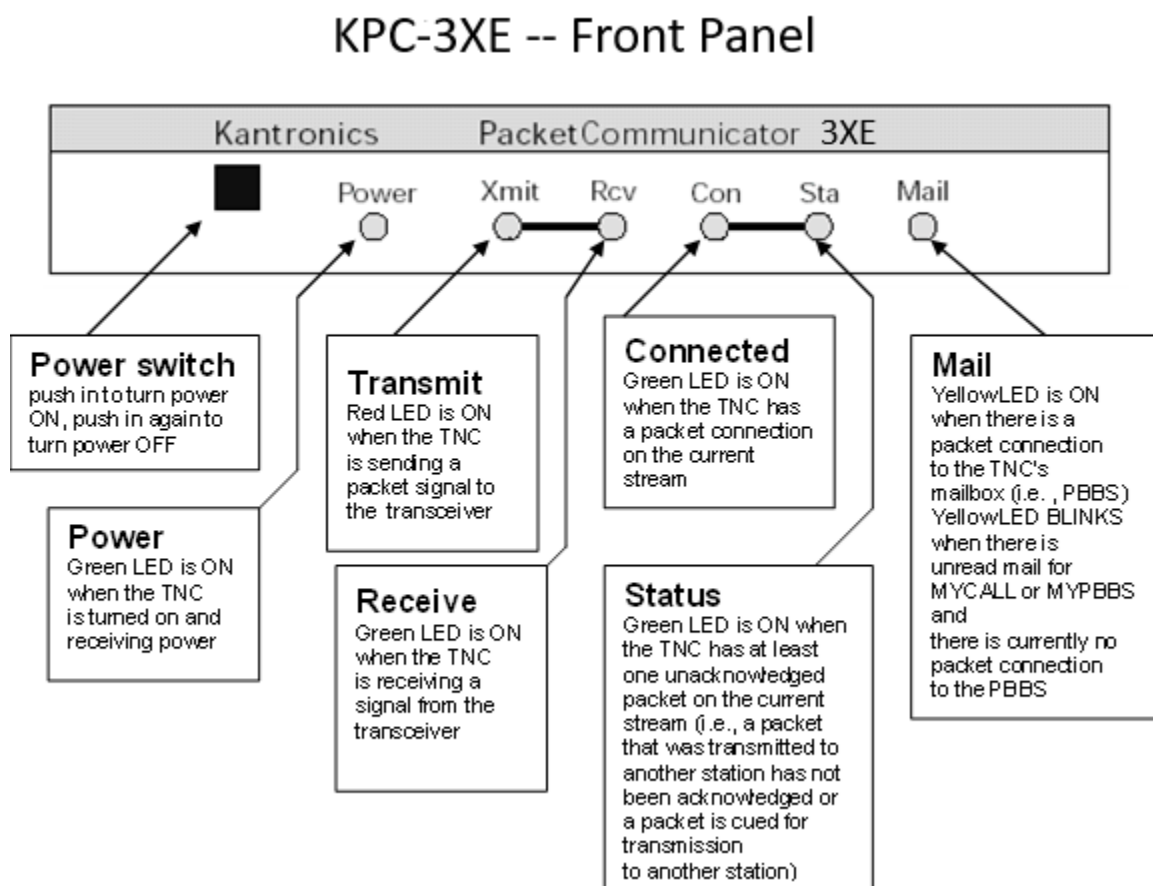
- the front panel of this TNC,
 - beginning a session,
 - issuing commands and transmitting data,
 - using the Personal Bulletin Board System in this TNC,
 - monitoring transmissions from nearby packet radio stations, and
 - connecting to another station and initiating a conversation.
- Reading this chapter, (or just skimming the material), will help when using the following chapters, about the TNCs functions, and different modes of its use., The “Command Reference” chapter, and the appendices have a lot of relevant information.

The Front Panel of the KPC-3XE

Use of the TNCs POWER switch to turn power on and off has been covered in previous sections.

This is a good time to get familiar with the front panel indicators.

The following diagram shows the placement and function of the controls and indicators on the front panel, as well as a brief explanation of each:



Beginning a Packet Session

Start the terminal program and turn on the TNC.

Once the TNCs initial setup is completed (sending * to set the interface baud, and entering a CALLSIGN, the TNC will first display its sign-on message.

```
KANTRONICS PACKET COMMUNICATOR KPC3XE VERSION x.xxxxx  
(C) COPYRIGHT 2025 BY KANTRONICS INC. ALL RIGHTS RESERVED.  
DUPLICATION PROHIBITED WITHOUT PERMISSION OF KANTRONICS.  
cmd:
```

The currently installed bios version will be displayed.

The last line of this message (“cmd:”) is the TNCs report (prompt) that it is now in “Command” mode, which means that it is now expecting information from the terminal to be commands for it to interpret and carry out. Before learning about the specific commands you can send to the TNC, it is important to keep in mind that there is a difference between sending commands and giving entering data that is to be transmitted.

Giving Commands and Transmitting Data

The TNC will react to data received from the terminal program either as a command to follow or as information to transmit.

Command Mode

When in the “COMMAND” mode the TNC will interpret data received from the terminal as a command to process, not as data to transmit. The TNC must be in its command mode (displaying its “cmd:” prompt) to make any parameters changes or to connect, to disconnect, or initiate a function.

Converse (Conversation) Mode

When in “CONVERS” (conversation) mode, the TNC will interpret data received from the terminal as data to be transmitted. Some certain pre-defined characters (which the user can change) are recognized as control characters, rather than as data to be transmitted. For example, a “backspace” character causes a backspace (back over and delete the previously entered character) and is not transmitted.

- The TNC will automatically switch to the CONVERS mode when a packet connect to another station has been established. It will return to its COMMAND mode when that connected link has ended (disconnected).
- When in command mode, the TNC can be switched to CONVERS mode just by giving the command K (or CONVERS).
- When the TNC is in the CONVERS mode, it can be returned to the COMMAND mode, by sending a <Ctrl+C>.

TRANS (Transparent) Mode

A second way to transmit data, such as including “control characters” (e.g., “backspace”) and just transmit every character as data, is to use transparent mode (“TRANS”) of communication.

For example, if data sent to the TNC for transmission in TRANS mode includes “backspace” characters (i.e., characters that would cause a backspace in CONVERS mode), they will be transmitted as entered.

- To switch to Transparent mode, just enter T or TRANS at the TNCs cmd: prompt.
- To exit TRANSPARENT mode and return to COMMAND mode, enter the “<Ctrl+C>” characters three times quickly, with a pause of less than one second between each character.

A Sample Connect

A good first packet connection, is to simply connect to the TNCs internal mailbox (PBBS):

- Since this connect is to the internal PBBS, there is no transmission and a radio does not need to be connected.

Step 1. Go to COMMAND mode (the “cmd:” prompt is showing). If not already there, send <ctrl-c>.

Step 2. Type CONNECT call, where “call” is the callsign of the internal PBBS (as set in the command MYPBBS, and press the ENTER (return) key. By

default, MYPBBS is the callsign entered right after the autobaud routine, with "-1" appended to it.

Note: To see the current mailbox callsign, type MYPBBS at the cmd: prompt, and press the ENTER (return) key. The callsign shown is the one to use when connecting to the internal mailbox (PBBS)

- Step 3. A connection to the mailbox will be established. The terminal should show something like this:

```
cmd: CONNECT NØKN-1
```

```
cmd:*** CONNECTED to NØKN-1  
[KPC3XE 1.0 HM$]  
102000 BYTES AVAILABLE IN 2 BLOCKS  
ENTER COMMAND: B,J,K,L,R,S, or Help>
```

- Step 4. The mailbox will accept any of the commands shown (B,J,K,L,R,S, or Help). Read the "PBBS (Personal Mailbox) section of the "Modes of Operation" chapter for more details. Now type the HELP command and press return, the mailbox information will be returned and then the standard mailbox prompt:

```
ENTER COMMAND: B,J,K,L,R,S, or Help> .
```

- Step 5. Now disconnect by sending the Bye command, and the mailbox will disconnect.

Or, return to command mode by sending <Ctrl+C> on the terminal keyboard.

The TNCs "cmd:" prompt should now be displayed.

Next type the DISCONNE command and press return.

The message *** DISCONNECTED will be displayed, then a new TNC command prompt (cmd:).

Congratulations. This is a completed packet connect session.

Monitor Communications From Nearby Stations

A good way to start using a packet radio station is to monitor (see) packet communications from nearby stations.

- Amateur packet-radio activity in the US is typically on the VHF FM band, on 2 meters, with 145.010 MHz being the most common packet-radio frequency. The most common packet baud is 1200.

To monitor packet communications from nearby stations (after having tuned in to an appropriate packet channel):

- Step 1. Disconnect from any other station (or mailbox).
- Step 2. Set the command MONITOR to ON (if it is not already ON).
- Step 3. As packets are received, they will be displayed on the terminal. A typical received packet might look like this:

```
NØKN>KBØNYK:  
Hi Mike. How are you today?
```

The first line is the header line. It shows the callsigns of the stations talking and the following line(s) are the actual message that the first station sent. In the example, NØKN is the transmitting station, and KBØNYK is the receiving station. The TNCs RCV LED should light when packet signals are being received, but sometimes nothing is displayed on the terminal screen. This is normal and may be the result of the settings of some of the monitoring commands, which are discussed in detail in the Command Reference.

Monitoring is a simple way to learn about other packet stations nearby. It should be possible to establish a packet connection with one of them.

Communicate Directly with a Nearby Station

Now, try to have a two-way conversation with another station. The following is what to do to connect to and converse with another station.

For example, to connect to the station callsign KBØNYK;

- Step 1. be sure the TNC is in its command mode. (the cmd: prompt is showing).
- Step 2. Now, send the command CONNECT KBØNYK and press return, same as in the sample connect to the mailbox.
- Step 3. Once the connection has been made, a message will be returned to the terminal screen that says *** CONNECTED to KBØNYK

- Step 4. The TNC should now be in its conversational mode and it is ready to start exchanging packets with that station.
- Step 5. When that conversation has completed, return to the command mode ("cmd:" displayed), and instruct the TNC to disconnect. This is the same as when ending the link with the TNCs mailbox, as shown in the sample connect previously. Enter the disconnect command D or DISCONN.

This concludes the quick tour of basic uses and operation of this TNC.

Modes of Operation

This chapter covers more ways in which this TNC can be used. For specific details on this TNCs commands, see the “Command Reference” chapter.

Packet Mode of Operation

This section adds to the information presented earlier (in the Getting Started and other sections) on how packet radio uses digital information for communication.

By becoming familiar with the concepts presented here, customization of commands and parameter settings, and advanced operations can be utilized.

Introduction

Information is Organized into “Packets”

A packet is a group of characters with a flag and header at the beginning and a checksum and flag at the end. A flag is a specific character used to signify the beginning and ending of a packet. The header is information indicating who the packet is from, who it is to, any relay stations needed to get to the destination and some control information. A checksum is a complicated mathematical formula that produces a number based on the combination of characters that are in the packet. This number is recalculated by every station that receives the packet, and if it does not match the number that is in the packet, the packet is thrown away, thus near error-free communications. A packet is also called a frame.

This unit is a TNC (Terminal Node Controller)

Packet radio modems, or packet modems, are generally referred to as TNCs. This “label” or “moniker” was adopted when the Tucson Area Packet Radio Group (TAPR) developed their first “TNC-1” packet radio modem kit in the early 1980s. The TNC is the workhorse of packet radio. As a listening device it hears an audio signal from the radio, changes the data to digital form, determines if the data is a good packet and sends it to whatever device is attached, usually a computer. As a relay device it also checks the packets it receives and determines if the packets need to be resent, then does so if appropriate. As a sending device it receives digital data from the computer, packetizes it and changes it into audio tones, which are sent out to the radio. The rule the TNC uses to do all of this is called a protocol.

Protocol for Amateur Packet Radio: AX.25

The most commonly used protocol in amateur packet radio is AX.25. The details of the inner workings can be found in a book titled “AX.25 Amateur Packet-Radio Link-Layer Protocol”, which is available from the ARRL. Most TNC users will not need to or care to go into the details. The TNC takes care of this communications protocol.

Protocol for Amateur Packet Radio: FX.25

FX.25 is intended to complement the AX.25 protocol, not replace it. It provides an encapsulation mechanism that does not alter the AX.25 data or functionalities. An error correction capability is introduced at the bottom of Layer 2 in the OSI model.

The AX.25 Link Layer Protocol is extensively used in amateur radio communications. The packets are validated by a 16-bit CRC and are discarded if one or more errors are detected. In many cases, such as space-to-earth telemetry, the packets are broadcast unidirectionally. No back-channel may be available to request retransmission of errored elements. Consequently, AX.25 links are inherently intolerant of errors.

The FX.25 protocol extension provides an error correction "wrapper" around the AX.25 packet, allowing for removal of errors at the receiving end. Data fields have been carefully chosen to allow reception of the AX.25 packet data within an FX.25 frame by a non-FEC decoder.

Command Mode

In order to change parameters, or give any other instructions to the TNC, it must be in Command Mode. This is the mode it will be in when it is first turned on, or after it has been reset (a soft reset).

A parameter in this TNC, called COMMAND, determines what special character it will recognize to return to its Command Mode. Its default is <Ctrl+C> (i.e., while holding down the control key, press “C”, then release both keys).

When in Command Mode, the TNC will send a prompt to the terminal screen that looks like this:

cmd:

Connected vs. Unproto

There are two ways to send data in packet radio, connected or unproto (unconnected). The Connected Mode is used when first establishing a packet connection. Then the TNC will send packets to that connected station and expects acknowledgments in return.

If an acknowledgment is not received, the TNC will resend the data (depending on the setting of AX25L2V2, it may send a poll first). The RETRY parameter setting will determine how many times this is done before the connection is lost due to bad conditions.

If the acknowledgment is received, the TNC is happy and will send more data when available. Therefore, the Connected Mode, barring impossible conditions, assures that the station to which it has a connection will receive everything this TNC transmitted, and in the order sent.

In the Unproto Mode, when this TNC sends a packet, no acknowledgment is expected (or required) and no retries are attempted. This mode is often used for calling CQ, for transmitting beacons, and for informal round table chats.

Monitoring and Calling CQ

If the MONITOR command is set to ON, packets heard from other stations will be displayed on the terminal screen. There are two callsigns at the beginning of each displayed packet separated by a ">". The first callsign is the station the packet is from (who transmitted the packet), and the second callsign is the station to which the packet is intended.

An Unproto packet may have a name or CQ for the second callsign. In this case, that second callsign makes no difference, and may be for informational purposes only.

The UNPROTO command is used to set what will be seen as the "to" callsign (the second callsign in the header) for Unproto packets. Its default is simply CQ.

In order to transmit a simple CQ, the TNC must be in its Convers Mode. At the cmd: prompt, enter K or CONV. Then, anything sent from the terminal to the TNC will be transmitted in an UNPROTO type packet.

Remember to exit convers mode and get back to Command mode, enter a <Ctrl+C>.

A Simple Connect

Starting from Command Mode, issue a connect request. For example:

```
C callsign<CR>
```

where 'C' is short for CONNECT and 'callsign' is the callsign of the station to connect to. <CR> is a carriage return, or the ENTER key.

If for any reason the connection fails, the TNC will return the following message to the terminal:

```
*** RETRY COUNT EXCEEDED  
*** DISCONNECTED
```

When the TNC does receive an acknowledgment for a connect packet it has transmitted, it will display a message back to the terminal, like:

```
*** CONNECTED TO callsign
```

Then the TNC will switch into the Convers Mode (dependent on the setting of NOMODE and CONMODE). Now what is typed will be data to be transmitted to the other station.

The MCON parameter setting is defaulted to OFF. This just means that once in connected mode, the only display to your terminal will be what you type and what the other station transmitted back. Packets transmitted by other stations on the same channel, will not be displayed.

Two things determine when entered data will be packetized.

One is the parameter setting of SENDPAC. Its default the carriage return or ENTER key. So, when a message is entered, the carriage return or ENTER key makes a transmit packet of that entered data.

A second parameter setting, PACLEN, determines the maximum number of characters that are in any one packet frame. If the number of characters entered reaches PACLEN, a packet will be transmitted even though the return or ENTER key was not pressed.

When a conversation is finished, end the connection. To do this, return to Command Mode and type a "d" for Disconnect. Remember that commands must be followed by the return or enter key. Once this TNC has received an

acknowledgment for the disconnect packet, the TNC will send this message to the terminal screen:

*** DISCONNECTED

Either station in a connected link can initiate a disconnect, no matter which station originated the connect.

Digipeating

Everything done so far will only be heard by those within range to hear this station's signal. With packet radio it is possible to reach stations farther away than can be heard direct.

The TNC parameter called DIGIPEAT that is set to ON by default. This makes it a possible relay station, or digital repeater, called a "digipeater" or just digi for short.

In many VHF communities one or more of these is put up in a good, high location and referred to as a dedicated digi. A TNC (configured as a digipeater) and radio is all that is needed for the digi to do its job. A terminal would be needed to change a TNC parameter setting, but it would not need to remain connected to the TNC for its digi to work. The higher the radio antenna, the more effective a digi will be.

If the TNC parameter setting MRPT is set to ON, more than just the "from" and "to" stations of the monitored packets will be displayed to the terminal. Callsigns of digipeater stations will also be shown.

The list of digipeater stations that relayed each packet, is called a path. An example of a digipeated packet heard, might be displayed as:

NØKN>KBØNYK, IAH*,LAG,AUS:
Hi there!

In this example, NØKN is talking to KBØNYK, using the digipeaters IAH, LAG and AUS to relay its packets. The asterisk beside IAH indicates that digi was heard direct by this TNC. Notice that IAH, LAG and AUS are not real callsigns but are aliases. The TNC provides a parameter called MYALIAS that allows an alias name for its digipeater function, which can often be easier to remember than a callsign. To make this connection NØKN would have entered the following TNC command:

C KBØNYK V IAH,LAG,AUS

V is short for via and up to 8 digi names may be used. The digis must be entered in the order they will be encountered along the path from this station to the intended destination station to which to connect. A space must be typed after the “C” and on both sides of the “V”, but digi names are separated by commas, with no spaces.

When digipeating, the transmitted packet goes all the way from the first station, through all specified relay stations, to the destination station. Then other stations response is returned by that same path in reverse. Chances for collisions and, therefore, for retries needed, are multiplied with every digi used. This is often called an end-to-end acknowledgment.

A CONNECT command issues a connect request packet to the specified station, via the specified path. Once a connection has been established, a virtually error-free conversation can take place between them.

A path can also be pre-configured in the TNCs command Unproto:

```
U CQ V NOM,LCH,SLI,BIX
```

Unproto sets up the path for anything that is subsequently typed in the Convers Mode where no connection exists.

Another way to get from one place to another is to connect to a “node”. A node will take care of the acknowledgment between it and the next node or end user. See the KA-Node section for more information. Ask other local packet users about other types of nodes which may be operational in the area, such as K-Net, TheNet, NET/ROM, G8BPQ, and ROSE.

- Kantronics’ TNCs support additional, advanced UI digipeating commands. For a description of Kantronics’ UI digipeating commands, see “Advanced GPS (or APRS) digipeating”, in the GPS Mode of Operation section of this chapter.

Multi-Connects

The TNC makes it possible to connect to more than one other station at the same time. This TNC supports up to 26 connection streams through its radio port.

The TNC command called MAXUSERS, determines how many connect streams are available at any time.

The TNC command called USERS, determines how many other stations are allowed to initiate connections to this TNC. Each incoming connect uses the next available stream, starting with stream A (if no other connections exist). If the number of streams set by USERS is full, then the next station attempting a connect with this TNC will receive a busy message instead of a connect.

However, when the command called MAXUSERS is set larger than USERS, those additional streams are available for outgoing connects to other stations.

To determine which current stream to which outgoing connected data will be directed, use the STATUS command (by typing STAT at the command prompt.) The TNC will report which streams are active and which one is current.

The TNC command called STREAMSW sets the character to be recognized, when switching streams. Its default is | (usually referred to as the “pipe”). It looks like a dashed vertical line and is located near the enter key on many keyboards.

To change to a different stream (to make a connection to a different station) or to send data to another station already connected on a different assigned stream, type the STREAMSW character (as is defined by that command setting), followed by the letter designation of that desired stream (with no space). No return or enter key is necessary. Again, use the STATUS command to view the current designated stream.

This TNCs CON and STA lights on the front panel will always indicate the connection status of the current stream.

When this TNC is in connected state, and the commands MONITOR or MCON are set to OFF, the normally displayed packet headers containing the “to” and “from” callsigns, will not be shown.

The setting of STREAMEV will then determine how often the stream designator will be included with received connected packet data. This parameter comes defaulted to OFF, so the stream designators are only shown when a change in streams occurs. Setting STREAMEV to ON will make the stream designators show on every connected packet received.

Setting the command STREAMCA to ON, will also add the callsign of the “from” station to be displayed beside the stream designator.

Round Table Discussions

Several stations talking together present a difficult situation for packet radio since the protocol requires two stations to connect in order to make sure they receive each other's packets correctly and in order.

To be absolutely sure that everyone got everything transmitted from this station, a packet connection would have exist to each other station, and a local streamswitch sent to the TNC, and re-send everything to each other connected station. That could get a bit cumbersome, so most stations just use the Unproto Mode and are aware that packet collisions may occur once in a while, resulting in lost packets. Missed parts of a conversation may be evident when an asked question is not answered.

Selective Monitoring

With MONITOR ON, the BUDLIST command can help in setting up selective monitoring. To limit which other stations' packets will be displayed, BUDLIST allows entry of up to 10 calls. Then, only packets heard from those callsigns in that list will be seen.

Timing

Dwait vs. Persistence and Slottime

When the TNC acts as a digipeater, packets received that need to be relayed are retransmitted as soon as the frequency is clear. Other TNCs on frequency (should) hold off or wait their turn, using a PERSISTENCE/SLOTTIME algorithm or DWAIT method to determine when to transmit.

This practice has been adopted to reduce collisions with packets being digipeated, since these packets must (by the AX.25 protocol) be acknowledged end-to-end. Kantronics' TNCs use the PERSIST/SLOTTIME algorithm to gain channel access.

The default value for DWAIT is set to zero. If the DWAIT method is used, the TNC sending non-digipeat packets will wait for DWAIT (times 10 milliseconds) time of clear channel before beginning to key-up the radio to transmit.

The algorithm used with the PERSIST and SLOTTIME parameters helps avoid collisions by randomizing the wait time before transmitting. The more random the timing the less chance of two TNCs transmitting at the same time and colliding.

Once the TNC detects a clear frequency it will wait SLOTTIME (times 10 milliseconds). Then, if the channel is still clear, it will internally generate a random number. If this number is smaller than the setting of PERSIST the TNC will transmit. If it is larger, it will wait another SLOTTIME and then generate another random number and again decide whether to transmit or not. When using PERSIST and SLOTTIME, DWAIT should be set to 0, since both will be used if specified.

As an example, assume that PERSIST is set to 63, and SLOTTIME is set to 10. This value of SLOTTIME results in a random number being generated every 100 milliseconds. When the TNC sees that the channel is clear, it waits 100ms, then generates a random number between 0 and 255 (inclusive). If, in this example, the number was 83, then the TNC would not start the key-up of the transmitter since 83 is greater than the 63 PERSIST value. Instead, it would wait an additional 100ms, and if the channel is still clear, generate a new random number. This time, it may come up with the number 27. Since this is less than the PERSIST value, we now start the key-up of the transmitter to send the packet.

TXDELAY

TXDELAY is an amount of time between the activation of PTT to the transmitter. Its setting should be adjusted to allow sufficient time for the radio to switch from receive mode to transmit mode and develop full power output on its transmit frequency.

The TNC sends flags (mark/space toggles) during the TXDELAY period, so if someone has this set extra long a repetitive sound will be heard at the beginning of the packet.

If the TNC starts sending a packet before the transmitter is at full power, the beginning of the packet may be lost, and other packet stations will not be able to decode it.

TXDELAY also allows the station to which data is being sent, sufficient time to detect that it is receiving a signal and start decoding the incoming packet.

TXTAIL

TXTAIL is amount time after the packet audio finishes, to hold PTT ON to the transmitter.

With some radios, an additional time at the end of a packet transmission may be needed, so that the transmitter is held ON, until the complete packet audio has

been transmitted. When this is the case, TXTAIL should be set to higher number than the default setting.

FRACK (Frame Acknowledgment Time)

If the TNC expects an acknowledgment of a packet it has sent, it will wait FRACK seconds for the acknowledgment. If an acknowledgment is not received it will either send a poll or retransmit the packet, depending on the setting of AX25L2V2. When digipeaters are used, extra time is allowed for each transmission using the following equation:

$$\text{FRACK} * ((2 * n) + 1) \text{ seconds}$$

where n is the number of digipeaters used for a packet connection. The length of the transmission (determined by PACLEN and MAXFRAME) also needs to be taken into account when deciding how to set FRACK. Longer packets (and more of them) require more time to be transmitted, more time to be repeated by each specified digipeater, and so on down the line. The FRACK timer begins when PTT is released (the complete packet has been sent) and is suspended when data carrier from the radio is present, or when this station is transmitting.

Retries AX.25 Level 2, Version 1 vs. Version 2

The way retries are accomplished depends on AX25L2V2 being OFF or ON. To explain this, follow a conversation through its path. Assume station "A" is connected to station "B" with Version 1 protocol (AX25L2V2 OFF). When station A sends a packet to station B, a receive acknowledgment is expected to be returned, indicating that station B has received the information. To verify that the proper packet (or frame) has been acknowledged, each packet frame has a sequence number. This number is sent as a part of the frame, so the receiving station knows where this packet belongs in the conversation.

Frame numbers range from 0-7. Because of this, we are limited to a MAXFRAME of 7 (we do not want the same frame number reused in the same transmission). This is also true for Version 2. If the first acknowledgment is received, there is really no difference between the two versions, practically speaking. The difference shows up with retries, so let's assume that the packet did not get through on the first attempt.

Assume that station A sends frame number 3 to station B. Station B does not receive the frame and therefore no acknowledgment is received by station A. With version 1, the entire packet is retransmitted (with the same frame number) to

station B, and this continues until station A receives an acknowledgment from station B. This acknowledgment can take two basic forms. The first time station B receives frame 3, an acknowledgment of the form "ready to receive frame 4" <rr4> will be returned. If this acknowledgment is sent, and station A did not receive it, station A will again send frame 3. Since station B already received frame 3, he would acknowledge it with the form "I've already got that frame, send me number 4" <rej4>. This is also known as Reject Frame sent. This process will continue until the retry count is exceeded when, under version 1, the sending TNC will initiate a disconnect and discard the packet. Local display of the frame response and frame error indicators (shown in < >), depends on the settings of the monitor commands MRESP, MCON and MCOM.

Looking at the same conditions under version 2 (AX25L2V2 ON). Station B does not receive frame number 3 from A and therefore does not send an acknowledgment to station A. Then, station A sends a POLL or question to station B saying, in effect, frame number 0 is expected; what frame should be sent?" <<RR0>>. Since station B did not receive the frame, station B would respond with <<rr3>>, saying "I'm ready to receive frame 3." At this point, station A, upon receiving the <rr3> would immediately resend the entire frame. If station B had already received frame 3 once but the acknowledgment never got to station A the question from station A for the retry would be the same. Station B's response, however, would be different. He would respond with "ready to receive frame 4" <<rr4>>. If station A does not receive station B's reply, this "POLL/REPLY" sequence would continue for the number of retries set in the sending TNC; if no response was received, and the RELINK parameter is ON, the TNC at station A would then begin to issue connect requests to station B since there is still an outstanding packet of information, and the connection needs to be re-established.

This is the major difference between AX.25 version 1 and version 2. The connect attempts would then continue for the number of TNCs retries setting, and if no response was received from station B after all of the above, station A would disconnect and discard the un-acknowledged packet(s). The parameter RELINK is defaulted OFF to avoid the reconnect attempt.

Flow Control

The flow control commands ensure that the TNC gets all data that is sent to it by the computer and that the computer gets all data the TNC sends it. When the computer sends the TNC data, the TNC stores this data in a buffer until it can packetize it, send it, and get acknowledgments. Similarly, when the TNC sends the computer data, the computer stores the data in a buffer until it can be processed, displayed, stored to disk, sent to printer, or whatever.

This buffer area in the TNC is of limited size; if more data is sent than will fit in the buffer the extra data will be lost. To make sure each device gets all the data it should from the other device, the two devices can tell each other to start and stop sending data. This is called Flow Control, and it can be accomplished in either of two ways, via software or via hardware.

Which way this is implemented, depends on the capabilities of the terminal program in the computer personal preference.

Software Flow Control

Software flow control sends special characters on the TXD and RXD lines of the interface cable. These are the same lines used for sending regular data between a TNC and a computer. Software flow control normally sends a <Ctrl+S> to stop data and a <Ctrl+Q> to restart data. When a buffer gets close to full, Either the TNC, or the terminal will send a <Ctrl+S> and expect the other device to stop sending data. When the buffer gets nearly empty, it will send a <Ctrl+Q> to tell the other device to send more data. How full or empty a buffer is when the special characters are sent is determined by the program. But, since the regular data lines are being used, a <Ctrl+S> manually sent as data from the keyboard, will also stop data from the TNC. And likewise, if there is a <Ctrl+S> in a file being sent, data flow from the TNC will stop until a <Ctrl+Q> is received.

XFLOW needs to be turned ON for the TNC to use software flow control. XOFF determines the character sent by the TNC to stop the flow of data from the computer, and the XON character restarts the flow. The TNC expects the computer to send the STOP character to stop data to the computer and the START character to restart data. To use software flow control, these commands would be set as follows: XFLOW ON, XOFF \$13, XON \$11, STOP \$13, START \$11. (Default settings.)

Two commands are provided to make it possible to send and/or receive these special characters in Transparent Mode and still use software flow control. TXFLOW enables flow control sent by the TNC to the computer and TRFLOW enables the TNC to recognize flow control sent by the computer. If both these commands are ON (and the above commands are set as stated) then software flow control will take place in both directions, to and from the TNC and computer.

- When the TNC is in its Transparent Mode, and the terminal is sending a file, the terminal does not need to tell the TNC to stop and start (since it is sending the file). By turning TRFLOW OFF and TXFLOW ON, the TNC will accept START and STOP characters embedded in the file and send them along to the radio as data. Meanwhile, the TNC can send the XOFF and XON characters to the

terminal to control the flow of data from the computer. When receiving a file, turn TRFLOW ON and TXFLOW OFF; the computer will accept all data and use START and STOP characters to control the flow of data from the TNC.

Hardware Flow Control

Hardware flow control in the terminal monitors the status of the RTS and CTS lines in the cable connected to the TNC. These two signals are inherent in a USB interface cable.

The TNC sets CTS high as long as it can receive data. Once its input buffer gets full, it sets CTS low. The terminal program monitors CTS input, and when it is low, it knows to stop sending data to the TNC. When CTS is again pulled high by the TNC, the terminal will restart sending data.

The terminal sets RTS high as long as it can receive data and sets it low to tell the TNC to stop sending data.

The TNC always uses hardware flow control, so set the terminal program to always use hardware flow control.

Convers Mode vs. Transparent Mode

In the Convers Mode the TNC has several defined special characters. To list a few:

Command	Default	Description
SENDPAC	<Ctrl+M>	Causes a packet to be formed
DELETE	<Ctrl+H>	Backspace character
REDISPLAY	<Ctrl+R>	Redisplays the keyboard buffer
CANLINE	<Ctrl+X>	Cancels a line
STOP	<Ctrl+S>	Stops output from TNC to computer
PASS	<Ctrl+V>	Pass a special character

These characters are all very useful when having a packet conversation with another station. To send a packet, simply press the return (ENTER) key.

To erase a mistakenly entered character, use the delete or backspace key. This is the same as sending a <Ctrl+H>.

To erase a whole line (before having sent that line), use the <Ctrl+X>. This will delete all characters back to those last typed since the last transmission.

To transmit one of these special characters as data inside a packet, that character must be preceded with a PASS character, the <Ctrl+V>.

Transparent Mode was designed for sending a file, whether that file contains plain ASCII text characters, or files containing binary or Hexadecimal data (such as a program file).

Special characters are not interpreted as commands by a TNC when it is in transparent mode. Those special characters are just part of data to be put in a packet and sent to the radio.

In Transparent Mode, a SENDPAC character cannot cause data to be formed into packets (as that character is just part of data to be sent); instead, packets are formed and transmitted at a rate set by PACTIME.

All monitor commands are treated as OFF in Transparent Mode. All that will be displayed on the terminal, what is being received from the connected station.

It may be advisable to set the USERS command to 1, other stations cannot interrupt the connection while transferring file data.

The setting of the ECHO command is also ignored. Even if ECHO is ON Transparent Mode will not echo data it receives from the attached terminal. Some terminal programs may have a setting to locally echo sent data to the screen while uploading.

Exiting Transparent mode

Getting into Transparent Mode is easy: just enter a “t” or TRANS while in Command Mode.

Since Transparent Mode allows the sending of all characters, the normal “return to command” mode character (<Ctrl+C>) will not work, as it would just be part of data to be transmitted.

To exit Transparent Mode, a special sequence of characters, or a “modem break signal” must be sent (if the terminal program has that function).

The special sequence of characters must be followed precisely. This example assumes the COMMAND character is defined as <Ctrl+C> and CMDTIME is 1 second:

- Wait at least 1 second since the last character was sent from the computer to the TNC

- Type a <Ctrl+C>
- Within 1 second type a second <Ctrl+C>
- Within 1 second type a third <Ctrl+C>
- Wait 1 second and the cmd: prompt should appear
- Or, just sending three <Ctrl+C> characters fast, should work

If the one second spacing before and after the three <Ctrl+C>s is not there, the TNC assumes that those characters are just part of data to be sent, and they are included in the packet to be transmitted. So be sure to allow at least one second before and after the three <Ctrl+C> characters.

Some terminal programs may have a macro function in which the three <Ctrl+C> characters can be stored, which will also work.

Remote Access

When the TNCs Remote Access function has been configured, it can be accessed remotely from another packet station, for the ability to change command settings or other parameters. Settings such as adding or deleting stations from LLIST, changing the size of the PBBS, changing that TNCs MYCALL and so on, all remotely.

Extreme caution must be used when accessing that TNC from a remote location. There is no built-in safeguard, and as such, it is possible to change parameters such that the remote TNC will no longer communicate.

To enable the remote access function, two parameters must be set: MYREMOTE, and RTEXT.

MYREMOTE is the callsign setting that the TNC would answer to, and that another station would connect to, to gain remote command access. It does have to be unique (not the same any other callsign in this TNC or any other TNC on the same radio channel). It does not have to be an actual callsign but could be an alias name.

RTEXT is the password string, from which a password access code is generated, which is sent to a station that connects to the MYREMOTE callsign.

For instance, RTEXT may simply be set to:

RTEXT Code

Noting the characters at each position in this string;

Character #:	1	2	3	4
Letter:	C	o	d	e

To determine a correct response to the password query, when a connection is established to MYREMOTE, that TNC will return three lines of numbers. Those numbers may look like:

```
1 1 1 3 4 3
3 1 4 3 1 3
2 1 1 2 1 3
```

Pick ONE of these lines and decode the password string using the example above.

If line 3 is chosen, (2 1 1 2 1 3), the following character string should be entered:

oCCoCd

(since small letter o is at position 2 in the RTEXT string, capital C is at position 1, and so on)

Note that case is significant and spaces are considered valid characters.

If a properly decoded password is not received correctly, that TNC will reply with three new lines of numbers. A maximum of three attempts to properly decode the password string is allowed. If those three attempts fail, that TNC will disconnect its MYREMOTE and will not reply to re-connect or new connect requests, for 15 minutes.

Also, if a connect was made to the MYREMOTE, and a disconnect happens before the password sequence has been completed, the 15-minute penalty timer will be effect.

Be careful when using the remote access feature. ANY command in that TNC can be changed, without restriction (other than viewing the current RTEXT setting).

This can lead to problems if, for instance, a change is made to an interface mode other than standard packet mode. Changing the INTFACE command to KISS (or some other non-packet operating mode) and then sending a RESET command, will place it into the KISS mode, and that TNC will no longer communicate with other stations.

Also, connected to the MYREMOTE of a TNC, then command it to do something like CONNECT W1ABC, that remote TNC will indeed attempt to connect to that station, but the myremote connection will not see any data from that other connection. That connected data would be sent to the computer interface port of that remote TNC, instead.

Other commands that could be sent through the remote access connection, which would cause that TNC to re-allocate its internal memory, and reset itself, should be avoided. (such as changing the settings MAXUSERS, NUMNODES,PBBS, etc.)

Another limitation applies to the amount of data in response to any command returned data from that remote TNC. Its response is limited to no more than 300 characters. For example, sending the command DISPLAY would result in only the first 300 characters of the normal display command response will be returned.

PBBS (Personal Mailbox)

Introduction

This TNC includes a Personal Bulletin Board System (PBBS), a mailbox, which can store and forward messages addressed to its MYCALL, or to callsigns of other users. It provides many of the same message facilities as a computer-based packet radio BBS, including the forwarding of different types of messages (such as Bulletins, Private, and NTS traffic).

The TNC front panel MAIL led (the yellow one), will be lighted when any connection exists to its pbbs.

Historical note: The first BBSs in packet radio were PC-based, using software developed by Hank Oredson, WØRLI. A national system of packet-radio based BBSs is accessible in many areas, permitting the automatic forwarding of messages, and files, from one BBS to another, or through internet links.

The TNCs personal mailbox is capable of interacting with the large community bulletin board systems (RLI, MBL, etc.) for exchanging messages with those systems. Messages entered locally, or from other local packet stations, can entered as forwardable to an interconnected BBS system, then will be passed on into that system.

The PBBS can be configured to periodically connect to a BBS to pass on forwardable messages, and to check for incoming messages to be stored locally.

Using the PBBS

To use this TNCs PBBS, while in command mode (the cmd: prompt is displayed), connect to the callsign of the PBBS. Find the callsign of the PBBS with the TNC command MYPBBS. For instance, if the MYPBBS is NØKN-1, connect to it simply by typing "C NØKN-1". In this example, a connection to the PBBS is inside this TNC is internal. It works the same as when connecting to any other station, just that no packets are transmitted. The pbbs connection prompt seen, will be the same as any other station that connects would see.

When connecting to the PBBS, the TNC will display its connect message, that same as when connecting to any other station;

```
*** CONNECTED to NØKN-1
```

The PBBS will then send its SID (Status ID - e.g., equipment ID) and initial sign-on message. If a PTEXT is defined, the TNC will send it as the next line, and then send the PBBS command prompt. This TNCs PBBS prompt is;

```
[KPC-3XE-1.0-HM$]  
100000 BYTES AVAILABLE IN 2 BLOCKS  
PTEXT would be here (if any)  
ENTER COMMAND: B,J,K,L,R,S, or Help >
```

Note the lighted mail led on the TNC front panel.

At this point the pbbs will allow listing and reading stored messages, entering a message to another user, or to issue any other mailbox command.

To enter a message to another station, such as KBØNYK, the pbbs command SP (send private) would be used;

```
SP KBØNYK
```

and the pbbs responds with:

```
100000 bytes AVAILABLE IN 1 BLOCK  
SUBJECT:
```

enter a short subject line:

```
Just a quick question
```

The pbbs responds with:

```
ENTER MESSAGE n—END WITH CTRL-Z OR /EX ON A SINGLE LINE
```

(where “n” is the next available message number in the mailbox).

Now enter the text of the message. When the message is complete, to end the message and have it saved, type a <Ctrl+Z> (hold down the control key and press Z) or type /EX) on a line by itself. When the pbbs receives this end-of-message, it responds with that confirmation, and its prompt:

```
MESSAGE SAVED  
ENTER COMMAND: B,J,K,L,R,S, or Help >
```

It is now ready for any mailbox command.

The “Mail” indicator on the front panel of the TNC blinks to indicate that there are unread messages addressed to the callsigns of MYCALL or MYPBBS (a local user). See the TNC command PBLIST for how you can add other callsigns to the list of callsigns that are to be included as “mine.” The PBLIST is also used with the PBPERSON command and the following PBBS commands: LMine, KMine, and RMine. Note that the mail-status indicator on the front panel of the TNC is turned ON (without blinking) when anyone is connected to your PBBS. The “Mine” part of those commands refers to the local user (this TNCs MYCALL or MYPBBS).

Also, when no connection exists to the pbbs, and unread messages exist (except forwardable messages), the TNC front panel mail led will blink at a regular interval.

PBBS Commands

The commands available to any users connecting to a Kantronics TNCs PBBS are as follows:

B(ye)

This command is entered by the PBBS user to cause the PBBS to disconnect the user.

E(dit) n [BPTYNFH] [>tocall] [<fromcall] [@BBS] “old” “new”

This command is available only to the system operator (SYSOP) or to a user at the keyboard of a computer connected directly to the TNC (i.e., not to a remote connection, except for one having gained SYSOP capability by successfully completing the password query). For an explanation of this command, see the section below on “Editing Message Headers.”

H(elp)

Displays a HELP menu.

J(heard)

Displays a list of stations recently heard by the TNC. The list will include a date and time stamp indicating when each station was heard.

J(heard) S(hort)

This command is like the J(heard) command above but only displays a list of callsigns of the stations heard.

J(heard) L(ong)

This command is like the J(heard) command above but also includes any(all) digipeaters used by the stations it has heard and the destination callsign.

L(ist) [x [y]] [;]

Lists all messages in the mailbox which the user is allowed to read. This will include all BULLETINS, TRAFFIC, and any PRIVATE messages addressed TO the MYCALL or MYPBBS entered by the local user. If connected from a terminal connected directly to this TNC, SYSOP access is inherent (keyboard or when sysop has been acquired from a remote user), ALL messages will be listed.

The option of specifying a beginning message (x = START #) and, to specify an ending message for the message list (y = END #) is available. Also, entering the LIST command with a following semi-colon (;) will also include the @BBS and BID (BULLETIN ID - see section on "Sending Messages") of the messages (if any) This applies to all LIST commands.

L(ist) <|> call [;]

Can be used to limit the pbbs message listing to only those messages which are addressed to a specific callsign (>), or which were sent from a specific callsign (<). For example, to list only messages addressed to AMSAT, the command would be;

L >AMSAT.

LB [;]

Lists all BULLETINS in the mailbox.

LC [cat [;]]

Using the LC command by itself will cause the PBBS to list the TO field of all BULLETINS in the mailbox. This will just be a list of the actual TO fields, and not a list of the messages. If a category is specified (i.e. LC RACES) the PBBS will list the full message headers of all BULLETINS addressed to that category.

LL n [;]

Lists the most recent number of messages in the mailbox. Again, only BULLETINS, TRAFFIC, and PRIVATE, which the connected user is allowed to read, will be listed.

LM(ine) [;]

Lists all messages in the mailbox which are addressed to MYCALL or MYPBBS.

LO [+|-]

This command allows order in which messages are listed to be changed. When set to +, the messages are listed in ascending numerical order (oldest to newest).

When set to “-”, messages will be listed starting with the highest message number (newest to oldest). This command will not be available if the SYSOP has the TNC command PBLO is set to FIXED.

LT [;]

Causes the PBBS to list all TRAFFIC messages that are currently in the PBBS.

K(ill) n

Deletes message number n from the mailbox. Only message types TRAFFIC, PRIVATE messages addressed TO or FROM the local user, or BULLETINS entered by the local user. A pbbs SYSOP may delete any message, including BULLETINS.

KM(ine)

Deletes any messages in the PBBS addressed to the local user that have been read. Any local users' message that has not been read, will not be deleted.

NL n

The NL (number of lines) command controls how many lines are output at a time when a large message is read. After n lines are printed, the message “PRESS Q TO QUIT, C FOR CONTINUOUS, RETURN FOR MORE” will appear. ‘Q’ will abort the printout of the message. ‘C’ will cause the rest of the message to be printed without stopping. Pressing RETURN will print the next n lines.

R(ead) n

Reads a specific message by number. Only PRIVATE messages addressed TO the local user, or sent by the local user, may be read, as well as any BULLETIN or TRAFFIC messages. After having read a PRIVATE message addressed to the local user, that messages STATUS flag will automatically be set to Y, indicating that message has been read.

RH n

The command RH (Read with headers) is a variation of the Read command. Using RH instead of just R gives will include the messages headers as well as the other information provided by the Read command. The message header contains the date/time the message was received at each forwarding BBS along the path to this station.

RM(ine)

Displays all messages in the PBBS addressed to the local user that have not already been read.

S(end) call

Sends a PRIVATE message to the callsign specified. Using Send is the same as using the SP command.

SB cat

The SB (Send Bulletin) command is used to send a BULLETIN to the PBBS.

SP call

The SP (Send Private) command is used to send a PRIVATE message to a specified callsign. Using this command is the same as using the S(end) command.

ST zip

The ST (Send Traffic) command is used to send NTS type traffic messages to the PBBS. Some of these commands are described in more detail below.

U(sers)

The Users command will display any other users who may also be currently connected to the PBBS.

MESSAGE STATUS Flags

Each message stored in the PBBS will include a flag character that shows the messages status. When listing any messages, each message status is located in the column with the heading ST.

Message status flags are; B,H,F,PN,PY,T,Y,

B indicates the message is a Bulletin

H indicates the message is Held for sysop review

F indicates the message has been forwarded

PN indicates the message is personal type, and has not been read

PY indicates the message is personal type, and has been read

T indicates the message is a Traffic type message

Sending Messages

The SEND command (and its many forms) allows the following syntax:

S	call	[@ bbcall[.haddr]]	[\$ mid]
SP	call	[@ bbcall[.haddr]]	[\$ mid]
ST	zip	[@ location[.haddr]]	
SB	cat	[@ location[.haddr]]	[\$ bid]

where:

- **call** is the callsign of the station to which the message is addressed.
- **bbcall** is the callsign of a full-service BBS to which the message is delivered (for possible forwarding).
- **haddr** is the complete Hierarchical address designator for the BBS system or destination of the message (see the section below on “Hierarchical Addresses” for more detailed information). Contact a local community BBS SYSOP for complete information on local Hierarchical addressing.

- **location** is the designator used for distribution of the message. For TRAFFIC, this is typically NTSxx where xx is the two letter POSTAL code for the state.
- **mid** is the MESSAGE ID assigned to the message by the originator.
- **bid** is the BULLETIN ID assigned to the message by the originator.
- **zip** is the 5-digit postal zip code (or postal code)
- **cat** is the message category. For instance, a message requesting help on a subject may be sent to the category HELP, info sent to INFO, items for sale to SALE, etc. Contact a local community BBS SYSOP for some other examples and suggestions.

Some examples of commands to enter a forwardable message would be:

SP WB5BBW @ W5AC.#STX.TX.USA.NOAM

This command sends a private message to WB5BBW. The message should be sent to the W5AC BBS system (and following the hierarchical address), in South Texas (.#STX), which is in Texas (.TX), which is in the USA (.USA), which is in North America (.NOAM) where WB5BBW can retrieve it.

ST 88030 @ NTSNM

This command sends an NTS traffic message to a non-ham, or to someone who is not on packet, living in zip code 88030 which is in New Mexico. The location field contains the NTSxx (xx = NM) to indicate that the 88030 zip code is in New Mexico.

SB RACES @ ALLUS \$RACESBUL.010

This command sends a bulletin addressed to RACES, which should be sent to all BBS system in the USA (ALLUS) and has been assigned the Bulletin ID (BID) RACESBUL.010. This BID prevents the same message from being duplicated as it travels throughout the BBS system.

When a message is entered into the PBBS, it must include the @BBS (bbcall[.haddr]) field if that message is intended to be reverse forwarded from the PBBS to a full-service BBS system. Any message entered in the PBBS from another station will initially be marked with a status of H (held) and will not be reverse forwarded until the SYSOP has edited the message header and changed

the H flag. This gives the SYSOP full control over the messages relayed by his station.

Messages entered from the local keyboard connected to the TNC do not require editing in order to be reverse forwarded but must include at least an @BBS field to enable the reverse forwarding.

Listing Messages

When listing messages with any of the LIST commands, a display similar to the one below will be shown. If the list command included the optional semi-colon (;), the @BBS field (if it exists) will be shown in square braces after each line.

MSG#	ST	SIZE	TO	FROM	DATE	SUBJECT
6	B	45	KEPS	W3IWI	12/19/91 09:37:11	2 Line
Element set						
4	B	26	HELP	WB5BBW	12/19/91 09:34:05	Xerox 820
3	T	38	66044	WØOUU	12/19/91 09:33:42	QTC
Lawrence 913/842						
2	PN	14	NØAPJ	WØSC	12/19/91 09:33:27	AMTOR
1	B	30	ALL	NØKN	12/19/91 09:32:49	Need help on
AMTOR						
9712 BYTES AVAILABLE						
NEXT MESSAGE NUMBER 7						
ENTER COMMAND: B,J,K,L,R,S, or Help						

The message number (MSG#) is listed, followed by the STATUS of the message. This status includes the message type (B=Bulletin, T=NTS traffic, and P=Personal message). The second character in the ST column is the current status of the message.

A **Bulletin status** (type B) can be:

F—it has already been forwarded to another full-service BBS

H—it is being held for review by the SYSOP because it was entered into the PBBS over the radio.

An **NTS traffic message** (type T) may have a status of:

H—indicating that it is being held for review by the SYSOP before it may be forwarded.

The **Private message** (type P) can have the following status characters in the second position:

H—This is a personal message that has an @BBS field but is being held for review by the SYSOP before it may be forwarded.

N—This message is a Personal message that has not been forwarded and has not been read by the station it is addressed to. If it is forwarded to a full-service BBS, it will be deleted if PBKILLFW is ON.

Y—This message has been read by the station it is addressed to, but has not been killed. It will not be forwarded even if it has an @BBS since it has already been read.

Reading Messages

To read a message (e.g., a bulletin such as that in the previous section listed as being to ALL), use the READ command (see PBBS command section for details on using this command).

Editing Message Headers

The following command can be used to edit message headers:

E(dit) n [BPTYNFH] [>toall] [<fromcall] [@BBS] “old” “new”

This command, available only to the system operator (SYSOP), or to a user at the keyboard, allows the sysop to edit the message headers listed below for any message in the mailbox. A local user is the sysop when connecting to the PBBS from a local connected terminal. Sysop status can be gained by connecting to the PBBS remotely from another packet station. see the section titled “Remote Sysop Access to the PBBS” for more detail.

When editing message(s) the following can be changed:

- The TYPE of the message:
 - B — Bulletin
 - P — Private
 - T — Traffic

- The STATUS of the message:
 - Y — Yes [it has been read]
 - N — No [it has not been read]
 - F — Forwarded [it has been forwarded]
 - H — Held [it is not available for reverse forwarding]

Note: F and H are toggles. Entering F switches the status of the message between FORWARDED and NOT FORWARDED. Similarly, entering H switches the status of the message between from HELD and NOT HELD (H or N).

- Who the message is to or from:
 - >toall
 - <fromcall
- The destination mailbox (@BBS):
 - @BBS[.haddr]
- And the subject or text of the message

When a message is listed, the “toall” and “fromcall” appear in the TO and FROM columns, respectively. If a message has been entered with an “@BBS” for forwarding, the complete hierarchical address is shown when the message is read:

```
MSG2 02/10/92 10:30:58 FROM KBØNYK TO HELP
@wa4ewv.#stx.tx.usa.noam
```

The @BBS is also listed when using the semi-colon (;) option with any of the list commands.

The message Edit command is available after connecting to the mailbox from the attached terminal, or by connecting over the radio. If connecting from another station over the radio, SYSOP privilege must be gained first, by giving the SYSOP command entering the password verification string. It must be sent correctly in order to gain SYSOP access remotely. This is explained in detail later.

For example, to edit message number 2 which currently is a PRIVATE message addressed to WØXI. the message has been read by WØXI so it shows a status of Y. It may list as:

MSG#	ST	SIZE	TO	FROM	DATE	SUBJECT
2	PY	53	WØXI	NØKN	01/14/92 03:36:45	Good afternoon

In another example: change this message so that it is addressed to NØGRG instead and also change the Y flag to N to show that it hasn't been read. Connect to the PBBS (either from the KEYBOARD or as remote SYSOP from another station), and issue the command:

```
e 2 N >NØGRG
```

This could be done with two separate commands, or it may be accomplished with the single command shown above.

A message subject or text in the message can be changed by entering two strings in the Edit command. The *first* occurrence of the string "old" will be replaced by the string "new". For example, to change the subject from Good afternoon to Good morning:

```
e 2 "afternoon" "morning"
```

Hierarchical Addresses

Hierarchical addresses (explained below) are used to accommodate the national packet traffic system, including messages processed by PC-based packet radio BBSs.

The HTEXT command sets this stations Hierarchical address. It is used whenever this TNCs PBBS forwards messages to another BBS to provide complete return addressing capability.

Other PBBS features include the ability to select whether to retain, or not retain, private and traffic messages after they have been forwarded, and whether or not to HOLD messages received over the radio, for sysop review.

A hierarchical address consists of state, country, and continent codes, separated by periods. Some areas (in large states particularly) use sub-state designators.

For instance, a station in Rhode Island might use a hierarchical address of RI.USA.NOAM. This means Rhode Island (RI) which is in the United States (USA) which is in North America (NOAM). Check with a local packet BBS sysop to be sure what the proper addressing is for your location.

Entering the command PHEARD will return a list of callsigns of stations that recently connected to this station's mailbox, along with their start and stop times.

Advanced Configuration of the PBBS

When this TNC was first initialized, (when a callsign was entered), its PBBS was will automatically enabled. Its MYPBBS callsign was set to that callsign with an SSID of -1. The PBBS is allocated 100 k Kbytes of RAM.

The size of the PBBS is set in the command PBBS. The maximum amount of memory that can be allocated for its use depends on the amount of free memory available. Other commands, such as NUMNODES, MAXUSERS, and MYREMOTE, (etc.) all share the TNCs internal RAM, and their settings will affect the amount of available memory.

If the size of the PBBS is changed, existing messages will not be re-numbered as long as the new size is large enough for all existing messages, no messages will be lost.

If the PBBS command is entered with a size the same as current size messages will be re-numbered beginning with 1.

- To see the current size of the mailbox, enter the command "PBBS" (without any number) at the Command prompt.

The TNC command PBUSERS sets the maximum number of users allowed to connect to the PBBS at the same time.

Changing the PBBS size, or the number of PBUSERS, will cause a soft reset.

An incoming connection to this TNCs MYCALL can be automatically switched to the PBBS, setting the CMSG command to PBBS.

When this is done, a user who connects to the MYCALL in this TNC, will be sent the content of CTEXT (if any), and then be automatically connected to the PBBS.

The PBBS will then send its System ID (SID) and sign on message. The SID is enclosed in square brackets and consists of the TNCs name and pbbs version

and the supported feature set. If a PTEXT has been configured, that will be sent, then the PBBS command prompt.

For example, this TNCs pbbs SID is:

[KPC3XE-1.0-HM\$]

This is the TNCs name (KPC3XE), its pbbs version number (1.0) and its feature set (HM\$). The H means it supports Hierarchical forwarding, the M stands for Message ID, and the \$ indicates BID support. These identifiers are the SID definitions published by Hank Oredson (WØRLI) with his Community BBS system.

The PTEXT command is a custom PBBS greeting message, to be sent to a connecting station. It is included in the PBBS sign-on when a connection is established. It accepts up to 128 characters as a text.

If it becomes necessary to disconnect a station from the PBBS, the local terminal can be used to send the TNC command DISCONNECT MYPBBS.

If a station connects to your PBBS and no activity occurs on the connection for 15 minutes, the PBBS will automatically disconnect that user in order to make the PBBS available to others.

The TNC command HTEXT must be set to allow forwarding and reverse forwarding capability in the PBBS. HTEXT specifies the local Hierarchical address, which is needed for complete return addressing capability.

When a community bulletin board (BBS) forwards a message into the PBBS, each message contains header lines beginning with "R:". These are routing headers that show the complete path taken by the message. By default, these headers will be stored in the PBBS with each message. If desired, setting the TNC command PBHEADER to OFF will prevent storing the routing header lines (for more information, see the section below titled "Routing Lines").

Setting the TNC command PBPERSON to ON, will set the PBBS to only accept messages addressed to this TNCs MYCALL, MYPBBS, or to any of the callsigns contained in the PBLIST command.

The TNC command PBLO is used to determine the order in which messages are listed to any station connected to the PBBS, when requested. When PBLO is set to OLD, messages will be listed oldest first (message 1, then 2, etc.). When set to NEW, the most recent message will be listed first. The second parameter of this command determines whether a user connected to the PBBS is allowed to change the listing order while connected. When set to FIXED, the user cannot

change the listing order, and when set to VARIABLE, the user may change the listing order while connected to the PBBS, by using the LO command.

Remote SYSOP Access to the PBBS

A user connected to the PBBS can gain remote SYSOP access (e.g., to edit messages, using the EDIT command) by entering the SYSOP command when first connected. When the SYSOP command is received, the PBBS will respond with the password query. That user will receive three sets of numbers from the PBBS, the same as when a station connects to the TNCs MYREMOTE. These numbers indicate the character positions of the RTEXT to be used as the password for logging on for this remote access.

For instance, let's say RTEXT is set as:

This is a sample rtext.

Now when the PBBS receives the SYSOP command, it might respond with:

1	12	3	18	6	9
2	10	22	5	7	18
13	16	4	9	1	20

If the first line of numbers is chosen, the following would be sent as a response:

Tairia

(T is the 1st letter, a is the 12th letter, i is the 3rd letter, and so on. See the RTEXT command in the Commands section for a more detailed explanation.)

Reverse Forwarding Messages from the PBBS

The pbbs allows entry of messages to be reverse-forwarded to full-service BBSs (RLI, MBL, etc.), when requested. These messages have a special format and can be entered in any personal mailbox. That BBS would connect the TNCs PBBS, then request any mail that is addressed to be forwarded. For example, a message is entered in the PBBS, that is to be forwarded to WA4EWV, in Texas. The BBS closest to that station is WB5BBW, A message to that station would be entered into the PBBS with the command:

S WA4EWV @ WB5BBW

Entering a message into the PBBS with an @ BBS, will cause the PBBS to be able to forward that message to a BBS, or for a BBS to get that message when

requested (reverse-forwarded). To improve the chances of this message reaching its destination, it could be entered with complete hierarchical forwarding:

S WA4EWV @ WB5BBW.#STX.TX.USA.NOAM

Complete information on Hierarchical forwarding can be obtained from a local BBS system operator, but basically the first field after the @ symbol is the HOME BBS of the station to which that message is being sent. The next several fields (separated by periods) are the state (two-letter postal abbreviation), country, and continent. In this case, since Texas is so large, it is sub-divided into smaller areas. These are indicated with the # symbol (in this case #STX—South Texas).

Messages that are entered into the PBBS in this format will be reverse forwarded to the full-service BBS when requested.

The TNC, PBBS acts like a “smart BBS” when forwarding to or from a full-service BBS. If the PBBS recognizes that another “smart BBS” or a full service BBS is connecting (that station send its SID contained in the [] brackets), it will no longer send its SUBJECT: prompt, nor will it send the ENTER MESSAGE prompt as it would send to a standard type user. Also, when in “smart BBS” mode, the PBBS will only send the prompt > to the connecting station. This is designed to reduce the amount of data on the packet network, since “smart” BBSs know those normal commands to send and receive messages.

If the TNC command PBKILLFW is set to ON, once a Private or Traffic message has been successfully forwarded out of the PBBS, that message will be deleted.

If PBKILLFW is set to OFF, the message will be retained in the PBBS, and its status will be set to F (forwarded).

Selecting a Home Full-Service BBS.

One important use of the Kantronics PBBS, is to connect to the national system of packet-radio based BBSs, which was described earlier in the “historical note” at the beginning of the PBBS section. A “home” full-service BBS can be used for sending and receiving mail, much like using P.O. Box at a postal facility. That full-service BBS should always be on and available.

Each PBBS user should only use one full-service BBS for sending and receiving mail. This BBS is then considered to be this stations HOME BBS and should not be changed unless moving to a new location. When connecting to a home BBS and listing messages (using the L command), a list would be returned containing information about each available message on the system. A list of messages on one local full-service BBS system might look like this:

Msg #	TSL	Size	To	From	@ BBS	Date/Time	Subject
=====	=====	=====	=====	=====	=====	=====	=====
59765	B\$	1491	NASA	N5IST	ALLUS	1004/1529	GALILEO STATUS 09/30/93
59764	BNL	468	WX	NONEJ		1012/1017	KC Forecast 10/12 400am
59763	BNL	659	WX	NONEJ		1012/1017	MO Forecast 10/12 400am
59759	B\$	2240	NASA	N5IST	ALLUS	1004/1529	MARS OBS. STATUS 9/27/93
59758	B\$	1642	NASA	N5IST	ALLUS	1004/1529	MARS OBS. STATUS 9/22/93

This list shows the message number, type and status information, the size of the message, the addressee (TO field), the originator (FROM field), and the @BBS field. In addition, the list shows the date and time the message was received at this BBS and a short subject for the message.

Routing Lines

Under current FCC requirements, BBS systems that can store and forward messages without an operator being present must have the capability to provide a record of the path the message has taken from its origination. To accomplish this, BBSs include a routing line, beginning with "R:". This "R:" line includes the date and time the message was received, message number, BBS call and hierarchical routing information.

When a message is read using the R command (e.g. R 59765) the header of that message is displayed. For example:

```

From      : N5IST
To        : NASA @ALLUS
Type/status : B$
Date/time  : 04-Oct 15:29
Bid       : NASA0930.GAL
Message #  : 59765
Title     : GALILEO STATUS 09/30/93
Path      : !NØKN!NOLLY!NØOER!NØOBM!NXØR!AGØN
           !N7MMC!KTOH!KAØWIN!N5IST!

```

The Path: statement in the header lists the most recent BBS systems that have been used to relay this message from its origin to that BBS. This path information is required by the FCC to allow complete traceability for any message in the system.

What is seen in the PATH statement is not the complete information on the routing, but simply a summary of the systems that have handled the message. To see the complete information, BBS systems allow a second version of the READ command (RH or V) that will display more routing information. A routing list of a sample bulletin message appears below.

```
R:931012/1107 27268@NØKN.#NEKS.KS.USA.NOAM
```

R:931012/1025 16433@NOLLY.#NEKS.KS.USA.NOAM
R:931011/2021 928@N00ER.#NEKS.KS.USA.NOAM
R:931008/1814 20728@N00BM.#NCKS.KS.USA.NOAM
R:931008/2003 19520@NX0R.#NKS.KS.USA.NOAM
R:931008/1153 30798@AG0N.#WNE.NE.USA.NOAM
R:931007/1147 35850@N7MMC.#SEWY.WY.USA.NOAM
R:931007/1712 49403@KTOH.#NECO.CO.USA.NOAM
R:931007/1639 63792@KA0WIN.#SECO.CO.USA.NOAM
R:931004/1529 46383@N5IST.#WTX.TX.USA.NOAM

By examining this list from the bottom up, we may see that the message entered the system on October 4, 1993 at 15:29 (R:931004/1529). It was message number 46383 on the N5IST BBS (@N5IST) which is located in West Texas (#WTX), which is in Texas (TX), which is in the United States (USA) which is in North America (NOAM). From this station, it was relayed on October 7 at 16:39 to the KA0WIN BBS in Southeastern Colorado. By following this information, it is possible to determine where the message traveled and when it was relayed from each station. The information following the @BBS callsign is called the **hierarchical routing information** (in this case #WTX.TX.USA.NOAM).

When connected to a local BBS and sending a message, that BBS automatically generates this first R: line. As the message is sent to its destination, each BBS that handles that message adds its own R: line to the message header. Besides the requirement of the FCC, the R: line provides a method for any user, anywhere in the world, to send a reply or respond to a locally sent message. As the message is passed through many BBSs, each BBS will add the message originator into its White Pages—a directory of packet users. Each BBS makes a note that the originator of the message sent the message, and that the message was entered at the original BBS listed in the last R: line in the message (the first BBS that handled the message).

Because of this, a distant user can simply send a reply to the message originator, using the send reply (SR) command of his local BBS. That BBS will then address the return message to the originator, using the @BBS and hierarchical routing information in the last R: line of the message sent. A user may also simply use the send private command (SP) to send a message to the originator. If the user does not enter complete hierarchical addressing with the SP command, the BBS will attempt to look up the originators' call in its White Pages and add the routing automatically. However, if the user did supply complete addressing information when entering that message, the BBS would normally assume it is correct and not check the White Pages.

BBSs use this hierarchical information to be able to return a reply message back to its originator. The message someone sent (using the above example) would be

addressed to URCALL@ N5IST.#WTX.TX.USA.NOAM. As the message passes through the BBS system for forwarding, the BBS first looks at the callsign of the addressee (URCALL). If that BBS doesn't know how to forward a reply message, it then looks at the @BBS field (N5IST). If it doesn't have any information on how to forward to N5IST, it looks at the first part of the hierarchical address (#WTX), and if not knowing that it would then look at the next part of the hierarchical address (TX). Assuming this BBS is in the United States, it knows TX means Texas and knows that this message needs to be relayed to a station in that area.

Once the message is passed on to the first BBS in Texas, that system must use the previous field for forwarding (#WTX). Once it reaches a system in West Texas, the forwarding occurs based on the @BBS.

When the message reaches the destination BBS specified in the @BBS field, it can forward the message directly to the originator, since using that HOME BBS system was listed.

When entering a forwardable type message into Kantronics PBBS and including the routing information, that message may be forwarded automatically to another BBS. When the message is forwarded out of the PBBS, an R: line is included as the originating BBS. This line includes the same information as any other BBS.

This R: line consists of the date/time the message was entered into the PBBS, the message number, TNCs MYCALL and the HTEXT that was set in that TNC.

For instance, R: line of that message might be:

R:931008/1255 23@URCALL.#WTX.TX.USA.NOAM

Some BBS operator groups are insisting that a PBBS system is NOT a BBS and therefore should not include R: lines. Their reasoning is that in the above example, EVERY BBS in West Texas would have to know how to send messages to a callsign not just to a HOME BBS. One solution to this is to include the callsign of a HOME BBS as part of the HTEXT. This would change the R: line to:

R:931008/1255 23@URCALL.N5IST.#WTX.TX.USA.NOAM

As this forwards through the system, all West Texas BBSs can still forward the message to N5IST because that call is a part of the hierarchical routing.

We suggest following local custom by turning the R: line feature ON or OFF accordingly.

If the local BBS SYSOP demands that messages from the PBBS not add R: lines to messages, set the TNCs PBPERSON command ON. This will limit the PBBS to receiving messages addressed ONLY to its TNCs MYCALL or MYPBBS call. Then, this PBBS will only forward messages entered locally (no third-party messages) and will not add the R: line to the routing.

Hint: When reading messages with the normal READ command, the PBBS will not display those R: lines, but instead will display a short PATH: line, indicating the most recent BBSs that have handled this message. To read the full routing of a message, use the PBBS command RH with the message number, read that message with all the headers displayed.

If the TNC command LTRACK has been configured, an option to access stored LTRACK messages with the LTRACK command will be available to the station connected as a sysop to the PBBS.

In the PBBS HELP listing, that command will be listed as;

LTRack n [string] DISLPLAY LAST n MESSAGES IN TRACK BUFFER

This concludes the section on PBBS mode of operation.

GPS NMEA Interfacing

Most Kantronics' TNCs, e.g. KPC-3XE, 9612XE, KAM XL, and older models, support GPS modes, explained below.

TNC commands allow for interfacing with a GPS device supporting the NMEA 0183 interface standard, storage of GPS data supplied from that device, and periodic beaconing of that stored data.

Other commands can be configured to set the unproto (path) of transmissions of those stored GPS data strings.

Overview

The Global Positioning System (GPS) was developed initially for military use by the U.S. Department of Defense, to provide worldwide positioning and navigational information for U.S. military forces. Since then, a broad range of commercial, civilian, and amateur radio applications have been developed, based on the GPS. The GPS consists of 24 (or more) satellites, each of which transmits positioning information continuously. GPS receiving units - devices that contain a receiver, LCD screen, and (usually) a data port - collect this positional information and use it to compute position.

This information can then be captured and stored, to be retransmitted by any means possible to report the position of the GPS unit (and its vehicle).

Most GPS units sold today are NMEA compatible. That means that not only can they display latitude, longitude, and time on an LCD screen, but the information can be presented to a computer or TNC via the GPS unit's serial data output port.

The information is presented in ASCII text, in the form of NMEA "sentences". A number of these sentences are available—with varying kinds of information, but the one used most often for amateur radio activities is "\$GPGGA". This sentence contains latitude, longitude, altitude, and time (in UTC, derived from the satellite clocks). The TNC, can receive and retransmit this information as beacons and/or store these sentences in a large tracking buffer for later retrieval.

The TNC, can be configured to update its internal clock to UTC, derived from the GPS data received from satellites. Hence, the location or track of a remote or mobile packet station, recorded at a specific time, can be received by other packet stations.

Packet radio all over the world can be used for reporting the position of their vehicles, following balloon launches, finding hidden transmitters, and determining

location information for more serious applications as well. A typical amateur location reporting system consists of GPS-packet equipped vehicles and fixed-location packet digipeaters. At a minimum, each vehicle would carry a GPS receiver, a packet unit (TNC), and a transmitter. Each TNC would be configured to record and beacon location periodically. The fixed location digipeaters would relay the reporting stations' locations to cover a wide area. For more details on setting up such a system, see the section, on "Advanced Digipeating."

GPS Equipment (or weather station) Requirements

To set up a packet GPS station that can be tracked, the following equipment is needed:

- a transceiver and antenna,
 - A TNC that supports the NMEA 0183 interface standard.
 - a GPS unit with an NMEA interface and portable or external antenna. Many vendors produce GPS units and most offer NMEA data interfacing which is essential.
 - Many weather station devices can output data that is in a format like GPS data. A TNC can capture and beacon weather condition reports, using the same GPS functions in the TNC.
-
- The Automatic Packet Reporting System (APRS), developed by Bob Bruninga defines many ways of operating this type of system. An APRS system may consist of a GPS unit, a computer, and (optionally) a packet unit.
 - Several software programs have been developed by amateurs to use packet and GPS to either beacon the position data, or to receive position reports from other stations and show those stations locations on a map display.
 - Some programs can take the place of the GPS functions in a TNC, using that TNC simply as an interface to the radio. In this case, the program would get the NMEA data from a GPS unit and would pass location reports through the TNC to be transmitted in UNPROTO packets.

Cabling a GPS Unit to a Kantronics TNC

The NMEA standard specifies that the signals in the GPS receiver data port follow the EIA-422 (or RS232) standard.

Technical note: some GPS devices vendors may use simple “TTL” voltage levels (+5 V and 0 V) but use the same polarity as RS232 or may have that polarity inverted (true TTL). The TNCs GPSPORT setting can accommodate either RS232 level data or TTL data of either polarity. The GPSPORT input is on pin 2 of the DSB-9 radio port.

To connect the data port of a GPS unit (or weather station device) to this TNC, the radio interface cable will need modification. A shielded cable is preferred. Refer to the manual for that device to determine which pins are used for signal ground, and data output. These two wires must be connected to appropriate pins on the DSUB-9:

- connect the signal ground wire of the GPS data port to pin 6 (or to pin 9 if the jumper inside the TNC sets pin 9 as ground). This pin on some GPS devices may be labeled as “NMEA-”.
- connect the “data output pin” of the GPS data port to pin 2 of the DSUB-9 radio port. This pin on some GPS devices may be labeled as “NMEA+”. This is the line that supplies the NMEA sentences to this TNC.

If GPS unit requires data input, for setting that device parameters, it may not be useable with this TNC.

Note: The default function for pin 2 on the TNC radio port is XCD (external carrier detect). If TNC radio port pin 2 has been connected to a GPS device, and GPSPORT has been configured for GPS input on this pin, the XCD function is not available.

Configuring a Kantronics TNC for GPS Operation

To configure the TNC for GPS data, several of its commands need to be set from the terminal, as follows:

- select the desired GPS strings to be captured in each of the Location Text buffers
- set the beacon rates for each Location Text buffer
- enter the unproto paths for each Location Text buffer
- configuring the GPS data input port

Use the TNCs GPSHEAD command to set which GPS strings will be captured and stored in the specified Location Text (LT) buffers. There are 4 available GPSHEAD selections, associated with 4 LT buffers.

For example: Enter GPSHEAD 1 \$GPGGA to capture that GPS data string.

The TNC will search all data from a GPS device for a GPS sentence that begins with \$GPGGA. When a complete sentence is detected, it will then store that sentence in LT buffer number 1.

The same applies to each of the other available buffers (GPSHEAD 2 and LT buffer 2, etc).

Each of the 4 buffers has its own BLT (beacon rate) setting, as well as an LTP (beacon PATH) setting.

If a GPSHEAD string for any LT buffer is empty, that LT buffer can be set up as an auxiliary beacon. They work the same as BEACON and its associated BTEXT but can have more flexible beacon interval timing and path settings.

The BLT (Beacon Location Text) command is set for each of the 4 buffers separately. The specified buffer number will be transmitted at every or after the specified interval, with an optional start time (see the slotted beacons note in the following section), and with an option to clear the buffer after it has been transmitted.

The interval time is in the format hh:mm:ss (for hours, minutes, and seconds). The hours and seconds intervals are optional.

The LTP command (Location Text Path) sets a destination address and a path for each of the four LT beacons, independently.

To use these beacons with the APRS digipeater network (typically on the 2-meter channel 144.39 MHz), the TNCs LTPath should be set to match UIDigipeaters that will be repeating these packets.

The TNC command GPSPORT configures that input on DSUB-9 pin2. It sets a baud to match the GPS data rate, with option setting normal or inverted polarity of that data, and if GPS sentence requires a CHECKSUM or not, for a sentence to be considered as valid data.

The data baud can be up to 9600 but must be set to match the GPS or weather station device. Most of them will use 4800 or less.

Other GPS Features

Slotted Beacons

The BLT command provides for the ability to specify not only the interval between beacons, but a starting time for the beacons. This permits a “slotted” system of

beacons by numerous stations. For instance, say a local area has 30 stations that each should beacon their location every 30 minutes. A beacon would be transmitted every n minutes from the time the user sets the time that the beacon should be transmitted. As an example, if the BLT command for LT 1 (BLT 1) is set with the command:

```
BLT 1 EVERY 00:30:00 START 00:00:01
```

the TNC would transmit a beacon starting at 00:00:01 (time after midnight) and every 30 minutes after that. A second user could set the BLT with the command:

```
BLT 1 EVERY 00:30:00 START 00:01:01
```

In this case, these two users would never beacon at the same time, in fact they would beacon 1 minute apart. Using this example, it is possible to set as many as thirty users to beacon one minute apart, avoiding collisions.

Tracking without Beacons

Using the TNC command LTRACK, an amount of memory can be designated as storage for copies of transmitted GPS beacons (any 1 or all 4 LTBUFFERS) at specified intervals, for later review.

This stored LTRACK data can then be retrieved when connected to the TNCs PBBS.

The TNC may be configured to just store the LT “beacons” in its LTRACK memory buffer without having transmitted those beacons. When LTP has been set to NONE, for an associated LT buffer:

```
LTP 1 NONE
```

the TNC will not transmit that beacon, but BLT setting for the associated buffer is still active, the data will be stored in the LTRACK buffer.

This can be used to keep a record of tracking data for local use when not needing to announce those reports to other stations.

Other Notes

The LT buffers, LT beacons and LTRACK buffer can be used even without a GPS unit attached. LT buffers can be used to store and/or beacon any desired text up

to 128 characters total. Just use the LT command to manually enter any desired text:

LT 1 This is ltext buffer number 1

The associated LT buffer will be transmitted based on the BLT and LTP commands for that buffer. Even if not actually transmitted, the buffer data will be stored according to the LTRACK command setting.

GPS Commands Summary

The following GPS commands are listed here for convenience and explained in detail in the Command Reference section.

BLT n {EVERY | AFTER} hh:mm:ss [START hh:mm:ss] | CLEAR (n = 1 - 4)

This command sets the interval between of beacons for the associated LT buffer number.

GPSHEAD n string (n=1-4) (string up to 8 chars)

This command determines which GPS NMEA sentence will be stored in each of the 4 LT buffers.

GPSINIT string (string up to 128 characters)

This command does not apply to sending an initialization string to a GPS device, in this KPC-3XE with a USB computer interface. When powered on or a soft reset occurs, the specified text string will be output on the computer interface port.

LT n text (n = 1-4) (text up to 128 chars)

This command fills the specified location Text (LT) buffer with “text”.

Entering the command LT alone, will simply display the content of the 4 TL buffers.

LTP n dest [via call1[,call2,...]] (n=1-4)

This command sets the destination callsign and any digipeaters to be used for transmitted LT strings.

LTRACK n [LT1] [LT2] [LT3] [LT4] [SYSOP] [TIME]

This command allocates memory for a special tracking buffer used to store LT messages. When configured, it can be accessed after connecting to the PBBS.

RPRINT text (text up to 128 characters)

This is an immediate command (remote print), which sends the following “text” string to the computer interface port of the TNC.

Advanced GPS (or APRS) Digipeating

The “UI” digipeating commands described in this section add flexibility in reporting GPS (or APRS) station positions. It is an established procedure of the reporting stations to digipeat with generic callsigns and through a number of repeaters (e.g., “APRS” repeaters). In this way, reports of their positions can reach out, via a number of digipeaters, to any wide area. At the same time, the digipeaters supporting these activities must establish their TNC parameters such that all UI frames “copied” are digipeated often enough (enough repeats of each frame), but not too often (too many repeats of each frame), to achieve the desired result.

The commands introduced here (i.e., UIDIGI, UIFLOOD, UITRACE, and UIWAIT) allow each digipeater owner a wide range of ways in which to configure the TNC (in cooperation with other digipeaters in a network) to establish an optimum GPS or APRS digipeating network.

- Note: This section assumes a familiarity with GPS, digipeating, and APRS (each topic is discussed elsewhere in this manual). For details about APRS (A Packet Reporting System) developed by Bob Bruninga, WB4APR, consider obtaining a copy of “Getting On Track with APRS,” by Stan Horzepa, 157-page paperback, published in 1996 by the ARRL, 225 Main Street, Newington, CT 06111.

Amateurs are using GPS receivers, a packet unit, and an FM transmitter to report the location of their vehicles or stations to others. Their broadcasts use one-way (unproto UI) packets, reporting callsign, latitude and longitude. Packet stations hearing these broadcasts can monitor the location of the vehicles.

A common goal of these position reporting stations is to digipeat a UI packet over a wide area (or a long distance), while moving through areas where the local callsigns may be unknown. An effective solution for not knowing local callsigns is to establish standard generic callsigns for digipeating stations and then use these names in digipeating paths. Of course, this requires the acceptance and use of

standards for naming and using digipeating stations, by both the position reporting stations and the digipeating stations.

- Common standards have been established by some local APRS groups, with names such as RELAY, WIDE, and TRACE. When unsure about the configuration of these settings, contact a local GPS (or APRS) interest group before setting up a TNCs digipeater capabilities. Usage may vary by area.

Improving Efficiencies of Advanced Digipeating

As described above, mobile location reporting stations use digipeating paths containing generic digipeating names (e.g., RELAY, WIDE, TRACE) to extend their range of reporting. Using generic digipeating enables mobile stations to use the same digipeating path while moving around, since there will (presumably) always be nearby GPS (or APRS) digipeating stations with one or more of these generic names to pick up transmissions (i.e., UI packets) and re-transmit them to a wider area.

Using generically named digipeaters does result in wider coverage of GPS stations, but several inefficiencies may arise from this common practice (depending upon the availability and names of the digipeaters).

Among the problems that may arise in this situation are that: (1) UI packets can be repeated much more than necessary, by the same or by different digipeating stations with the callsigns RELAY, WIDE, or TRACE clogging up the airwaves, and that (2) several stations may re-transmit the same packet at the same time, resulting in a combined signal that cannot be copied by other stations.

Kantronics' TNCs have a suite of UI digipeating commands that can be used to deal with inefficiencies that can arise when a number of digipeaters are transmitting in the same area, at the same time, using the same generic names.

Overview of UI Digipeating Commands

The following UI digipeating commands support advanced GPS (or APRS) digipeating capabilities.

The UIDIGI command may be used to set up to four additional aliases/call signs for "special" digipeating service. To-be-digipeated packets received containing one of these aliases will be repeated (once) with the call sign (MYCALL) of the digipeater substituted for the alias in the digipeated frame. See the example shown below and the command description for more detail.

UIFLOOD and UITRACE are a bit more exotic. Each provides multi-hop digipeating with just one digipeater address per packet, thereby keeping the transmission time short. For example, to digipeat through three TNCs supporting the UIFLOOD command, the reporting station might set a GPS position path as follows:

LTP 1 GPS via wide3-3.

A digipeater TNC supporting “wide” set by the UIFLOOD command and hearing the reporting station’s transmission would then digi the UI location packet (assuming it had not done so already, within a preset time), using an address of wide3-2. In turn each similar digipeater down line would digi the reporting station’s UI packet and reduce (decrement) the SSID of the digipeater address again. A TNC using UIFLOOD has the option of inserting MYCALL, creating two rather than one digipeater addresses in each transmitted digi packet. This will reduce unnecessary repetition.

With UITRACE, each time a packet is digipeated, each TNC adds its MYCALL, thus creating a “trace” or return path. In effect, the size of the packet grows by one digipeated address with each hop. A parameter, n, sets a delay time “do not digipeat within n seconds.” Again, see the examples below and the command descriptions for detail.

The current setting of UIDWAIT determines whether a delay is added to UI digipeat packets (those formed by UIDIGI, UIFLOOD, or UITRACE) before transmission, once the channel is clear. If UIDWAIT is ON, the delay is determined by the TNCs SLOTTIME and persist settings.

Each of these commands is documented in the “Command Reference” section of this manual.

Note: Digipeater priority for call signs is as follows: UIDIGI, UIFLOOD, UITRACE, MYCALL, MYNODE, MYALIAS (e.g. if the same callsign/name is assigned to UIDIGI and MYALIAS, a to-be-digipeated frame with that call sign will be digipeated according to the rules that apply to UIDIGI).

Using “UI” Digipeat Commands: UIDIGI, UIFLOOD, and UITRACE

The following are actual monitored outputs of a lab system consisting of one position reporting station and three digipeaters, where each digi is configured using the UIDIGI, UIFLOOD, or UITRACE command and where UIDWAIT is set ON. We set UIDWAIT ON to force the system to digipeat all or most of the UI frames without collision. In all examples, the reporting station, WØXI, is used to launch a UI packet with the path set by the UNPROTO command.

With UIDWAIT OFF, several digipeaters could transmit at the same time, resulting in packet collisions and lost data. In actual on-the-air use, a system of digipeaters may work well with UIDWAIT OFF, depending upon their mix of location and transmitter power.

- In actual use, GPS (or APRS) reporting stations would set their path with LTP.

Configuring Three Digis Using the UIDIGI Command

Here, three digis, with MYCALLs of A, B, and C, are configured with aliases of RELAY, WIDE, and TRACE (using UIDIGI); and UIDWAIT is set ON. For example, station A's UIDIGI aliases are set as follows:

```
cmd: UIDIGI ON RELAY, WIDE, TRACE
```

The reporting station path is then set to GPS via RELAY, WIDE, TRACE, and a UI packet is launched.

Any one of the stations monitoring will then display the resulting action as follows:

```
cmd: WØXI>GPS,RELAY,WIDE,TRACE: <UI>:5
WØXI>GPS,B*,WIDE,TRACE: <UI>:5
WØXI>GPS,A*,WIDE,TRACE: <UI>:5
WØXI>GPS,B,A*,TRACE: <UI>:5
WØXI>GPS,A,B*,TRACE: <UI>:5
WØXI>GPS,C*,WIDE,TRACE: <UI>:5
WØXI>GPS,B,C*,TRACE: <UI>:5
WØXI>GPS,A,C*,TRACE: <UI>:5
WØXI>GPS,B,A,C*: <UI>:5
WØXI>GPS,A,B,C*: <UI>:5
WØXI>GPS,C,A*,TRACE: <UI>:5
WØXI>GPS,B,C,A*: <UI>:5
WØXI>GPS,C,B*,TRACE: <UI>:5
WØXI>GPS,A,C,B*: <UI>:5
WØXI>GPS,C,A,B*: <UI>:5
WØXI>GPS,C,B,A*: <UI>:5
```

All digis hear the first UI frame and store it. Due to the setting of UIDWAIT, some stations may wait longer than others to digipeat; hence, when they do, they may have several frames stored up - for example, the original and a digipeat from another station. Trace station B as an example. It digipeats the original UI frame from the "reporting station," swapping its MYCALL for RELAY (B) and marking it (*), then repeats a frame from A and one from C, and, finally, repeats two more

(from A,C and C,A) —a total of five! Note that each repeater digis five times so the total number of digipeated packets is 15!

Configuring Three Digis Using the UIFLOOD Command

Here, three digis, with MYCALLs of A, B, and C, are configured with an alias of WIDE, and UIDWAIT is set ON. For example, station A's UIFLOOD call is set as follows:

```
cmd: UIFLOOD wide,30,ID
```

The reporting station path is then set to GPS via WIDE4-4, and a UI packet is launched. Any one of the stations monitoring will then display the resulting action as follows:

```
WØXI>GPS,WIDE4-4: <UI>:5  
WØXI>GPS,A*,WIDE4-3: <UI>:5  
WØXI>GPS,B*,WIDE4-3: <UI>:5  
WØXI>GPS,C*,WIDE4-3: <UI>:5
```

Note that all three digis see the UI frame addressed to GPS via WIDE4-4. They then, in turn digipeat that frame, inserting and marking as digipeated (*) their MYCALL, and include the new to-be-digipeated and decremented field of WIDE4-3. Note that each digi hears the UI frames repeated by the others but does not digipeat those in turn since a timeout of 30 seconds was specified by the UIFLOOD command. See the command specification for details.

Configuring Digis Using the UITRACE Command

Here, digipeaters A, B, C, and D are set with UITRACE trace, 30. Digipeater B then launches a UI frame using UN GPS via TRACE4-4. The following results:

```
B>GPS,TRACE4-4/1<UI>:QWER  
B>GPS,A*,TRACE4-3/1:<UI>:QWER  
B>GPS,D*,TRACE4-3/1:<UI>:QWER  
B>GPS,C*,TRACE4-3/1:<UI>:QWER
```

Stations A, D, and C hear B and digipeat the UI frame, inserting their MYCALL and decrementing the SSID of trace. A does not re-digipeat D or C because its “n” timer is set to 30 seconds, that is, it will not repeat any frame again within that time that includes the same source or destination and data (in this case QWER).

Configuring Digis for HF/VHF Gateway Operations

- This example assumes the use of at least one multi-port TNC that supports both HF and VHF operation.

In this example, we demonstrate cross-band (gateway) digipeating, using selected Kantronics' TNCs. A UI frame is launched on HF and repeated on VHF four times. A KAM XL and two KPC-3XEs (with call signs C, B, and D) are configured as digipeaters and another KAM XL (call sign A) is set to launch a UI frame from its HF port. The digis are configured with an alias of FLOOD and UIDWAIT is set ON. In addition, MYGATE for digi C (the KAM XL) is set to C-3 and its UIGATE is set ON/OFF; that is, it is enabled to digi from HF to VHF but not from VHF to HF. The other KAM XL, with call sign A, is then used to initiate an unproto packet with the path GPS via C-3, FLOOD4-4. Either KAM XL will then display the resulting action as follows:

```
A>GPS,C-3,FLOOD4-4/H: <UI>:234
A>GPS,C-3*,FLOOD4-4/V: <UI>:234
A>GPS,D*,FLOOD4-3/V: <UI>:234
A>GPS,C*,FLOOD4-2/V: <UI>:234
A>GPS,B*,FLOOD4-3/V: <UI>:234
```

Station C hears the HF unproto packet of station A and digipeats it via its GATEWAY out its VHF port. Digis B and D hear the VHF UI packet and store it for digipeating, and D transmits first. In turn station C hears station D's repeat of the frame and digipeats the frame, decrementing flood to "FLOOD4-2." It didn't digipeat its own "FLOOD4-4" frame because it can't hear itself! Station B finishes last due to the slot/persist algorithm. D digipeats the frame only once even though it hears both B's and C's transmissions since they are within 30 seconds of its own transmission.

Bibliography

For more information on the Global Positioning System, we recommend the following resources:

General Information

Bruninga, Bob, WB4APR, "Automatic packet reporting system (APRS)", *13th ARRL Digital Communications Conference Proceedings*, 1994. (address: ARRL, 225 Main Street, Newington, CT, 06111).

Horzepa, Stan, "Getting On Track with APRS," ARRL (225 Main Street, Newington, CT 06111) 1996.

Sproul, Keith, WU2Z, "MacAPRS", *13th ARRL Digital Communications Conference Proceedings*, 1994.

Technical Information

Collins, J., et al., *GPS Theory and Practice*, Second Edition, Springer-Verlag, 1993.

Etherington, Michael, "FM subcarrier network extends differential GPS nationwide", *Mobile Radio Technology*, February 1994.

Kaplan, Elliot, "The global positioning system (GPS)", *Communications Quarterly*, Summer 1994.

Langley, R., "The mathematics of GPS", *GPS World Magazine*, July/August 1991.

"NMEA 0183 ASCII Interface Standard", (NMEA 0183 version 2.0), NMEA, P.O. Box 50040, Mobile, AL 36605.

GPS Device manufacturers

Some of the more than 50 vendors currently supplying GPS units include:

- Garmin, 151st St, Olathe, KS 66062
- ICOM America, Inc., 2380 116th Ave. N.E., Bellevue, WA 98004.
- Magellan Systems Corp., 960 Overland CT., San Dimas, CA 91773.
- Sony Electronics Inc., 1 Sony Dr., Park Ridge, NJ 07656.
- Trimble - Mobile Computing Products, 645 N. Mary Ave., P.O. Box 3642, Sunnyvale, CA 94088-3642.
- And others.

KA-Node

Overview

Most Kantronics TNCs include the KA-Node. It is a packet networking node. When it is enabled, this packet station (unattended) can be used not only as a digipeater, but as a node. Other stations are then able to find pathways to other stations that are not directly reachable.

KA-Nodes, like other networking nodes such as NET/ROM, operate more efficiently than do digipeaters as a link between two stations. End-to-end acknowledgement of received packets is not required with the nodes; instead, they handle errors between each other, rather than from end to end (which can cause extra traffic when errors or interference occurs).

A KA-Node, however, is “silent” in that it does not automatically connect to and exchange routing data with adjacent nodes, as do fully featured nodes such as NET/ROM, X1J, or Kantronics’ K-Net. Consequently, users cannot make a connection to a distant station without knowing the path - as is possible when using a full-featured network node. At the same time, KA-Nodes are useful in that they are more efficient than digipeaters in using channel time, while not requiring the effort and time required to maintain a full node.

When packet got started in the early 1980s, the initial packet units - terminal node controllers (TNCs) - were designed and coded not only to accept station-to-station connects, but to act as digipeaters for other stations. It was the first attempt at packet radio networking, linking two stations together via several others. It soon became evident, however, that digipeating, particularly through busy channels, was an inefficient method of linking two stations via others. First of all, the station initiating the “connect” had no way to know ahead of time whether the digipeating stations were available. Second, the AX.25 protocol called for the station being connected to - at the end of several digipeaters - to acknowledge each packet of the initiating station. Packets not acknowledged (due to collisions) had to be retransmitted by the initiating station AND ALL DIGIPEATERS in the communications chain. As a result, communication often grounds to a halt when channels were busy.

To solve these problems, NET/ROM, a PROM-based networking program that was installed inside some TNC models, and several derivatives of NET/ROM (e.g., G8BPQ, X1J, KA-Node, K-Net) were developed. These networking programs

provide the user connecting to a station via one of these nodes with a choice of pathways to other packet stations and with lists of stations heard.

NET/ROM and several of its derivatives provide for automatic routing of connect requests - much like the phone system routes long-distance calls. Others, such as the KA-Node, assist in building a pathway by allowing a connection to each node in turn through the pathway. All the nodes correct the “end-to-end” acknowledgment problem mentioned above. This is accomplished by building the pathway with a number of individual “local” connecting links; that is, each link in the pathway is considered a connect and all acknowledgments of packets are kept local between each link. Hence, if only one link of a pathway (chain) is troublesome, the local nodes take care of it.

The following material is provided to assist in configuring a KA-Node and informing others how to connect to and use it. A listing of the KA-Node commands for users follows.

KA-Node Commands for Remote Users

When accessed by radio, the KA-Node supports several user commands. These commands are listed here for reference, with a description of each command. In these descriptions, at least the UPPER-case characters of the command are required, while LOWER case characters are optional. Those items listed within [] are optional and if used the UPPER/LOWER case convention applies.

ABORT

Aborts a KA-Node Connect request, if it is the first data sent after the connect request. It must be spelled out entirely.

Bye

Causes the KA-Node to initiate a disconnect.

Connect callsign [Stay]

Causes the node to issue a connect request to “callsign” in the usual AX.25 mode. If the connect is successful, a link will be made to the next node or end-user station called. The optional Stay feature provides a way to disconnect one link, without losing the entire link series. Normally when a disconnect occurs, whether issued by a local or by the remote station, the connection to the KA-Node is also lost.

Using the Stay option allows for staying connected to the KA-Node when the remote station issues a disconnect.

Help

Causes the KA-Node to send a brief help list, showing all commands available, with the options and a short description of each command.

Jheard [Short | Long]

Causes the KA-Node to return its TNCs MHEARD log. If the Short or Long option is not used, the returned list may look like this:

LAWKAN	01/09/88	08:25:15
N66046*	01/10/88	00:03:10
NØKN-3	01/10/88	00:03:19
NØGRG	01/10/88	00:04:15

The left column indicates the callsign (and SSID if appropriate) of a station heard. The asterisk indicates the station was heard via a digipeater. The center and right columns indicate date and time the station was last heard. The last call on the list will probably be the local TNCs call. The above JHEARD list shows NØGRG as the last station heard.

The short (Jheard S) version of this command would produce the same list, without dates and times.

The long (Jheard L) version of this command will also show the destination field and any digipeaters used.

Nodes [Short | Long]

Causes the node to return a list of other KA-Nodes it has heard, as well as network nodes such as; K-Net, G8BPQ, NET/ROM and TheNet nodes. The format of the list is like that of the JHEARD list above.

Configuring a KA-Node

The following commands are used to configure a KA-Node: NUMNODES, MYNODE, NDWILD, and KNTIMER.

To set up a TNC for use as a KA-Node several conditions must be met.

First, the number of circuits (channels) to be allowed through the KA-Node, are set with the TNC command NUMNODES. Each circuit consists of an “IN” and an “OUT” side.

Secondly, the callsign assigned to the KA-Node in the TNC command MYNODE must be different from all other callsign settings in this TNC (such as MYCALL, MYALIAS, MYPBBS, and MYREMOTE). By default, when this TNC was first initialized, it set its MYNODE callsign using the MYCALL entry with an added SSID of -7.

Setting the TNC command NDWILD to ON, will allow the KA-Node to accept a connect request to any SSID of its MYNODE callsign (except those mentioned above).

Packets passing through a KA-Node are displayed as monitored packets, unless MYNODE callsign has been added to SUPLIST when it is turned ON. If it becomes necessary to disconnect a station that is connected to the KA-Node, the TNC command DISCONNECT MYNODE x (x = channel) from the local terminal connected to this TNC. The status of connected channels in the TNC is shown by the TNC command STATUS.

The TNC command KNTIMER sets a “time-out” for the KA-Node. If no data passes through an established (connected) KA-Node circuit for n minutes, that circuit will be disconnected. The default time-out is 15 minutes. This timer can be disabled by setting n to 0.

Each KA-Node circuit allowed will require approximately 4.3 Kbytes of the TNCs internal memory. If an attempt is made to set NUMNODES to a value requiring more memory than is currently available, a message will be displayed indicating that the value is out of range and no change will be made. The total number of circuits which may be allocated will also be affected by the amount of memory allocated by other TNC functions.

Using a KA-Node

The material that follows is meant for stations using the KA-Node. It may be used as a guide to anyone using this node from another location.

Assuming that the KA-Node has already been enabled (the TNC command NUMNODES is set to a non-zero number), it is available to be used by other stations.

A connection must first be made to the KA-Node callsign as set in MYNODE. As an example, let's say the MYNODE has been named LAW. When a connection is made to LAW, the following message will be returned to a connecting station:

```
*** CONNECTED TO LAW
### CONNECTED TO NODE LAW (NØGRG) CHANNEL A
ENTER COMMAND B,C,J,N, or Help ?
```

The *** CONNECTED TO LAW message is returned to the connecting station. If this TNC's MYCALL was entered as "NØGRG", that callsign will be shown after the node name (LAW). CHANNEL A indicates that this connection has been assigned to channel A of the node. If channel A was already in use by another connection, channel B would be displayed. The channels, or circuits, are assigned by the KA-Node as needed. The last line of the KA-Node sign-on message is its command line.

If the TNC command NDWILD has been set to ON, the second line of the connected message would be shown as ###CONNECTED TO WILD NODE LAW...

After connecting to the KA-Node, the KA-Node is waiting for a command. The KA-Node commands are; B (for BYE to disconnect), C (with a callsign, to make a connection to that callsign), J (for stations heard by the KA-nodes TNC), N (for nodes heard by that TNC), Help (to receive descriptions of the node commands).

Use the command N (for nodes), to see a list of other nearby KA-Node stations that TNC has heard. For example, let's suppose that KC was heard. The list received from the Nodes command would be:

```
KC (NØAPJ-2) 12/23/25 02:38:45
ENTER COMMAND B,C,J,N, or Help ?
```

With a re-display of the nodes command line.

KC denotes the KA-Node callsign (MYNODE), with the MYCALL of the KA-Node TNC in parentheses, followed by date and time it was heard. If that TNC had not heard any nodes id packets, it would respond with:

```
NO KNOWN NODES
```

And a re-display of its command line.

Use the J (JHEARD) command to what other packet stations that KA-Node TNC has heard recently. The node will respond by listing the MHEARD log in that TNC.

After having connected to the KA-NODE LAW, to connect to another node named KC, just issue a connect request to KC as follows in response to the “enter command” from LAW:

CONNECT KC

The response will be:

```
###LINK MADE
###CONNECTED TO NODE KC (NØAPJ-2) CHANNEL A
ENTER COMMAND B,C,J,N,X, or Help ?
```

At this point a connection was made by LAW to the node KC. When LAW issued the connect request to KC it used the callsign of the original connecting station with one number subtracted from the SSID of that original connecting station. For example, if connected to LAW from WØXI, and WØXI instructed LAW to connect to KC, that connection would use the callsign WØXI-15 (WØXI-0 minus 1) when connecting to KC. This is automatic.

At this point, a further connection could be made connect to any other station within the range of node KC.

Note: If any connection is made from the KA-Node with the S (stay) option to another station’s call, when a disconnect happens with that other station, the last node (KC in this case) would remain connected to the previous connecting station.

If a connect command to another station has been sent to a KA-Node, but a mistake was made with the call, or to cancel that connection for any other reason, send the ABORT command. For the abort command to work, it MUST be the first word entered following the connect command and must be spelled out entirely. For instance, if a connect command was issued to the KA-Node:

C NOKN

and then it was decided to abort that connection, the next entry must be:

ABORT

Any other entry will cancel any possible abort of the connection, and KA-Node must retry out on its own, when that connection to NOKN fails.

If that connection to NØKN was successful, a normal packet QSO ensues. While everything appears “normal” and AX.25 compatible, acknowledgments for packets are done between the local station and the first KA-Node, then between the first and the second node, then between the last node and the NOKN station. Each link in the chain/pathway takes care of detecting errors. In other words, the link between KC and LAW handles its own error checking. In this way, one weak link will not cause end-to-end packets and acknowledgments to be repeated through each link, as they would when digipeating. The result is substantial improvement in throughput for connections using nodes.

When it comes time to disconnect, do so in the standard AX.25 manner. To disconnect the link described above, type <Ctrl+C>, obtain the cmd: prompt on the local TNC, and issue the disconnect command:

```
cmd: D
***DISCONNECTED
```

All links to the KA-Nodes and the station at the other end (NOKN), will disconnect. If however, that distant connection, in this case NØKN, disconnects the link, all linked stations will disconnect. An exception would be if the S (STAY) option was used with any connect request in a link connection, a disconnect would only happen between that end station, and the last KA-Node where the STAY option was used. Such as:

```
###DISCONNECTED BY NØAPJ AT NODE KC
ENTER COMMAND B,C,J,N, or Help ?
```

Frequently when connecting to a node in a distant city, there may be more than one station to which a connection is desired. Perhaps when a connection was made, a message was returned from that station, saying “I’m not here right now, please leave a message in my PBBS”.

By using the “S(tay)” option when telling a KA-Node to connect to another station, the KA-Node will not disconnect when it receives a disconnect from the distant station. It will, instead, return a message from that KA-Node saying
###DISCONNECTED BY (call) AT NODE (MYNODE).

Automatic Disconnect

If a user abandons a connection to a KA-Node or a link between two KA-Nodes without disconnecting and there is no activity through the link for a specified period of time (see KNTIMER), the node will initiate a disconnect.

Introduction to Basic Packet Networking

A network node is a "collection point" in a packet network. It is at these node locations that packets are collected and routed to other nodes and end-users. Many network nodes operate a "backbone" channel at higher data rates, usually at 9,600 or 19,200 baud. End-user access to the network is usually at a lower baud (1200 baud) and on a different frequency. The advantage of a network system is two-fold: automatic routing of connections to other network nodes and increased data throughput. A network node "knows" how to automatically route data through neighboring nodes to distant nodes on the network. They do this by "hearing" other nodes broadcast this information. This means that a user connected to a node does not need to specify the particular route to take in order to reach a distant node as long as the local node contains a listing for that distant node. Using a network node can also dramatically increase the data throughput to distant locations. The packet data and the acknowledgments travel from node to node (rather than from end-to-end), greatly reducing the inevitable data collisions that will occur when attempting to communicate over long distances.

The defacto standard networking protocol, NET/ROM, was developed in the late 1980's by Software 2000, Inc. Since then, other node derivatives such as TheNET, TheNET Plus, TheNET X-1J, and G8BPQ have evolved. Although these variants differ in their features, they all utilize the same basic networking protocol.

Some network nodes have more than one radio port in order to provide a high-speed network "backbone" on one frequency, as well as Local Area Network (LAN) access to the high-speed network at a lower data rate on a different frequency. This approach to packet networking makes good sense for a number of reasons. First, the high-speed backbone is free to pass large amounts of data without competing with end users for the channel. Generators of large amounts of data are called servers. Many times, servers are busy communicating with other servers, so dedicating a channel to this function is desirable. Second, users are "shielded" from the server-to-server activity. Many users could peacefully co-exist with a server if all the server did was to respond to user queries. User input to a server generally consists of very small amounts of data that generate fairly large responses. In a local area, this works quite well since all users can detect the presence of the server's data. Carrier Sense Multiple Access (CSMA) allows the channel to be shared nicely. However, servers, and in particular many Bulletin Board Systems (BBSs), spend much more time forwarding and receiving messages from neighboring bulletin board systems than they do actually serving the local user population! It is not unusual for a BBS to attempt forwarding (*and* reverse forwarding) on an *hourly* basis. This situation quickly evolves into quite a

paradox - BBSs provide an extremely popular service that has been a boon to the growth and popularity of packet radio. As user numbers increase to take advantage of the services a BBS provides, BBS systems also increase in number in order to provide the service to more and more users. More and more BBS systems must now forward to each other, generating huge amounts of data on the channel, leaving precious little time for the very source of their existence - the user community. User-to-user (keyboarding) QSOs become, at best, a very frustrating proposition. As a result, many users become ex-users, interest declines, and packet usage stagnates. With the K-Net function enabled in the TNC, cost-effective networking can benefit all packet users in the area.

Basic Networking Guidelines

The intent of this section is not to dictate how to configure a packet network. However, certain goals and concepts of design should be carefully considered. The primary goal of any data network should be the efficient transfer of data from point A to point B (high speed), coupled with ease of use (low frustration level). The efficient transfer of data is made possible with the advent of high-speed modems and "data-ready" radios capable of operating at 9,600 and 19,200 baud. These modems and radios are now available from a number of commercial sources. This means that making internal radio modifications for high-speed data operation is quickly becoming a thing of the past.

User access to the high-speed channel is generally accomplished by using a separate, low-speed port at a network node site on a different channel. This protects users from server-to-server activity and also provides a relatively clutter-free Local Area Network for keyboard-to-keyboard activities.

In an ideal packet network, all high-speed links would operate on a dedicated channel (backbone) without contention. This would require that no stations other than network nodes and servers reside on the backbone channel. Each node on the backbone would have a "rock solid" link to the next, preferably on alternating, or full-duplex channels for maximum data transfer efficiency. All users in a community would be grouped into Local Area Networks (LANs) with each LAN assigned a specific operating frequency such that adjacent LANs would not interfere with each other. Each LAN would have a user port at a local network node that would provide access to servers and other LANs in the region. This situation would provide very efficient server-to-server communication, while offering users an enjoyable environment consisting of a usable keyboard-to-keyboard channel and easy access to servers and distant LANs. In reality, much of the above is quite difficult to achieve. Obstacles include finding suitable node sites, available funds for equipment, and dealing with the differing philosophies and interests of other node sysops. There are many differing opinions on how to

go about achieving "Network Utopia", but virtually all of them have one goal in common - make packet radio fun!

K-Net Network node

Quick Start

The factory default setting in the TNC has the K-Net network node function disabled. To enable the K-Net node, the NETALIAS and the NETCALL must be entered.

The NETALIAS is usually a mnemonic that gives users an idea about where the node is located (other network nodes do not care what the alias is). For example, SUTNE is a good NETALIAS for Sutton, Nebraska.

The NETCALL is the callsign of the node. It consists of a valid callsign plus an optional SSID (Example - KA0DNV-10). It must also be different from any other callsign used in this TNC (MYPBBS, MYALIAS, MYNODE, etc.). Use the DISPLAY ID command to make sure that the NETCALL entered is different from this TNC's other callsigns. Many times, local "customs" will serve as guidelines for what NETALIAS and NETCALL to use.

Enter the NET ALIAS:

```
cmd: NETALIAS SUTNE
```

Set the NETCALL:

```
cmd: NETCALL KA0DNV-10
```

This command will cause the TNC to perform a soft reset as memory is allocated for node operation. The K-Net node is now in operation!

There are six additional network node commands that can only be accessed when at the TNC's regular command prompt. These commands and their current settings can be displayed with the DISP N command.

```
cmd: DISPLAY N <CR>
```

```
NETBUFFS 32
```

```
NETCIRCS 5
```

```
NETDESTS 25
```

```
NETLINKS 10
```

```
NETROUTE 5
```

NETUSERS 5

Other node commands are accessed only after connecting to the node, either locally with a local terminal, or remotely from another packet station.

When connected to the node from a local terminal, Sysop access is automatic.

If connected remotely, sysop access (see SYSOP command) is required to list and change sysop related parameters. NOTE: the RTEXT command must be set to allow remote sysop access. Following the above example:

```
cmd: C SUTNE <CR>
```

This will cause the TNC to do an "internal connect" to the K-Net node alias SUTNE, much like connecting to this TNCs PBBS. When connecting to the K-Net node in this manner, a node command prompt (cmd:) will not be automatically displayed. Hit the Enter key to see the node commands available.

To see the current setting of any of the node's commands, just press the enter key. When at this "invisible command prompt" all node commands (including "SYSOP" only commands) are accessible.

The entire Help contents can be displayed by entering H H<CR>.

A short help description of any node command is also available by entering H(elp) or ? followed by a node command (Example - To get Help for the ADDNODE command, enter H ADDNODE<CR> or? ADDNODE<CR>.

Text Messages

Now that the node is operating, it is desirable to enter some text messages that will provide node users with some helpful information. These messages can be entered after having connected to the node (cmd: C SUTNE) and entering one of those commands followed by the text. Those commands that allow setting text messages are CTEXT, INFO, and PORTS. Their descriptions are in the following Commands Section.

K-Net Node Commands

An asterisk (*) preceding a command indicates a sysop command. All other commands are available to users of the node.

NOTICE - in order to gracefully integrate a K-Net node into an existing network, it is *imperative* to coordinate with existing network sysops. There are numerous

node parameters that can seriously degrade network performance if set to conflicting values.

Here are some networking terms that should be well understood by any network node operator.

NODE - The term node refers to a network node such as a G8BPQ, X1-J, a NET/ROM node, a THE-NET NODE, or this K-Net node. Non-network nodes such as KA-Node, conference bridges, and digipeaters do not apply to this concept of operation.

USER - A user is either a real, live person using a TNC to connect to and use a node, or a server (BBS) that is basically doing the same thing under computer control.

NEIGHBOR NODE - A neighbor node is a node that this node can hear and connect to "direct", without using an intermediate node. This means that a node can be "out of range" but can still be considered a neighbor since the connect path does not use any other nodes. All known neighbor nodes are displayed with the ROUTES command.

DISTANT NODE - A distant node is a node that is too far away to connect to "direct", requiring the use of one or more network nodes to reach. Distant nodes will *not* be displayed with the ROUTES command.

DESTINATION NODE - A destination node is a node that this node knows how to connect to, either directly (i.e. neighbor node), or, in the case of a distant node, by using one or more intermediate network nodes. All known destination nodes (neighbor nodes and distant nodes) are displayed with the NODES command,

All neighbor nodes are destination nodes, but all destination nodes are not necessarily neighbors. Destination nodes will automatically appear in the nodes table by virtue of this node monitoring neighbor nodes node broadcasts. A neighbor node will automatically be listed in the nodes table and routes table. Neighbor node broadcasts also contain information about other nodes that this node may or may not be able to hear "direct". These distant nodes will also be listed in its nodes table if they pass the criteria for being listed (MINQUAL), and if there is sufficient room for them to be listed (NETDEST).

ADDNODE [alias:]call port neighbor [via digi1[,digi2]] quality [obscent]

This command is used to add a new destination node or to modify an existing destination node in the nodes table. If the optional obscent (obsolescence count) is

set to 0, the node will be *permanently* listed in the nodes table. Such a route will not be updated or dropped from the nodes table by the K-Net auto-routing logic. If the neighbor node callsign entered is not in the neighbor routes table, it is added with the default quality for the specified port. As long as MINQUAL is equal to or less than QUALITY, destination nodes (neighbor nodes as well as distant nodes) will be added automatically to the nodes table as neighbor node broadcasts are received.

EXAMPLE - Addnode KSLVN:WDOEMR-1 1 WK5M-1 200

K-Net responds: Node added with new route

This adds the KSLVN node with the netcall of WDOEMR-1 to the nodes table. KSLVN is set as an available (assumed to be "reachable") node using the WK5M-1 (netcall) node as a neighbor node. An initial quality of 200 has been assigned to the new node. Unless "locked in" with the ADDROUTE command, this path quality will most likely change since it is recalculated based on node broadcasts heard. In this case, WK5M-1 was a previously unknown neighbor node, and the new route was automatically added.

EXAMPLE - Addnode WLAW:WOXI-1 1 WOXI-1 200

K-Net responds: Node added with new route

This would add the WLAW node with the netcall of WOXI-1 to the nodes table. In this case, WOXI-1 is also the neighbor node callsign, and is available with an initially assigned quality of 200.

EXAMPLE - Addnode YORK:WAOCPS-1 1 WK5M-1 70

K-Net responds: Node added

This would add the YORK node with the netcall of WAOCPS-1 to the nodes table. Since WK5M-1 is an existing route (neighbor node) a new route is not added.

EXAMPLE - Addnode CRT:KG5DT-1 1 WK5M-1 700

K-Net responds: Node added

In this case, the optional obscnt (obsolescence count) has been specified as zero. This special case is useful in the event that this node is not able to hear this neighbor's node broadcasts (i.e. if a digipeater(s) or gateway is being used). Normally, if a node is not heard from for a period of time, the obsolescence

counter will eventually drop it out of the nodes table. However, this node (CRT) will remain in the nodes table permanently, unless manually removed by a sysop.

ADDRROUTE port call [via digi1[,digi2]] quality [!]

This command will add (or modify) the specified netcall of a neighbor node to the routes table. If the optional '!' is specified, the route entry is either locked, if unlocked, or unlocked if locked (toggle). An existing locked route does not have to be unlocked before it can be modified. Locking a neighbor's route is used to assure that an assigned quality will always exist for that neighbor. The locked quality will also be used in the automatic quality computation of other distant nodes contained in this neighbor's node broadcasts to determine if they should also be added to the nodes table in this node.

In the event that this locked neighbor disappears from the network, and then returns, it will automatically be assigned the locked quality. During the time that this neighbor is absent, it, and distant nodes associated with this neighbor, may eventually be removed from the nodes table in this node (even if the route is locked), because the obsolescence count for this neighbor could drop to zero. If this neighbor's route is locked, it will still be listed in the routes table, but with a zero count for destination nodes using that route.

EXAMPLE - ADDRRoute 1 WOXI-1 110 !

K-Net responds: Route added and locked

This adds the route to the WOXI-1 node to the routes table in the K-Net node. The route is "locked in" and assigned to port 1 with a quality of 110.

Locking a route assigns a fixed *quality* to a neighbor node. This can ensure that any connects to a distant node using this neighbor node as a route will take the "hard coded" route, even if the route automatically generated by the K-Net node is different, such as may occur during band openings.

EXAMPLE - ADD Route 1 WOXI-1 110 !

K-Net responds: Route modified and unlocked

Now the WOXI-1 node is "unlocked" since the '!' is a toggle function.

EXAMPLE - ADD Route 1 W4NGA-1 0 !

K-Net responds: Route added and locked

This adds the W4NGA-1 node to the routes table with a locked quality of 0. This node, and any distant nodes associated with it, will not appear in the nodes table since the locked in quality of zero is probably less than MINQUAL.

NOTE - The W4NGA-1 node can be *completely* ignored by putting its NETCALL (W4NGA-1) in the TNCs SUPCALLS list and turning LLIST ON (from the TNC command prompt - cmd:). This would prohibit the K-Net node from connecting to as well as accepting a connect from the W4NGA-1 node.

BBS [/S]

This command connects a user of the node to the PBBS TNC of this node. If the TNCs PBBS is not enabled, this command is not operational and will not appear in the node's Help list. If the optional S(tay) parameter is used, the user that entered the BBS command will be returned to the node after sending the B(ye) command to the PBBS. This allows a user that is connected to the node, to then access the PBBS in the same TNC, to read or send messages, the same as anyone else that may connect directly to that PBBS.

BYE

This command causes the node to send a disconnect packet to the user.

CONNECT [call | alias] [/S]

The CONNECT command is used to establish a connection to another node or to another user/server. If connecting to another node that is listed in the nodes table (see NODES command), the node will automatically select the best path to use. A connect using the optional /S will use the Stay option - if the remote station issues a disconnect, the node connection will still exist.

If a node user wishes to connect to this TNCs MYCALL and enters just a C (or C /S to Stay), the node will attempt to "connect" internally to the sysop (MYCALL) of this TNC. The local terminal connected to this TNC will allow that communication just like a traditional TNC to TNC packet session. If this TNCs CMSG command has been set to PBBS, that user connecting through the node, will be routed to the PBBS instead.

CQ [text]**(up to 128 characters)**

This command places the user in the nodes CQ mode. That users text is then transmitted by the node on the TNCs radio port, as long as the node command CQBC is set to ON. If another user tries to connect to the callsign of the user contained in the CQ packet during the next 15 minutes, the node will then establish a link back to the user initiating the CQ. Any other text sent by the user that does not begin with CQ will cancel the CQ mode.

EXAMPLE - If a user (WAOCP5) connects to the node and gives the command:

CQ This is Ed in York, Nebraska - Please connect to me for a short QSO.<CR>

The node will transmit a packet addressed *to* CQ *from* the node-modified callsign of WAOCP5-15.

WAOCP5-15>CQ: This is Ed in York, Nebraska - Please connect to me for a short QSO.

The node automatically subtracts Ed's SSID (0) from 15 (15 minus 0 = 15) since the callsign WAOCP5 is already being used in the circuit. If Ed's callsign was WAOCP5-2, the node would modify it to WAOCP5-13 (15 minus 2 = 13).

If another user tries to connect to WAOCP5-15, the node recognizes the connect request and the two users are both connected through the node.

CQBC ON | OFF

Default ON

This command controls if the node will allow a CQ broadcast from a user. By default, the CQ broadcast is allowed. For example, if it is desired to not allow any CQ broadcasts to take place through the node, set the CQBC to OFF.

CTEXT text**(up to 128 characters)**

Default blank

This command sets a text message that is sent to users when they connect to the node alias (NETALIAS), but not when a user connects to the node callsign (NETCALL). CTEXT is also not sent when a user connects to the node through another network node.

EXAMPLE - CText Welcome to the SUTNE K-Net node, operated by KA0DNV, in Sutton, Nebraska

Ctext can be cleared by entering CTEXT %.

DELNODE [alias:]call port neighbor [via digi1[,digi2]]

This command is used to delete a destination node from the nodes table. If the node being deleted has an alias (showing with the NODES command), both the alias and the call of the node must be specified. If this node is the only one using the specified neighbor as a route, and if the neighbor is not locked in the routes table, the neighbor will also be deleted from the routes table.

DELROUTE port call [via digi1[,digi2]] quality

The DELROUTE command is used to delete a neighbor from the routes table. A neighbor route cannot be deleted if there are any nodes in the nodes table using this neighbor as a route.

EXAMPLE - DELRoute 1 WOXI-1 110

K-Net responds: Route deleted

EXAMPLE - DELRoute 1 WK5M-1 110

K-Net responds: Route unlocked

In this case, the route was not deleted, and the "Route unlocked" response simply indicates that the route is not locked in the routes table. This response will be received if this neighbor is used as a route by other destination nodes in the nodes table. In order to delete this route, *all* destination nodes that use this neighbor, as a route, must be deleted first.

HELP

This command displays a list of all commands available to the connected user. Giving the HELP command followed by an available node command will display a one-line help message about that command. Entering the HELP HELP command will give a one-line help message of all available commands.

IDINT n**(n = 0 - 255)**

Default 10

The IDINT command sets the time interval (in minutes) between node ID packet beacons (*not* node broadcasts). Setting this command to 0 disables node IDs.

INFO text**(up to 128 characters)**

Default blank

This command sets a text message that is sent to users when they enter the INFO command after connecting to the node.

EXAMPLE - INFO Located 3 miles N. W. of Sutton, NE - 50 watts at 100 feet. Hit Enter for available commands or C to Connect to my local terminal.

The INFO text entry can be cleared entering the command INFO '%'.

LINKS

The LINKS command returns a list of current AX.25 links existing in the node. It shows the callsign of the two connected stations in a link, the AX.25 State (normally 5), the Type of link (U = Uplink, D = Downlink, and X = Crosslink), and the AX.25 Version in use (1 or 2).

EXAMPLE - Links

K-Net responds: WK5M KARL S=5 P=1 T=U V=2

This shows that WK5M is connected to KARL, the AX.25 state is 5 (connected), using port 1, it is an *uplink* using AX.25 version 2. Note: An *uplink* is an end user/server connected *to* the node, a *downlink* is *from* the node to an end user/server, and a *crosslink* is from the node to another node.

Possible State Values (S) are:

- 1 Disconnected
- 2 Connect in progress
- 3 FRMR condition detected
- 4 Disconnect attempt in progress

- 5 Connected
- 6 Waiting acknowledgment
- 7 Device busy
- 8 Remote device busy
- 9 Both devices busy
- 10 Waiting acknowledgment and device busy
- 11 Waiting acknowledgment and remote device busy
- 12 Waiting acknowledgment and both devices busy

The next six commands deal with the ISO OSI (Open Systems Interconnection) Reference Model of network structure, in particular level 3 (Network Layer), and level 4 (Transport Layer).

L3TTL n (n = 0 - 255)

Default 25

This command (Level-3 Time-To-Live) sets the limit on the number of times (in a counter) a packet originated by an uplink (normally a user) at this node may be transferred through other nodes. Each time a message is handled by a node, the time-to-live for that message is decremented. If the L3TTL counter reaches zero, the message is discarded. It is possible for network nodes to create circular paths, and therefore a limit is required to prevent a packet from relaying in the node system forever. L3TTL should be standardized network wide.

L4DELAY n (n = 1 - 60)

Default 5

This command (Level 4 Delay) sets the delay time in seconds used by the node after receiving a level 4 frame from another node before the ack is returned. Frequently there will be data ready to be passed in the opposite direction shortly after receiving a frame, and this delay permits the acknowledgment to be "piggy-backed" with the data frame, thus reducing the number of transmissions on the channel.

L4LIMIT n**(n = 0 - 65,535)**

Default 900

This command (Level 4 Limit) sets a no activity timeout (in seconds) on a node crosslink. When a user connects to another node, the node first establishes a level 2 (AX.25) connect with that node and then establishes a level 4 circuit on behalf of that user. When the user is finished, the level 4 crosslink is closed, but the level 2 connection remains until the L4LIMIT is reached, at which point that connection is terminated. When a user disconnects, the level 2 connection between the nodes remains intact and the L4LIMIT timer is restarted. This decreases data congestion if another crosslink request using the same path occurs before the L4LIMIT expires since the node will not have to re-establish the level 2 connect.

L4N2 n**(n = 1 - 127)**

Default 3

This command sets the number of retries used to pass data between nodes (level 4) in the system. After the L4N2 count reaches zero, alternate paths will be tried.

L4T1 n**(n = 5 - 600)**

Default 120

The L4TI timer sets the time in seconds that the originating node will wait before resending a level 4 data packet if no acknowledgment is received from the destination node (end-to-end info-ack). If set too small, multiple copies of the same packet will exist within the node system even if the first packet was received correctly.

L4WINDOW n**(n = 1 - 127)**

Default 4

The L4WINDOW sets the maximum number of outstanding frames that may exist in the network for a given circuit. When the node has L4WINDOW number of unacknowledged frames, it will not transmit any more frames until a level 4 info-ack is received. Network nodes may allow up to 127 outstanding frames across the system. The L4WINDOW size is negotiated between nodes when a crosslink is requested, therefore the L4WINDOW size should be standardized network wide.

MHEARD [S | L]

This command displays a list of stations recently heard by the node. It has the options to receive a short list or a long list. MHEARD alone, will return a list of callsigns heard with a time stamp for each. If the Short option is specified, the list contains only callsigns heard. If Long is specified, the node displays the *to and from* callsigns of the stations heard, a time stamp for each one, as well as any digipeaters heard with each station.

MINQUAL n

(n = 0 - 255)

Default 70

This command sets the minimum quality required to add a destination node to the nodes table. Making this value higher will decrease the size of the nodes table, making this value lower will increase the number of destination nodes listed. When K-Net hears neighbor node (A) transmit a node broadcast, it computes the quality to distant nodes (B, C, D, etc.) contained in that broadcast by using the quality that is assigned to neighbor node (A). If the resultant computation is less than MINQUAL, the distant nodes (B, C, D, etc.) are not added to the nodes table (see QUALITY command). The default settings of QUALITY = 255/70 and MINQUAL = 70 means that all nodes that are heard "direct" (neighbors) will be added to the nodes table. If MINQUAL is greater than QUALITY, a new neighbor node will not be automatically added to the nodes table, nor will it be displayed with the ROUTES command.

NODES [* | alias | call]

This command displays a list of all known destination nodes. The list displays the ALIAS:CALLSIGN for each destination node. If the optional * is given, hidden nodes (those beginning with #) are also shown in the nodes list. Specifying an alias or call with the NODES command will display up to three possible routes (neighbor nodes) of descending quality that will be used to establish a link to the specified destination.

EXAMPLE - N YORK

K-Net responds:

Routes to YORK:WAOCPs-1 14051 WAOCPs-1 865 1 KA0DNV-1 753 1 W0XI-1

This list shows the three best routes K-Net will attempt to use (in descending order) to connect to the YORK node. The first number is the route quality, the

second number is the obsolescence count, and the third number is the assigned port.

NODESINT n

(n = 0 - 255)

Default 60

This command sets the time interval (in minutes) of nodes broadcasts (not node IDs). A new setting will be in effect after the previous interval has expired. This interval should be set to an agreed-upon value for all nodes in the area. This broadcast includes neighbor nodes with update information, which they use to add, modify, and delete nodes automatically from their nodes and routes tables. Each time this node performs a nodes broadcast, the obsolescence counter for all destination nodes in its nodes table is decremented by one.

OBSINIT n

(n = 0 - 255)

Default 5

The OBSINIT command sets the initial obsolescence count assigned to a destination node. When a nodes broadcast is heard from a neighbor, or when a link is established with a destination node, the obsolescence count is set to OBSINIT for that destination node. Each time this node transmits a nodes broadcast, the obsolescence for each destination node is decremented by one. When the obsolescence value drops below OBSMIN for any destination node, that destination node will not be contained in the nodes broadcast. However, it will remain in the nodes table until the obsolescence count reaches zero, at which time it will be removed, unless that node had been added (ADDNODE) with an initial obsolescence count of zero.

OBSMIN n

(n = 1 - 255)

Default 4

This command sets the minimum obsolescence count, to allow a destination node to be included in the nodes broadcast. The default values of OBSINIT 5 and OBSMIN 4 allow this node 2 chances to receive a "refreshing" nodes broadcast or a connect from a destination node before that destination node is dropped from its nodes broadcast. When the obsolescence count for a node reaches 0, that node is dropped from the nodes table unless it had been added or modified (ADDNODE) previously with an initial obsolescence count of zero.

PORTS text**(up to 128 characters)**

Default blank

This command sets a text message that is sent to users in response to the PORTS command, after connecting to the node.

EXAMPLE - Ports 145.01 MHz

The text in the PORTS command can be cleared by entering PORTS %.

QUALITY m/n**(n = 0 - 255)**

Default 255/70

This is a two-port command (Port 0 = computer interface port, Port 1 = radio port) that assigns the quality that is automatically given to a new neighbor node when its nodes broadcast is heard. It is also used to calculate the path quality to distant nodes that are listed in the nodes broadcast from a neighbor node. Port 0 quality defaults to the highest value possible (255) because a back-to-back computer interface port path is extremely reliable (see Node Stacking section). The quality value is important for two reasons.

The quality of a node (either automatically or manually assigned) will determine how far away that node will be visible in the node network. Proper use of the QUALITY parameter is necessary to maintain a *useable* node listing by not allowing a node to propagate useless distances over the network.

The QUALITY command can also be used to ensure that known "good" routes are chosen by the auto-routing routines. In the event of a band opening, it is possible for distant nodes to be heard "direct" and be automatically assigned a higher quality than normally calculated by using "good" neighbor routes. If a user attempts to connect to this listed distant node, the first-choice route (direct) will probably fail, and then the second best choice (which may actually be the most *reliable* route) will be tried.

The K-Net node will automatically calculate the quality to distant nodes based on the quality that is received from neighbor nodes when neighbor nodes broadcast their known nodes. If this node hears a neighbor node (A) broadcast a quality of 70 for a distant node (B), this node will automatically determine the quality of the path to distant node (B) that will use neighbor (A) in the route. This is done by multiplying the quality of the route assigned by this node to neighbor node (A) by the quality of the distant node (B) received within the nodes broadcast from

neighbor (A) and dividing by 256. As an example, if this node has assigned a quality of 70 to neighbor (A), and receives a nodes broadcast from neighbor (A) containing a quality of 70 to distant node (B), this node will assign a quality of 19 to distant node (B) (70 multiplied by 70 divided by 256 and dropping any fraction of a whole number). If MINQUAL is set to any value greater than 19, distant node (B) will not be added to the nodes table in this node. On the other hand, if the route quality to neighbor node (A) had been modified with a higher value, such as 200, a quality of 54 would then be assigned to distant node (B) (70 multiplied by 200 divided by 256). In this case, if MINQUAL is set to 54, this distant node would be listed in the nodes table in this node. After determining what quality works best for a neighbor node, "locking" the quality (!) will ensure that the desired quality will always exist for that neighbor - even if that neighbor disappears from the network and then returns.

ROUTES

The ROUTES command will display the list of *neighbor* nodes. For a new neighbor to be contained in the nodes and routes tables, MINQUAL must be equal to or less than QUALITY. The routes table shows the port number, the neighbor node callsign, quality value, the number of destination nodes in the nodes table that use this neighbor as a route, if the route is in use, and if the route is locked. For example:

R <CR>

KARL:WK5M-1}Routes:

```
>1 KA5ZTX-1 110 1
1 WD0EMR-1 63 7
>1 KA0DNV-1 85 15
1 WA0CPS-1 110 12 !
1 W4NGA-1 0 0 !
```

This Routes listing shows that the KARL:WK5M-1 node can reach neighbor node (KA5ZTX-1) on port 1 with a quality of 110. There is only one destination node (itself - KA5ZTX-1) in Karl's nodes table that can be reached using this neighbor. A right arrow (>) indicates that this route is either in use or has been used within the L4LIMIT time period.

WDOEMR-1 is also a neighbor node for WK5M-1. This node has a port 1 quality of 63. Using the WDOEMR-1 node as a route, Karl's node can reach seven of the destination nodes listed in its nodes table (WDOEMR-1 plus 6 others),

The KAODNV-1 node has a path that is available to WK5M-1 on port 1 of the K-Net node. The path quality is 85 with 15 destination nodes in Karl's nodes table able to be reached using this neighbor as a route (KAODNV-1 plus 14 others).

The next neighbor node is WAOCPS-1. It is reachable on port 1 with a quality of 110. There are 12 destination nodes in Karl's nodes table that can be reached using this neighbor as a route (WAOCPS-1 and 11 others). The exclamation point (!) indicates that the route as well as the route quality (110) has been "locked in" by the sysop of the WK5M-1 node.

The last neighbor node listed is W4NGA-1. Since the quality has been set and locked to zero, there are no destination nodes (including W4NGA-1) listed in Karl's nodes table that can be reached using this neighbor as a route. This node will not be listed in Karl's nodes table, and all node broadcasts sourced from this node will be ignored.

STATS

This command displays a summary of statistics concerning level 3 and level 4 activity on this node. A typical display:

KARL:WK5M-1}

Time active 3986 min

Buffers: 200 max, 199 available, 169 min, 0 out

Known nodes: 19

L4 Connects: 26 sent, 5 rcvd

L4 Frames: 205 sent, 392 rcvd, 6 resent, 0 resequenced

L3 Frames Relayed: 2303

This shows that the node has been active for 3986 minutes, there are 200 buffers in the node, 199 of those buffers are available. In addition, it indicates that, at some time, there were only 169 buffers available, and that the node has never run out of buffers. If min Buffers nears zero, or if the buffers have run out, the NETBUFFS parameter should be increased. It further shows that there are 19

nodes listed in the nodes table, there have been 26 level 4 connects sent and 5 received. 205 level 4 frames have been sent, 392 received, 6 resent (retries) and none have been resequenced. The level 3 activity shows that 2303 level 3 frames have been relayed.

SYSOP

The SYSOP command can be used by a remote user to gain access to sysop only commands (marked with a single *). When a user gives the SYSOP command, the node responds with three lines of numbers. The user must then convert one of these three lines into the corresponding characters of the RTEXT to be validated as sysop. This operates the same as the remote access (connection to the MYREMOTE callsign) in the TNC and uses the same RTEXT. (See the "Remote Access" section in the TNC Manual), If connecting to the node with an attached terminal, sysop validation is automatic.

USERS

This command displays a list of users currently connected to the node. It also displays the number of buffers currently available (in parentheses). Uplinks, downlinks, and crosslinks will be displayed by this command.

EXAMPLE -

Node user enters: USERS

K-Net responds:

KSLVN:WDOEMR-1} K-Net Packet Switch v1.0 (145)

Uplink 1 (WAOPWS) <-> Circuit (KSBLC:WZOM-3 WAOPWS)

Circuit (KSLAW:WK5M-1 WAOCPS) <-> Downlink 1 (WAOCPS-15 KBOVA) Uplink 1 (KAODNV)

KSLVN:WDOEMR-1} K-Net Packet Switch v1.0 (145)

There are currently 145 buffers available (145) at the KSLVN:WDOEMR-1 node.

Uplink 1 (WAOPWS) <-> Circuit (KSBLC:WZOM-3 WAOPWS)

WAOPWS is a user that has connected (Uplinked) on port 1 to the KSLVN node. WAOPWS has requested, and KSLVN has established a Circuit (Crosslink) to the KSBLC:WZOM-3 node on behalf of WAOPWS.

Circuit (KSLAW:WK5M-1 WAOCPS) <-> Downlink 1 (WAOCPS-15 KBOVA)

A Circuit, (Crosslink) from KSLAW to KSLVN has been established by the request of WAOCPS with WAOCPS-15 (node modified callsign) then Downlinking to KBOVA on port 1.

Uplink 1 (KAODNV)

KAODNV has Uplinked to KSLVN on port 1 but has not established a Crosslink or Downlink at this time.

Fine-tuning the K-Net Node

Although the default parameters of the K-Net node will get it "up and running", there are some guidelines that can enhance the desired operation. Good neighbor relationships are really the key to good networks. This takes the form of proper quality settings for neighbor nodes, as well as good relationships with neighbor sysops. If the nodes table in this node has a consistent listing of destination nodes that have reliable paths, users will not be frustrated, and this node will be perceived as working quite well. On the other hand, if its nodes table is frequently allowed to accumulate a large number of destination nodes that are not fairly solid links, it becomes difficult to use, non-responsive, and very unfriendly to the network and especially to a new user.

After the K-Net node has been running for a while using the default parameters, some "tweaking" of commands may be in order. *The quality settings for neighbor nodes will have the largest impact on the usefulness of this node.*

REMEMBER - The QUALITY number determines how far away a node will be visible in the network and the order of automatic route selection.

The routes table contains information that will help when changing the default settings. The quality assigned to neighbors listed in the routes table can be changed to achieve the desired node set up. There are two very general philosophies that will help determine how to "finalize" and maintain the node.

Static Node

The default values for the K-Net node will cause every neighbor node heard, to be included in both the routes table and the nodes table (QUALITY = 255/70 and

MINQUAL = 70). Initially, this may be a bit of a problem since weak neighbor nodes and their distant nodes may be saved in the nodes table. Try to connect to all the neighbor nodes that are listed in the routes table.

Be sure to set NETROUTE such that all neighbor nodes heard direct will be included. Make note of which nodes respond with a good signal and modify the routes table with the ADDROUTE command to set and lock the quality of these nodes to a number higher than either value automatically assigned by the QUALITY command for either port. For example, if the QUALITY command is set to 255/70, and all good routes are locked at a quality of 115, setting MINQUAL to 115 results in a static node that is only aware of the neighbor nodes that are "locked in". This node will *never* have any new neighbor nodes *or* distant nodes automatically added to the nodes table. If distant nodes are to be listed in the nodes table, they must be manually added to the nodes table using the ADDNODE command, listing one of the existing neighbors as the route. Since distant nodes may have multiple paths from this node, the preferred route and alternate route(s) to a distant node can be specified with multiple ADDNODE commands. When connections to these distant nodes are requested, this node will *always* use the locked route path(s) in spite of changing band conditions.

Part of the responsibility of being a node sysop is maintenance. It's probably a good idea to set MINQUAL equal to or less than QUALITY from time to time to see if any new neighbor nodes appear in the routes table. If a good path to a new neighbor shows up, lock the new node route with a "good" neighbor quality and then reset MINQUAL to its previous value.

Dynamic Node

The default values for QUALITY and MINQUAL will allow *all* neighbor nodes that are heard to be listed in the nodes table. This means that both good and marginal neighbors will be listed in the nodes table. Of course, the nodes table will be larger than with the static node settings, but the benefit obtained is that the node will know when new neighbors appear and will automatically include them in the nodes table. Although distant nodes that may have good routes from this node will not be listed, some sysops feel that this is one method to control the propagation of marginal nodes. Another effect of dynamic node settings is that users can take advantage of band openings that could allow the use of nodes not normally visible on the network. This can lead to some frantic DXing when band conditions permit. The dynamic approach allows the nodes table to be elastic. It will grow as previously unknown neighbors appear and shrink as the obsolescence counters for each route decrement.

Once the K-Net node has been in operation for a while, "good" and "not-so-good" neighbor nodes can be recognized, a combination of static and dynamic settings can be used. By locking in "good" neighbors at high quality values, routing logic to take a preferred route in spite of changing band conditions can be "locked in". If MINQUAL is equal to or less than QUALITY, new neighbor nodes and nodes heard by virtue of a band opening will also come and go as the obsolescence counter is either reset or decremented. In all probability, neighbor nodes will have "good" nodes that this node should know about, and a neighbor's neighbor node will also have "good" nodes, as well as its neighbor's neighbor, and so on. Once "good" neighbor nodes have been established, the quality to assign to each neighbor will have to be determined experimentally in order to control the listing of marginal destination nodes that will appear as a result of receiving neighbor node broadcasts.

As a practical point, having a distant node is 3 or 4 RF hops away from this node listed in the nodes table in this node, is not good practice unless its path is extremely reliable. The chances of a user being able to access a node this distant diminishes greatly as the number of RF hops increases beyond this point.

Also, allowing destination nodes that are more than 3 or 4 hops to be listed can result in *very* large nodes tables being generated. When this node receives a nodes broadcast from a neighbor node, it will automatically calculate a quality to all the nodes contained in that broadcast. If the computed quality is less than MINQUAL, that distant node is not added to the nodes table. Therefore, it is desirable to have a combination of route quality and MINQUAL that will only allow nodes that are 3 or 4 node hops away to appear in the nodes table.

As an example, if a "good" neighbor node (A) has a "good" neighbor node (B), and node (B) has a "good" neighbor node (C), this node has a good chance of reliably reaching node (C). The computed quality at this node for distant node (C) must be greater than MINQUAL for it to be listed. If a quality of 70 is assigned to all neighbors in this example, this node would assign a quality of 5 for distant node (C), computed as follows (it helps to draw this out on paper): The quality from (B) to (C) is 70, when (B) broadcasts (C)'s quality as 70, (A) multiplies 70 by 70 and divides by 256 to arrive at a quality of 19 for (C).

When neighbor (A) does a node broadcast, its broadcasted quality of 19 for (C) is multiplied by this nodes path quality for (A) and divided by 256, resulting in a quality of 5 for distant node (C) (19 times 70 divided by 256). Since 5 is probably below MINQUAL, distant node (C) will not be listed, even though it is probably useable. What to do? The quality to neighbor node (A) could manually be changed to 255. This will result with this node assigning a quality of 18 for distant node (C), but it does not leave much room for a special case route that may be 5 or 6

hops away. This will also cause neighbor (A) to propagate further than desired in the network; creating useless routing. As a compromise, lock the quality of (A) to 140. If neighbor node (A) would also lock in a quality of 140 for his neighbor (B) and node (B) would do the same for node (C), then this node would now assign a quality of 41 for distant node (C), leaving some "wiggle room" for other nodes as they join the network. Now a MINQUAL setting of 41 will allow these "good" distant nodes to be listed in the nodes table of this node.

When working through the above exercise a few times using different quality values, it becomes apparent that there are many settings of quality that will produce the desired effect as long as all sysops adhere to a common guideline. However, the quality values used in the above example were not just "picked out of the air": 140 is a *reasonable* quality to assign to a good 1200-baud neighbor, since this allows good distant nodes to appear in the nodes table while also limiting node propagation. It also leaves room in the quality range for ultra-reliable routes.

This is where good neighbor relationships with other sysops in the area become important. Neighbor node quality must be continually examined to arrive at a usable network configuration that makes good use of all possible paths. Coordination of efforts and philosophies should result in an expanding network that is low in frustration and fun to use.

Node Stacking

Note: node stacking capability with other TNCs, is not available when using TNCs with the USB computer interface connection, such as this version of the KPC-3XE.

It can, however, be used with a node program in a computer, to add a node to the system.

Node stacking mainly applies to multiple TNCs (in network node mode) connected together through their RS232 interfaces.

The K-Net node function provides a quick and easy method to increase network usage and efficiency. As the network grows there will be an increasing demand to provide an even higher level of service to the community. This can take several forms: Adding more user ports to the network as existing user ports become heavily loaded, adding dedicated server access ports to "unload" activity from existing user/server shared channels, and the creation of point-to-point dedicated backbone channels to increase overall network performance.

Since knowledge of how to properly configure the K-Net node is assumed, this section will deal only with the physical connections and parameters that are required to create a node stack.

Before deciding to "help" the community, be aware of the impact that adding more ports of any nature will have on the existing network. Generally speaking, more available radio ports will provide a more efficient network provided that this effort is well coordinated with the other node sysops in the area that will be affected by additional network nodes. Sysop coordination and cooperation must exist for the network to provide the maximum benefit for all users.

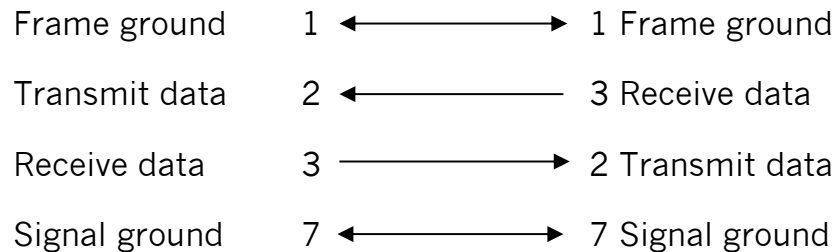
The concept of a node stack is practically self-explanatory. A node stack is formed when two or three or more network nodes are "hard-wired" together (stacked) and allowed to communicate with each other. Since the radio ports are already in use, the serial port of the TNC is used to pass the networking information and data between the nodes in the stack. Setting the TNCs command `INTFACE` (from the `cmd:` prompt) to `NET` forces the K-Net node to use the serial port as well as the radio ports for passing network information. Because of this, the serial port can no longer be used as an ASCII terminal port since all communication is now "net-ese" and used by the other nodes in the stack for routing and other network functions. If there is a need to change a node parameter, the terminal mode can be regained by connecting a terminal to the serial port and using the standard packet transparency escape sequence (three control C method), or the node can be remotely accessed via a radio port if the `RTEXT` has been set, and the proper response to the `SYSOP` command is sent to the K-Net node.

Again, the `QUALITY` parameter plays a major role when node stacking is used. The default `QUALITY` settings are 255/70 (255 for port 0 - serial port, 70 for port 1 - 1200 baud). The `QUALITY` setting for the serial port (port 0) is defaulted to the maximum value allowable because the serial link between nodes in a stack is very reliable. This value can be changed, but doing so may cause undesirable node propagation and routing.

The most basic form of a node stack would be to add another node TNC to an existing node TNC. The hardware required is two TNCs with RS232 computer interface ports, transceiver, feedline, antenna, for each TNC, and radio interface cable. The network node TNCs can be operating as a K-Net, NET/ROM or TheNET type nodes. The TNCs must be set to communicate in "NET" mode on their computer interface ports.

The serial interface cable between those two TNCs, needs to be a simple "NULL" type cable.

Constructing such a cable is straightforward. Using two male DSUB-25 connectors, between those two connectors, connect pin 1 to pin 1, pin 2 to pin 3, pin 3 to pin 2, and pin 7 to pin 7.



With the above cable assembled, the two network node TNCs are free to exchange network node data through their serial ports.

Each TNC must be given a unique NETALIAS and NETCALL because they are separate independent nodes. They should also be configured with the node parameters that are recommended in the local area coordinating with other area node operators. The serial cable functions as a "virtual radio circuit" for node-to-node communication in the stack.

REMEMBER! The serial ports are talking to each other - their serial port interface baud parameter must be set the same in both TNCs (i.e. 9600), and both TNCs must be set to communicate in NET mode.

Three Ports (One High-Speed and two 1200-Baud Ports)

A three-port node stack consisting of a high-speed port and two 1200-baud ports can be created by using a one-port 1200 baud TNC, with a dual-port TNC (such as a KPC-9612 Plus, or 9612XE).

Since the high-speed port of the 9612 is capable of 4800, 9600, or 19200 baud operation, this set-up offers tremendous flexibility in network configuration. As above, each of the K-Net nodes must be programmed with a different NETALIAS and NET CALL and use the serial port cable described above to tie them together. Now set the INTFACE command in each TNC to NET and power them off and back on to complete the installation.

More than Three Ports (three or more TNCs)

The two and three port stacks discussed above are relatively simple to cable together since there is no contention for the serial port. Each node is free to broadcast and pass data over the serial port "radio". Adding another node to the stack means that now there needs to be a method to control access to the

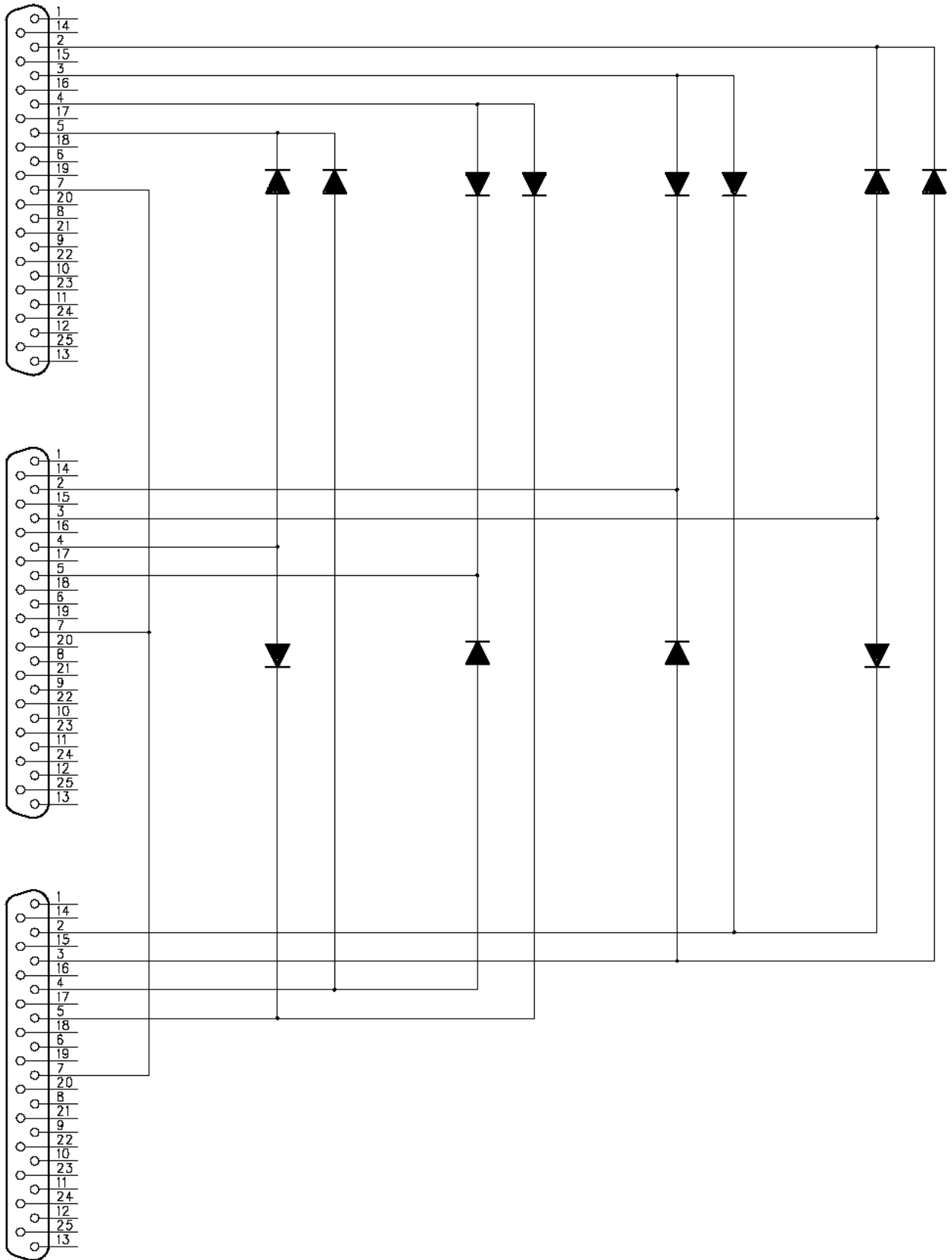
common serial port cable to avoid data collisions. The K-Net node uses the RTS (Request To Send - pin 4) and CTS (Clear To Send - pin 5) control lines of the serial port for this purpose.

When a K-Net node has data to send to the other nodes in the stack, it first checks the RTS line for a true (logic high) state. If RTS is true, then the node drops CTS and the data is sent to the other nodes in the stack. The dropping of CTS by any node in the stack will inhibit all the other nodes from using the serial port by forcing RTS false (logic low), thus avoiding data collisions within the node stack. The additional hardware required for this operation is commonly referred to as a Diode-Matrix.

K-Net Diode Matrix

A three, four, five, or six serial port Diode Matrix follows the same logic as above, with six ports probably being about the limit of practicality. The number of diodes required for a Diode Matrix with N serial ports is $2*N*(N-1)$.

- **Note:** This only applies when using TNCs that have RS-232 type serial ports.



Cross-reference Guide (TheNET X1-J / K-Net)

This guide is included to assist in the coordination of node parameters between an X1-J node and a K-Net node. The *basic* X1-J PARMS can be listed by connecting to an X1-J node and giving the P command. This should help in coordinating different types of network nodes in the same system. It can lead to meaningful discussions among node sysops to arrive at a "Network Standard" of node parameters that benefit all users.

X1-J PARM	K-Net COMMAND
1. Max. Destination Node Size	NETDEST
2. Min. Auto update quality	MINQUAL
3. Neighbor default quality	QUALITY
4. RS232 default quality	QUALITY
5. Initial obsolescence count	OBSINIT
6. Min. Obs. count to broadcast	OBSMIN
7. Node broadcast interval	NODESINT
8. Initial Time-to-Live	L3TTL
9. Transport Frack timeout (sec)	L4T1
10. Transport Retry counter	L4N2
11. Transport Ack Delay (sec)	L4DELAY
12. Transport Busy Delay (sec)	N/A
13. Transport Window Size (frames)	L4WINDOW
14. Transport Overfill Limit (frames)	N/A
15. No Activity Time Out	L4LIMIT
16. Persistence	cmd: PERSIST
17. Slot	cmd: SLOTTIME
18. Link Frack (T1)	cmd: FRACK

19.	AX.25 Maxframe	cmd: MAXFRAME
20.	AX.25 Retries	cmd: RETRY
21.	Link Response Time (T2)	N/A
22.	Active Check (T3)	cmd: CHECK
23.	AX.25 Digipeat	cmd: DIGIPEAT
24.	Validate Callsigns	N/A
25.	ID Beacon	cmd: IDINT
26.	CQ Broadcasts	CQBC
27.	ACL (Access Control List)	cmd: SUPCALLS/LLIST

To *completely* ignore uplinks from *and* downlinks to specific user callsigns and neighbor node *callsigns* (not alias)

cmd: BUDCALLS/CONLIST

To allow uplinks from and downlinks to *only* specific user callsigns and node *callsigns* (not alias)

Other ways of using this TNC are covered in this section.

Some of the following modes of operation require special software, so they may not be available if you are using standard terminal programs.

Remote Sensing and Control

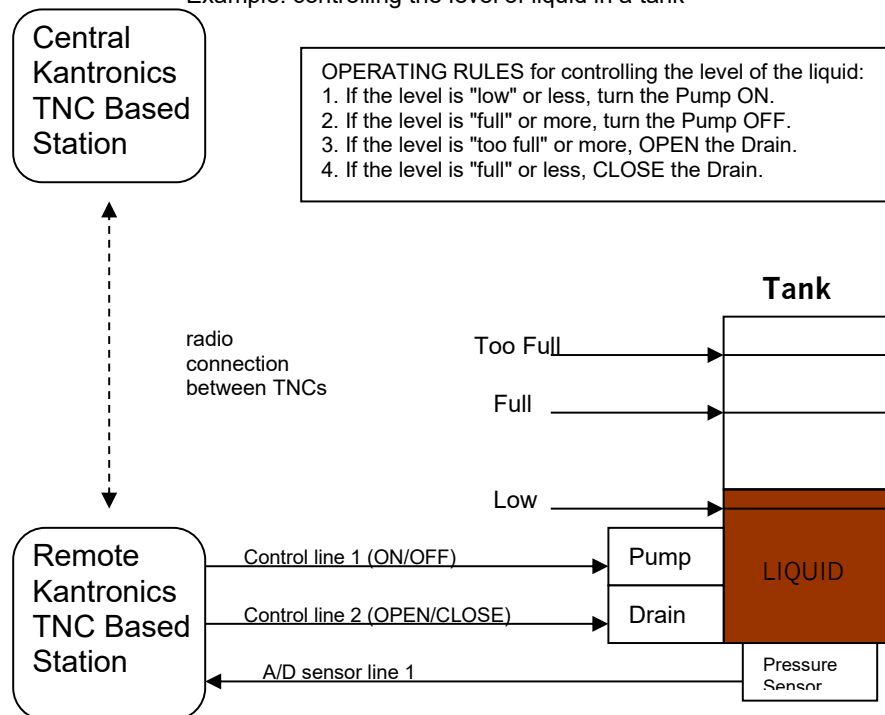
Two (or more) packet radio stations, each containing a Kantronics TNC, to implement remote sensing and/or remote control, as follows:

- **remote control of another TNC:** from a local Kantronics TNC-based radio station, use the MYREMOTE command to establish control of a remote Kantronics TNC-based radio station and issue commands for the remote TNC to carry out. For details, see the Command Reference.
- **sense analog inputs:** use the ANALOG command to gather and report the status of one or two analog input lines in the remote TNC. Kantronics single port and multi-port TNCs contain an 8-bit A/D converter and additional circuitry that converts the analog voltage presented at two specified inputs into digital values from 0 to 255. These digital values (reported in binary) as an approximation of the analog voltage values at the time the TNC interrogated the analog inputs. For details on the ANALOG command, see the Command Reference.
- **Note:** As documented in the ANALOG command and the radio port pin specifications, single port devices and multiport devices use different input lines and ports for analog sensing.
- **control output voltages:** use the CTRL command to control the voltages on selected output lines in the remote TNC (up to two output lines in single port devices and four output lines in multi-port devices). Each of these output lines can be set to OPEN (i.e., OFF) or GROUNDED (i.e., ON) by the CTRL command. Also, the CTRL command can be used with any or all of these output lines to send a specified number of short ON/OFF pulses (each pulse lasting about 100ms, with a fixed length pause between pulses) or a single long (about 1.5 s) ON/OFF pulse. For details on the CTRL command, see the Command Reference.
- The ANALOG and/or CTRL commands can also be used without the REMOTE command, to carry out sensing and control functions at a local Kantronics TNC.

The following diagram illustrates the use of two Kantronics TNC/radio stations for remote control and sensing. These operations could be carried out manually or via a terminal program running in the computer in the central TNC station.

Remote Sensing and Control

Example: controlling the level of liquid in a tank



As noted earlier, the Command Reference section and the ADVANCED INSTALLATION section provide details needed to adapt these broad capacities to a particular situation, with sensors, controls, at either local or remote locations.

OTHER MODES OF OPERATION

The following modes are typically used in industrial type systems. They would be used to link a control station to one or more remote site stations. The purpose would be for sending control signals or commands to a remote, or to gather data from remote sites.

LSUB (“wire” Line Substitution) Mode

Protocol Description

LSUB mode provides a method of transmitting and receiving raw ASCII data via a radio. Data is sent and received in a totally transparent fashion, allowing all ASCII characters to be utilized (hex 00 through hex FF). There is no addressing, error-detection or automatic retransmission of the data. Also, there is no flow control.

When the receiving modem detects the proper Start-of-Data sequence generated by the transmitting modem, it will output the data that follows to its computer interface port.

At the end of the received data there is an End-of-Data sequence that permits the receiving modem to recognize the end of the data and quit translating the receiver audio. This allows the modems to operate with open squelch audio from the radios.

Since the protocol constitutes a very small portion of the transmission, the effective data throughput is very nearly that of the radio baud rate (HBAUD).

LSUB Mode Operation

Commands Used

TXDELAY - The TXDELAY (TXD) command determines how long the modem must key the Push-to-Talk line of the radio before sending data. This time will have to be determined experimentally since it will vary from radio to radio. If the TXD is set too short, the data transmitted will be garbled and not received by other units in the system. Setting TXD too long will reduce channel capacity.

PACTIME - The PACTIME parameter determines how long the modem will wait after receiving data to be transmitted before beginning the radio key-up

sequence. Normally, this parameter should be set to AFTER 1 (100ms.) to minimize "dead-air" time on the system.

PERSIST and SLOTTIME - These parameters determine how aggressively the modem will attempt to access a multi-user channel based on carrier detect.

Entering LSUB Mode

Set the INTFACE command to LINESUB:

```
cmd:INTFACE LINESUB <cr>
```

Exiting LSUB Mode

Sending three control-Cs (unless a different control character has been defined). (See TRANS command in the Reference Manual).

Poll Mode

Poll Mode Protocol Description

The POLL mode is generally used in system applications that require and implement a poll and response protocol between a central site (Control Site) and one or more remote locations (Remote Site).

It is similar to the TUP mode since it uses a data transparent broadcast mode and a packet protocol with CRC-16 error-checking to ensure the integrity of the received data. All ASCII characters (hex 00 through FF) can be transmitted and received.

POLL mode differs from TUP mode by allowing an application program to address specific locations without changing any TNC parameters. All data sent by an application program at the central site must begin with the address (POLLID) of the intended remote modem. When any remote modem receives data from the radio, the data is checked for errors and the received POLLID is compared with its own POLLID. If they match, the POLLID is stripped and the data that follows is sent to the computer interface port. All data received by the central site from the remote location will arrive with the POLLID of the remote unit automatically attached to the beginning of the data.

It is the responsibility of the application program to: (1) address data to be sent to the remote sites, (2) determine if the data was received, and (3) recognize responses that contain the remote site's POLLID and data.

Since Poll mode is intended for "site-to-site" operations, there is no digipeating capability.

Poll Mode Operation

Commands Used

Each modem is set up differently depending upon whether it is to be a Control modem or a Remote modem.

Control Site

The POLLID command in the Control site modem must be set to 0000.

Control Site Transmit Format - All data sent to the Control modem via its computer interface port **must** begin with a POLLID (a four-digit remote address between 0001-9999) and be followed immediately by the poll data. The data sent by the application to the Control modem must contain less than the TNCs PACLEN (maximum characters in a transmitted data frame) setting.

If the poll data contains more characters than allowed by PACLEN, the data will be fragmented, and multiple packets will be formed and transmitted by the modem. These additional packets will contain **only** the balance of the data received by the modem. Since these packets do not begin with a POLLID, they will be ignored by all Remote modems and lost.

Control Site Receive Format - All Remote transmissions received by the Control modem will consist of the Remote POLLID of that sending unit, followed immediately by the Remote data.

Remote Site

Configuration - The POLLID command in each of the Remote modems must be set to a different number between 0001 and 9999. This ensures that data is received by just one remote device.

Remote Site Transmit Format - Remote sites generally do not have much computing capacity. Therefore, the transmitted format of the data is straightforward. All data input to that modems computer interface port will be

formed into packets and transmitted with the address (POLLID) attached to the beginning of each data packet.

Remote Site Receive Format - All data packets received by the Remote via the radio are first examined for errors using a CRC-16 error-checking algorithm. Data packets that contain errors are discarded. If the data contains no errors, it is then checked for a match of the received POLLID and its own POLLID. If there is not a match, the data is discarded. If there is a match, the POLLID is stripped and the raw data is passed to the computer interface port for processing. The received data format will be the same as the data sent to the Control Modem by the application program.

The TNCs CONMODE parameter affects when data will be transmitted in Poll Mode.

If CONMODE is set to CONVERS (default), data will be transmitted when any of the following conditions are met:

1. The SENDPAC character is received by the modem. The default SENDPAC character is a carriage return and can be changed to suit an existing application. The SENDPAC character can be either stripped or appended to the data by setting the CR command ON to append, or OFF to strip.
2. PACLEN is exceeded. **CAUTION:** *(Control Site) operation will be unusable if PACLEN is exceeded since each packet of data sent MUST begin with a (Remote Site) POLLID (0001-9999) address.* Remote transmission may exceed the setting of PACLEN since each packet transmitted from the Remote will contain the proper POLLID. If fragmentation of data occurs, the application program is responsible for reconstructing the data. PACLEN determines the *maximum* size of data packets (default 128 characters). Normally, the amount of data to be transmitted will be considerably less than this.
3. CPACTIME is on and PACTIME expires. If CPACTIME is ON, data will be periodically transmitted based on PACTIME timeout.

If CONMODE is set to TRANS, data will be transmitted when:

1. PACLEN is exceeded, or

2. PACTIME expires.

Suggested Settings

Usually, the TNC CONMODE command is set to TRANS and the PACTIME command is set to AFTER 1. The CONMODE TRANS command makes the modem data transparent to the Remote device. The PACTIME AFTER 1 command allows the modem to transmit the data received via the serial port with as little delay as possible. This speeds up the response time of the system by reducing "dead-air" time.

It is also recommended to set the PACLEN command to zero. This allows the modem to send a single packet of data to the radio of any size between 1 and the maximum of 256 characters. If the response data from the remote contains more characters than the PACLEN command allows, the data will be transmitted in multiple packets with the Remote POLLID automatically attached to the beginning of each data packet. In such a case, it is the responsibility of the Control Application Program to reassemble the data.

CAUTION: Allowing the Control Site application to send more data than that permitted by the PACLEN command is the most common cause of failure using the POLL mode. BE ABSOLUTELY SURE THAT CONTROL SITE TRANSMISSIONS NEVER EXCEED THE PACLEN LIMIT!

Entering POLL Mode

Set the INTFACE command to POLL:

```
cmd:INTFACE POLL <cr>
```

and power the unit off and then back on.

Exiting POLL Mode

To exit POLL mode to adjust TNC parameter settings (or for any other reason), the command mode of the modem can be regained as follows:

If CONMODE = CONV

```
Enter: 0000QUIT<cr>
```

(NOTE: No space between the 0000 and QUIT. QUIT must be in upper case)

If CONMODE = TRANS

Use the transparent data escape sequence, i.e., send a series of three control-Cs (unless a different control character has been defined). (See TRANS command in the Reference Manual).

TUP (Transparent Unproto Packet) Mode

Description

The TUP mode protocol utilizes the Unproto mode of the AX.25 protocol, allowing digipeaters (VIA path specified with UNPROTO command) to be used to effectively extend the range of a radio data system beyond simplex distance. This protocol uses the standard AX.25 CRC-16 error-checking to provide a high degree of data integrity. Data is sent and received in a totally transparent fashion, allowing all ASCII characters to be utilized (hex 00 through hex FF). I. Radio transmission is based on the PACTIME AFTER parameter setting of the modem, the setting of the carrier detect method, and the settings of the slot and persist channel access algorithm. Any data that is received by the modem via the radio is output to the serial port *only* if it contains no detectable errors. The received data will be *exactly* the same as the data that was sent to the input serial port of the transmitting mode, without any address headers or added formatting of the data. The modem is capable of selectively filtering monitored data from or to different modem addresses using the BUDLIST command.

If digipeaters must be used to extend the range of the system, the information frame sent by any modems in the system will be output to the computer interface ports of the receiving modems only once, regardless of how many times it was received correctly. Also, data sent by any modem will not be decoded by itself when the data is digipeated. The TUP mode protocol provides an automatic "Time to Live" timer that inhibits any modem from transmitting or receiving additional data until sufficient time has expired to allow in-progress data to travel through all the digipeaters specified in the UNPROTO field of the originating modem.

Since the time for a given packet to travel the entire UNPROTO path may vary due to the presence of other signals or noise, the command TUPTIME is used to impose an additional delay time, if needed, during which each modem will be inhibited from decoding or transmitting new data. This setting can be determined experimentally. It is important to note that TUPTIME applies *only to the receiving modems* and should be set to the same value at all sites. It is the responsibility of the host system to ensure that additional data is not transmitted until sufficient time has elapsed to allow data to travel to the last digipeater and the response, if

any, to travel back to the host. Note that the automatic "Time to Live" timer and the TUPTIME timer are only in effect if packets are being digipeated.

TUP Mode Commands

Basic TUP Operation involves just a few commands:

MYCALL - The MYCALL command determines the MODEM address. This must be entered after the autobaud routine has detected an asterisk (*).

TXDELAY - The TXDELAY (TXD) command determines how long the modem will assert the Push-to-Talk line of the radio before sending data. This time will have to be determined experimentally since it will vary from radio to radio. If the TXD is set too short, the data transmitted will not be received by other units in the system. Setting TXD too long will reduce channel capacity. NOTE: Since TXD is used in the computation of the automatic timer used to inhibit transmission until a digipeated packet has had time to reach its destination, TXD must be set to the same value in all modems in the system

PACTIME - The PACTIME parameter determines how long the modem will wait after receiving data to be transmitted from the DTE before beginning the radio key-up sequence. Normally, this parameter should be set to AFTER 1 (100ms) in order to minimize "dead air" time on the system.

PACLEN - PACLEN sets the maximum size of the data packet(s). Most applications do not require large amounts of data to be transferred so the default value (128) will probably be satisfactory, allowing packets up to 128 bytes. If the amount of data to be sent exceeds the PACLEN parameter setting, the transmission will be composed of multiple packets containing the data.

PERSIST and **SLOTTIME** - These parameters determine how aggressively the modem will attempt to access a multi-user channel based on carrier detect.

Entering TUP Mode

Set the TNCs INTERFACE command to TUP:

```
cmd:INTERFACE TUP <cr>
```

and power the unit off and then back on.

Exiting TUP Mode

To exit the TUP mode to "fine tune" parameter settings (or for any other reason), the Command mode of the modem can be regained by using the transparent data escape sequence, i.e., by sending a series of three <ctrl-c> characters (unless a different control character for COMMAND has been defined).

Modem Mode

In MODEM mode, the TNC becomes a 1200 baud "dumb modem" device. It uses the RTS signal line in the computer interface port, to toggle between receive and transmit.

When in transmit mode (RTS input has been set to ACTIVE state), ASCII text that is sent to the TNC is converted to audio tones that are sent through its radio port to a transmitter.

When in receive mode (RTS input has been set to in-active state), receiver audio is converted to a string of ASCII data that is output through its computer interface port, un-modified.

No "protocol" adjustments or decoding are made to the data.

The TNCs ABAUD should be set to 1200, before entering modem mode.

When using a terminal program to view the ASCII output from the TNC in modem mode, that terminal should also be set to 1200 baud interface.

Entering modem mode

Set the TNCs ABAUD command to 1200 (if not set already)

ABAUD 1200 <CR>

Reset the TNC and set the terminal interface baud also to 1200,

Set the TNCs INTface command to modem:

INTFACE MODEM <CR>.

Then power the TNC OFF and back ON.

Exiting modem mode

To leave modem mode, send three <ctrl+c> characters to the TNC at 1200 baud, or set the RTS line active and power the TNC off and back on.

- Note: In some cases, it may be necessary to set the TNC command CD to EXTERNAL. The CD line state in the computer output port of the TNC follows the state of the TNCs receive led. Some computers may lose some of the data stream, as the CD line cycles high and low.

Modemk Mode

In MODEMk mode, the TNC demodulates the receive audio signal from the radio and passes the resulting string of ASCII data through to its computer interface port, unmodified. No “protocol” adjustments are made to the data; the TNC simply acts as a 1200 baud ASCII signal de-modulator modem.

The TNCs ABAUD should be set to 1200, before entering modem mode.

When using a terminal program to view the ASCII output from the TNC in modem mode, that terminal should also be set to 1200 baud interface.

Entering modem mode

Set the TNCs INTface command to modem:

INTERFACE MODEM <CR>.

Then power TNC OFF and back ON.

Exiting modem mode

To leave modem mode, send three <ctrl+c> characters to the TNC at 1200 baud.

- Note: In some cases, it may be necessary to set the TNC command CD to EXTERNAL. The CD line state in the computer output port of the TNC follows the state of the TNCs receive led. Some computers may lose some of the data stream, as the CD line cycles high and low.

Receiving NWS EMWIN broadcasts

MODEMk mode can be used to copy emergency weather broadcasts by the National Weather Service. The NWS Emergency Managers Weather Information Network (EMWIN) is a wireless direct-to-computer information service that the National Weather Service (NWS) provides with help from the Federal Emergency Management Agency. The NWS feeds a constant stream of weather information (@ 1200,8,N,1), from around the world, to a satellite orbiting the Earth. In many areas, a local broadcast of that data may be available on a fixed radio channel (usually on an FM VHF frequency).

Using the TNC in MODEMk mode, with a generic terminal program, these plain ASCII text data streams can be viewed.

Information about EMWIN and software required to utilize its data, search the Internet.

Kantronics Host Mode Operation

Kantronics' Host Mode allows the TNC to communicate with computer software that is more sophisticated than standard terminal programs. Programs that utilize HOST interface mode may automatically set the TNCs INTERFACE command to HOST mode and perform a soft reset when started.

A host mode interface program will provide some means to give commands to the TNC. For a complete description of that host program, refer to the manual that comes with that software.

After switching into to HOST mode, that TNC will remain in HOST mode until it is returned to COMMAND mode. The easiest way to switch out of HOST mode is to use that host mode software to return the TNC to command mode on exit (see instructions with that software).

When using a Terminal program while the TNC is in HOST mode, a message "SØØ ", sent by the TNC indicates that it is in HOST Mode. When seeing this message, the Host Mode must be exited before using a regular terminal program.

The Host Mode escape sequence is simply 3 characters (a small Host data frame).

The first and last characters are FENDs (frame ends). The middle character is a single q.

If the terminal program allows higher numbered ASCII characters to be entered and sent, send a FEND character (ASCII code 192), then the letter q (upper or lower case is ok), and finally another FEND character.

When the TNC receives the host escape sequence it will then leave Host Mode, switch to COMMAND mode, and send its usual Kantronics sign-on message. When using a Host Mode program, it may send these characters automatically when the program exits, or it may have a function to perform that procedure to leave the TNC in its COMMAND mode when that program exits.

See the Appendix for details on this procedure.

KISS Mode

Amateur packet radio communications can use the TCP/IP suite of protocols instead of AX.25 by switching to KISS mode and running software that supports the KISS code designed by Phil Karn (see references at end of this section). The KISS protocol specifies the communication between a TNC and a host (e.g., computer).

KISS Mode allows the TNC to act as a modem and packet assembler/disassembler (PAD). In KISS Mode, data processing is shifted from the TNC to a computer running special software supporting the KISS code designed by Phil Karn. KISS code supports higher level protocols (i.e. TCP/IP) for sharing computer resources in a network fashion. Examples of TCP/IP programs are NET, NOS, and JNOS. Also, some networking software (G8BPQ) requires the KISS Mode of operation.

To manually place the TNC into KISS Mode, from its command prompt (cmd:), type `INTERFACE KISS` and press return. Then, either enter the `RESET` command, or power the TNC off and back on.

When in KISS mode, no message or any prompt will be returned from the TNC.

The TNC will remain in KISS mode when you turn it back on. To exit KISS mode, either:

- perform a hard reset, which will clear all previous settings changes, or
- use the TCP/IP command to exit KISS (available in KISS software while in KISS mode), or
- send manually send the KISS escape character sequence in hexadecimal characters (`C0 FF C0`) if allowed by your terminal program.

To send the `C0 FF C0` sequence from the keyboard (using a PC compatible computer):

1. Press and HOLD the ALT key. Type the numbers 192 from the numeric KEYPAD, not the keyboard. Release the ALT key.
2. Press and HOLD the ALT key. Type the numbers 255 from the numeric KEYPAD, not the keyboard. Release the ALT key.

3. Press and HOLD the ALT key. Type the numbers 192 from the numeric KEYPAD, not the keyboard. Release the ALT key.

If the terminal program sent those characters, the TNC will be out of the KISS Mode and back to terminal mode, now.

References on KISS Mode

Karn, P.: "TCP/IP: A Proposal for Amateur Packet Radio Levels 3 and 4", pp. 4.62-4.68 ARRL Amateur Radio Computer Networking Conferences 1-4, Newington, CT: ARRL, 1985.

Chepponis, M. and Karn, P.: "The KISS TNC: A Simple Host-to-TNC Communications Protocol", pp. 38-42, 6th Computer Networking Conference, Newington, CT: ARRL, 1987.

XKISS (Extended KISS) Mode

The TNC also supports the G8BPQ multi-drop KISS Mode, called extended KISS (XKISS). With this mode, the TNC can be connected to a G8BPQ node along with the other TNCs on the same serial port, to provide multiple radio ports to the node. Otherwise, XKISS works like KISS. For information on this mode of operation, see the documentation for the XKISS software.

DAMA (Slave Mode) Capacity

All of Kantronics' current amateur TNCs (KPC-3XE, 9612XE, and KAM XL) support the DAMA protocol (slave mode) as used by many amateur radio operators in Germany, Belgium, Luxembourg, the Netherlands, France, Switzerland and other parts of Europe.

Note: To the best of our knowledge there are no DAMA nodes in operation in the US or anywhere else outside of Europe.

What is DAMA?

DAMA (Demand Assigned Multiple Access) is a network protocol for regulating packet traffic to eliminate or dramatically reduce collisions among users on the crowded channels available to amateurs in densely populated northern Europe. As the European channels have become more and more crowded and more and more DAMA systems have been established, it has become a necessity for TNCs to be equipped with the DAMA slave protocol in order to use the local networks. In the DAMA system, a "master" node is set up in each local area to control transmissions from all stations connected to the node. By local agreement, users do not connect directly to each other; they connect only through the master.

DAMA was developed in Germany by NORD<>LINK. (References to more information on DAMA are provided in the bibliography at the end of this section.)

What Does a DAMA System Look Like?

The DAMA master node requires a computer running specialized DAMA software. TheNetNode (TNN) software, developed by NORD<>LINK is one such specialized program which is used in the computer of a DAMA master along with one TNC, with a special EPROM for each radio frequency used on the system. RMNC/FlexNet software is another program, which is very popular in Germany.

Within each Local Area Network (LAN), end users set themselves up as DAMA "slaves," using specially equipped TNCs, which contain firmware (EPROM) that can operate in a DAMA "slave" mode (such as the Kantronics TNCs with DAMA slave mode support). The slaves may use a standard terminal or host mode program or any of a variety of specially written European programs, such as SP, Grafik Packet and ESKAY, some of which make use of the KISS mode feature of the TNC.

To use the DAMA system, the user connects to the DAMA master with a normal connect frame. When the master responds to the connect frame, the TNC is automatically placed into DAMA slave mode. The DAMA master polls each slave

within the LAN in a “round-robin” fashion (somewhat like the token passing method) and the slave TNCs transmit packets when the master has granted permission to do so. The slave TNCs continue to be polled with priority among the slaves assigned according to the number of times each responds with information frames (as opposed to rr frames - i.e., no data to send). Priority is dynamically assigned so that those stations that are sending information are polled more frequently than idle stations. However, once an “idle” station responds with information, its priority is upgraded.

Communication over the wider area is handled between the DAMA master nodes which are connected to each other using a standard NET/ROM™ type protocol. For the DAMA system to work effectively, each slave associates with only one master node and connects only to that master. If an end user is connected to more than one master, *each* master will poll the slave TNC according to its polling schedule. When polled, the slave TNC will transmit *all* frames it has available to whichever master polls it. This would increase the probability of collisions with other users, thus defeating the purpose of the DAMA system.

How is DAMA Implemented in Kantronics' TNCs?

Kantronics TNCs, implement the DAMA slave mode, using the following commands (see the Command Reference section for details):

DAMA {ON|OFF}

Set DAMA to ON to have the TNC operate as a DAMA slave station once a connection is established with a DAMA master station. When DAMA is OFF, the TNC will operate in its carrier-sense-multiple-access mode (CSMA). For details, see the CD, SLOT, and PERSIST commands in the Command Reference section of this manual.

DAMACHCK n (n = 0 to 225)

Set a “DAMA time-out” timer (in 10-second increments) to specify how long the TNC (running in DAMA slave mode) will wait to be polled by a connected DAMA station before disconnecting from the DAMA station and reverting to CSMA operation.

Bibliography

Bertrem, Eric, F5PJE. “RMNC/FlexNet: The Network of Choice in Western Europe.” In “Packet Perspective” conducted by Stan Horzempa WA1LOU. QST April 1995: 88.

Jost, Gunter, DK7WJ, Sonnabend, Joachim, DG3FBL, and FlexNet-Group. "FlexNet. The European Solution". *Proceedings of the 9th Computer Networking Conference*. American Radio Relay League, Newington, CT, USA 06111.

Medcalf, Karl, WK5M. "DAMA - Another Network Solution." *Proceedings of the 14th Computer Networking Conference*. American Radio Relay League, Newington, CT, USA 06111.

Rech, Wolf-Henning, N1EOW/DF9IC and Kneip, Johannes, DG3RBU. "The German (Central European) Packet Radio Network: An Overview." *Proceedings of the 11th Computer Networking Conference*. American Radio Relay League, Newington, CT, USA 06111.

Schmid, Detlef J., DK4EG. "DAMA - A New Method of Handling Packets?" Trans. Mark Bitterlich, WA3JPY. *Proceedings of the 8th Computer Networking Conference*. American Radio Relay League, Newington, CT, USA 06111. Published in German as "DAMA, ein neues Verfahren für Packet Radio?" *cq-DL* April 1989.

TheNetNode (TNN) Copyright 1994, NORD><LINK e.V., c/o Klaus-Dieter Vieth, Goergesstrasse 3, D-38118 Braunschweig, Germany. NET/ROM is a trademark of Software 2000, Inc.

CAUTION! The DAMA firmware in Kantronics' TNC is based on documentation published in the ARRL Computer Networking Conference papers (see Bibliography) and documentation provided by NORD><LINK, the developers of DAMA. This firmware was tested with TheNetNode firmware (Version 1.56), TFKISS (Version 2.0) and by volunteer testers in Europe with firmware used in existing DAMA networks. Kantronics cannot be responsible for subsequent changes to the DAMA protocol which may render this firmware obsolete nor guarantee satisfactory operations with newer versions of the protocol and/or complementary software.

Command Reference

Introduction

This chapter documents all KPC-3XE commands.

There are many commands which affect operation of the KPC-3XE. Some commands affect performance under specific conditions, some change parameters affecting general operation and others direct a one-time action.

The user changes parameters and issues instructions to the KPC-3XE by typing commands composed of English-like word abbreviations and variables which are numbers or strings of characters chosen by the user. It is likely that some of these parameters will never need changes.

If any setting or value is changed, the new setting or value will be stored in the TNCs battery-backed RAM and will be that new value used at future power-on.

The availability of the commands listed here depends upon what INTFACE (i.e., INTERFACE) mode the KPC-3XE is currently in (see the INTFACE command in the command listing below and check the index for more information on INTFACE modes).

Format for Listing Commands

All of the TNCs commands are listed alphabetically below.

A “❖” bullet in front of the name of a command indicates that the command is a NEWUSER Command— it will be available for use when the TNC has had its INTFACE command set to NEWUSER Mode.

Format Defining Commands

The Command Line

The first line of each command definition shows the syntax for entering the command, beginning with the name of the command, followed by a listing of any parameters that may be used with the command.

The full name of each command is shown in CAPS and a “short-form” of the command is shown by the underlined characters in the name. The short-form of

the command is the shortest string of characters, starting from the beginning of the full name, which will allow the TNC will recognize to uniquely identify the command being requested.

The command name entered alone, will return the current setting of that command.

When changing a commands setting, the command name is followed by a space, then after the space, there may be one or more parameters that need to be used, or that may optionally be used, with that command.

- Values that need to be entered as they are shown (or by using the short-form indicated by the underlined character(s)) are shown in CAPS, as in “ON | OFF”. Parameter names for which a value needs to be substituted are shown in lower case letters, as in “callsign,” which means a particular callsign needs to be entered.
- Lists of parameter values are enclosed in curly brackets “{ }” for grouping clarity and lists of optional parameter values are enclosed in square brackets “[]”.
- If more than one choice is possible for the value of a given parameter (i.e., flag), choices are separated by a vertical bar, as in “ ON | OFF” or “OPTION1 | OPTION2 | OPTION3”.

If a parameter can take a range of values (or in cases such as ABAUD, any of a set of numerical values) the parameter is shown as a name in lower case (e.g., “n”) and the permissible range of values (or the list of possible values) is given in parentheses “()” to the right side of the command, after the syntax has been presented. For example, possible values of the parameter “n” are defined as “(n = 1 - 255)”.

The Line Below the Command Definition

If commands have parameters, default values are shown on the line below the command definition.

Some KPC-3XE commands take effect immediately, so they are called “immediate” commands (e.g., K means “switch to CONVERSE mode”). This will be stated on the second line. For example, the command to go to CONVERS mode says “immediate” on the second line.

More Information on Commands

Each command is explained in detail and examples are given where appropriate. Following the explanation of the command, there may be a “**See also**” list that shows related commands.

Parameter Types

Almost all parameters are of one of the following types (exceptions are discussed when they are encountered in the command listing).

n (range)

Any number within the range is permissible. The unit of measure (s, ms, baud, count, etc.) for the number will be given in the description. These are decimal numbers.

n (\$00 - \$FF)

Special functions: Some parameters are used to control special functions, such as specifying the character to use to enter the Command mode from the Convers mode or specifying the character to use for “backslash”. Possible values and default values for these parameters are shown in HEX format.

The “digits” of a hex number represent powers of 16, analogous to the powers of 10 represented by a decimal number. The decimal numbers 10 through 15 are represented in hex by the letters A through F (e.g., decimal 10 is hex A). And a hex number is distinguished from a decimal number by preceding it with a “\$” prefix.

For example:

$$\text{\$1B} = (1 \times 16) + 11 = 27$$

Number codes for these special characters are shown in hexadecimal (hex) form (i.e., base 16). They can be entered either in decimal or in hex. Permissible values are shown in HEX: for example (n = \$00 - \$FF).

Note: Setting a function character to zero (\$00) disables that function.

- See the ASCII Chart at end of this manual for a listing of ASCII codes for all alphanumeric characters, including A to Z and 0 to 9, and conversions between hex and decimal representations.

Several parameters are ASCII numerical values for characters, which perform special functions. Most of these special function characters have “control characters” as default values. Control characters are entered by holding down a special control key (normally marked “Ctrl”) on the keyboard while typing the indicated key. For example, to enter a <Ctrl+X>, hold down the Control key while typing an X, then release both keys.

These special characters cannot be sent in a packet unless preceded by the pass character (see the PASS command) or when operating in the Transparent Mode.

If a streamswitch (STREAMSW) character or any other special character is defined as “\$” then values will be entered in decimal numbers, or precede the \$ with the PASS character in order to enter hex numbers.

flags ChoiceA|ChoiceB

Many parameters are “flags”, meaning they have two possible values, ON and OFF, or YES and NO. All of the command descriptions show ON and OFF as the options; however YES (Y) and NO (N) may be typed instead. A few parameters are really flags, but rather than indicating that something is “on” or “off”, they select one of two ways of doing things. Some of these parameters have the values EVERY or AFTER indicating operating modes for data transmission.

The possible choices are separated by a vertical bar. Some of the flag parameters will allow many choices, such as ON|OFF|TO|FROM.

callsigns xxxxxx-n

Several commands require callsigns as parameters. While these parameters are normally Amateur callsigns, they may actually be any collection of numbers and/or letters up to six characters; they are used to identify stations sending and receiving packets. A callsign may additionally include an “extension” (SSID, Secondary Station Identifier), which is a decimal number from 0 to 15 used to distinguish two or more stations on the air with the same Amateur call (such as a base station and a repeater). The callsign and extension are entered and displayed as call-SSID, e.g. NØGRG-3. If the extension is not entered, it is set to -0, and extensions of -0 are not displayed.

text

There are some commands, which have a parameter text string. This string can be any combination of letters, numbers, punctuation, or spaces up to 128 characters.

In order to be used, all string parameters must contain at least one non-space character. Characters with special meanings can be included, such as carriage return, into the string by preceding them with the PASS character. The string ends when a (non-passed) carriage return is entered.

Entering Commands

To enter a command, the KPC-3XE needs to be in COMMAND Communication Mode (as opposed to CONVERS or TRANS communication mode).

The prompt for Command Mode is:

cmd:

- **Hint:** Once in Packet Convers Mode, a <Ctrl+C> (see COMMAND) needs to be entered to return to the Command Mode. In the Packet Transparent Mode or GPS Mode, a special sequence is needed to return to Command Mode (see CMDTIME).
- **Hint:** If packets have been printing on the screen, it may not be evident in which mode the TNC is operating. To see the cmd: prompt, type a <Ctrl+C>. If the prompt does not appear, the TNC may be in Transparent Mode. If the TNC is in Command Mode, typing a Return will generate the “cmd:” prompt. When it is in Transparent Mode, the “cmd:” prompt will not be displayed when the Return key is pressed. If this is the case, three <Ctrl+C>s in rapid succession will need to be sent to the TNC to return to the Command Mode and get the “cmd:” prompt.

When at the Command Mode prompt, enter a command for the TNC by typing the command name (in upper or lower case) and, if desired, any required and/or optional parameter values (argument settings or values).

- The command name and each parameter value must be separated from each other by at least one space.

To ENTER the command, press the carriage return “CR” (usually an ENTER key). As soon as the “CR” key is pressed, the TNC will process the command request and take whatever action is specified (or will display an “EH?” if it does not understand what was entered).

As noted above, a short-form of a command name—the characters underlined in the listing can be entered.

*The short form is not necessarily the shortest unique string in the name or the first match in alphabetical order—rather it is the first match that the KPC-3XE will find as it scans through the **non-alphabetical listing** of commands in its EPROM.*

- **Hint:** Examine the value of any parameter by typing the command name followed by a “CR”. A special command, DISPLAY, allows returns a list of all command names with their current.

Commands

All KPC-3XE commands are documented below, in alphabetical order. The underlined characters in a command's name show the short-cut version of the command. Required or optional parameters are shown after the command name, as are allowable ranges of values, where appropriate.

8BITCONV {ON | OFF}

default ON

- This command was added to the first TNCs built (1980s) to accommodate terminals that would not receive (or would be confused by) 8-bit characters, only 7-bit ASCII. It is retained just in case the terminal or computer being used requires only 7-bit data. Also, it might be that the computer is capable of handling 8-bit characters, but terminal program will reject (or be confused or "hung") by them. In these cases, set 8BITCONV OFF.

When ON, transmission of 8-bit data is allowed in the packet Convers Mode and Transparent Mode. If it is desired to obtain 8-bit data transmission but do not want all the features of Transparent Mode, set this command to ON. If OFF, the 8th data bit is stripped (set to 0) for transmission and all data received will have the 8th bit set to 0. This does not affect the KA-Node, digipeat or PBBS functions.

ABAUD n (n = 0,1200,2400,4800,9600,19200.38400)

default 0

The parameter n sets the baud used for input and output through the computer interface port of TNC and the computer. If 0 is used, the TNC will run an autobaud routine upon power-up. This routine looks for an asterisk (*) character from the attached computer to set the ABAUD parameter.

When a different interface baud is desired, or to perform autobaud the next time the unit is powered up, change the ABAUD parameter. Change the ABAUD setting the TNC first, then issue the RESET command. This will set the TNC to use the new interface baud. Then set the terminal program to the new baud. Otherwise, a hard reset will be required to erase the ABAUD setting and reinitialize the TNC to perform the autobaud routine. (See Hard Reset section.) Note also that a hard reset will erase ALL stored parameter changes in the TNC and return them to factory defaults.

See also: reset, restore

ANALOG

immediate

Entering the ANALOG command returns a string of 8 values, based on the current voltage readings from 8 A-to-D lines. Only the first two of these values are relevant to the user, reporting two external voltages.

At any given time, each of the 8 A-to-D lines will have a voltage value somewhere in the range of 0 - 5 V dc. Instead of reporting voltages, the ANALOG command returns a decimal number in the range of 0 - 255 for each line. Each decimal value corresponds to a voltage input.

The reason ANALOG returns 8 values is that it is attached to an 8-way mux input of a single A/D converter. In the KPC-3XE, only the first two values are relevant to the user (several other lines could be used for special purposes, with appropriate modifications of the board, but they are normally intended just for internal use within the KPC-3XE).

The response is in the following form:

ANO/AN1/AN2/AN3/AN4/AN5/AN6/AN7

- AN0, input (0), reports a decimal number in the range of (0-255), representing the current dc voltage of an external input read through jumper J20 from pin 4 on the Radio Port. When J20 is in connected position, AN0 and CTRLB both are connected to pin 4 on the radio port. When J20 is in open position, only CTRLB connects to pin 4.

AN1, input (1), reports a decimal number in the range of (0-255), representing the current dc voltage of external input read through J21 from pin 8 on the Radio Port. When J21 is in connected position, AN1 and CTRLA are both are connected to pin 8 on the radio port. When J21 is in open position, only CTRLA connects to pin 8.
- AN2, AN3, AN4 are spare inputs, and are not connected for external access. Pads are provided on the board for user connections, if desired.
- AN5, is used to detect supply voltage on the power input jack.
 - AN6, and AN7 are only for internal use.

- Data can be gathered from a remote site by connecting to MYREMOTE and then issuing the ANALOG command.

See also: range, telemetry, myremote

AUTOLF {ON | OFF}

default ON

When ON, a line feed is sent to the terminal after each carriage return. This parameter should be set on when overprinting occurs and the terminal being used does not automatically supply its own linefeed after a carriage return. This command affects only the data sent to the terminal, not data sent to the radio.

See also: cr, lfadd

AX25L2V2 {ON | OFF}

default ON

This command provides compatibility with all known packet units implementing AX.25 protocol. When ON, Level 2 Version 2 protocol is implemented and the TNC will automatically adapt to whichever version the connecting station is using. When OFF, Level 2 Version 1 is implemented. Set this command to OFF if to digipeat through other units which do not digipeat version 2 packets. Setting this command OFF may be of benefit when using several digipeaters (not nodes) to send packets, or when conditions are marginal between the two stations involved. (NOTE: Changing this setting after connecting to another station will have no effect on the current connection.)

The major difference in V1 and V2 protocol is the method used to handle retries. In the connected mode, if a packet is sent and not acknowledged, Version 1 will re-send the entire packet and then disconnect if the RETRY count is reached. Version 2 will first send a poll, the response to this poll will determine if the packet was received. It is possible that the ack was collided with and therefore the data packet does not need to be resent. If the ACK shows that the data packet was not received, it will be re-transmitted. Each time a poll is answered the TRIES count is reset to 0. If the RETRY count is reached, version 2 will attempt to re-connect unless RELINK is OFF. If the re-connect attempt is unsuccessful, then version 2 will issue a disconnect.

See also: relink, retry, tries

For more information, see the book *AX.25 Amateur Packet-Radio Link-Layer Protocol Version 2.0 October 1984*, which can be obtained from the ARRL.

AXDELAY n

(n = 0 - 255)

default 0

The value of “n” is the number of 10 millisecond intervals to wait, in addition to TXDELAY, after keying the transmitter before data is sent. This delay can be helpful when operating packet through a standard “voice” repeater, or when using an external linear amplifier which requires extra key-up time. Repeaters using slow mechanical relays, split-sites, or both require some amount of time to get RF on the air.

See also: axhang

AXHANG n

(n = 0 - 255)

default 0

Each increment specifies 10 millisecond intervals. This value may be used to improve channel utilization when audio repeaters with a hang time greater than 10 ms are used. If the repeater squelch tail is long, it is not necessary to wait for AXDELAY after keying the transmitter if the repeater is still transmitting. If the TNC has heard a packet within the AXHANG period, it will not add AXDELAY to the key-up time.

See also: axdelay

BEACON [EVERY | AFTER] n

(n = 0 - 255)

default Every 0

The value entered for “n” is the number of minutes (i.e., one-minute intervals) to use with the beacon command.

A value of 0 turns the beacon OFF. Setting a value greater than 0 activates the beacon under the conditions specified. If the optional keyword Every is used, a beacon packet will be sent every n minutes. If set to After, a beacon packet will be sent ONCE after the specified interval with no channel activity.

The beacon frame consists of the text specified by BTEXT in a packet addressed to "BEACON". Beacon messages will be digipeated via any addresses specified in the UNPROTO command.

See also: btext

❖ **BKONDEL {ON | OFF}**

default ON

When ON, the sequence backspace-space-backspace is sent to the attached terminal when the DELETE character is entered. When OFF, the backslash character "\" is sent to the terminal when the DELETE character is entered.

See also: delete, redisplay

BLT n {EVERY | AFTER} hh:mm:ss [START hh:mm:ss] [CLEAR] (n = 1 - 4)

default EVERY 00:00:00 (for each of the 4 LT strings)

This command (used with GPS) sets the interval between beacons for the associated LT (Location Text) string.

n identifies which LT is used (1-4). If EVERY is specified, the LT will be transmitted at the interval specified by the hh:mm:ss parameter. This is set in hours:minutes:seconds, and all characters must be specified. If the optional key word START is given followed by a time (in 24 hour format), the beacons will not be transmitted until the specified time. The beacon will then be transmitted according to the setting of the EVERY parameter. (NOTE: If the current time is past the start time, the beacon will start at the next scheduled interval based on the start time.)

If AFTER is specified, the beacon will be transmitted once after no activity is detected on the channel for hh:mm:ss time. Selecting AFTER does not allow the optional START parameter.

For example, setting BLT 3 EVERY 00:15:00 START 03:30:00 would cause the third LT string (LT 3) to be transmitted every 15 minutes, with the first transmission each day occurring at 03:30:00.

If CLEAR is specified, the LT buffer is cleared after transmission. This way no copies of a NMEA string are repeated.

BREAK {ON | OFF}

default OFF

If BREAK is ON, a break at the keyboard causes a return to Command Mode from Converse or Transparent Mode.

See also: COMMAND

BTEXT text

(0 - 128 characters)

default (blank)

BTEXT specifies the content of the data portion of the beacon packet. Any combination of characters and spaces may be used with a maximum length of 128.

Entering a single “%” will clear BTEXT.

See also: beacon

BUDLIST [ON|OFF] [NONE | {+|-}call | call1,call2,...]

where call={callsign | {<|>}callsign | callsign{>|<>}callsign}

default OFF NONE

BUDLIST is used to determine which received packets will be monitored. When OFF or NONE, BUDLIST will allow monitoring of all packets, even if it has callsigns in its list.

The maximum number of callsigns allowed in BUDLIST is 10. Up to 10 callsigns may be entered at once, using the format shown after the command name, and the new list of from 1 to 10 callsigns will replace whatever list was there before. The BUDLIST command can be used to add a single callsign (+callsign), so long as there is room for the new callsign on the list (if there is not, the TNC responds with “EH?” and returns to its command prompt). Similarly, a single callsign (-callsign) can be removed from the list. And to remove all items from the current list, enter BUDLIST NONE.

A callsign entered without an SSID will match any SSID of that callsign. To match a specific SSID only, enter the callsign with that SSID: for example, callsign-n, where n = 0-15.

More selective monitoring is enabled when a callsign or two callsigns is combined with a ">" and/or a "<" character. For example, entering ">callsign" will enable monitoring of packets TO that callsign (assuming BUDLIST is ON), entering "<callsign" will enable monitoring of packets from the callsign, entering "callsign1>callsign2" will enable monitoring of packets from callsign1 to callsign2, and entering "callsign1<>callsign2" will enable monitoring of packets from callsign1 to callsign2 and vice versa.

- Note that entering callsign1>callsign2 or callsign1<>callsign2 counts as two of the 10 maximum allowed callsigns.

CALIBRAT

immediate

The CALIBRATE command generates a signal which may be used as an aid in adjusting the audio drive level of the modem connected to the transceiver.

The use of this command is explained in the Calibration/Equalization Section.

See also: Calibration/Equalization section

CANLINE n

(n = \$00 - \$FF)

default \$18 <Ctrl+X>

This command is used to change the cancel-line input editing command character. When in Convers or Command Mode entering a <Ctrl+X> will cancel all character input from the keyboard back to the last un-PASSed carriage return (unless PACTIME has expired, CPACTIME is turned on, and when in Convers Mode).

See also: canpac, cpactime, pass

CANPAC n

(n = \$00 - \$FF)

default \$19 <Ctrl+Y>

This command is used to change the cancel-packet command character. When in the Convers Mode entering a <Ctrl+Y> will cancel all keyboard input back to the last unpassed SENDPAC character (unless PACTIME has expired and CPACTIME is turned on).

This character also functions as a cancel-output character in Command Mode. Typing the cancel-output character a second time re-enables normal output. For example, if the command DISPLAY was sent to the TNC, a <Ctrl+Y> will cancel the display and a second one re-enables the cmd: prompt after the next “CR”.

See also: canline, cpactime, sendpac

CD {INTERNAL | EXTERNAL | SOFTWARE}

default INTERNAL

The CD command selects which carrier detect method will be used.

When set to INTERNAL, the TNC will detect a signal present on the channel, using an energy type carrier detect, allowing shared voice and data on the same channel.

When set to EXTERNAL, the carrier detect is supplied by an external device, connected to the XCD pin on the radio port.

If set to SOFTWARE, the firmware inside the TNC will detect the presence of data to enable the carrier detection, allowing operation with un-squelched audio. Correct operation of SOFTWARE carrier detect is affected by proper equalization and the SWP parameter. If the RCV LED flickers, this is an indication that receive equalization may need to be adjusted. (Equalization is set with an internal jumper.)

When CD is set to INTERNAL or SOFTWARE, the external carrier detect pin on the radio connector (if wired) is used to hold off the TNC from keying the radio.

- Hint: For one use of this function, see the APRS and GPS section of this manual.

See also: swp

CHECK n

(n = 0 - 255)

default 0

This “timeout” command is used to prevent a “hang-up” in a connected mode when a link failure occurs as a result of conditions beyond control of the connected stations. The value entered for “n” is the number of 10 second

intervals to wait during “no activity” before making a periodic check (poll) to determine that a connected state still exists.

If n equals 0, this “timeout” function is disabled. While using Version 1 (AX25L2V2 OFF), a check timeout will initiate a disconnect.

See also: ax25l2v2, kntimer, relink, rnrtime

CMDTIME n

(n = 0 - 15)

default 1

This command sets the time allowed for entry of required characters to exit the Transparent Mode. In order to allow exit to Command Mode from Transparent Mode, while permitting any character to be sent as data, a guard time of CMDTIME seconds is set up.

Entering a value for “n” sets the value of CMDTIME to “n seconds”.

To exit Transparent mode, wait at least CMDTIME since the last data character was sent. Then enter the COMMAND character (e.g., <Ctrl+C>)) three times, with a wait of LESS THAN the value of CMDTIME between the first and second entry of the COMMAND character and also a wait of LESS THAN the value of CMDTIME between the second and third time of the entry of the COMMAND character.

After a final delay of CMDTIME the TNC will exit Transparent Mode and enter Command Mode. At this time, the cmd: prompt should be displayed. If CMDTIME is set to zero, the only exit from Transparent Mode is a modem break signal, assuming BREAK is set to ON.

For example (if CMDTIME is 1 second and COMMAND is <Ctrl+C>): wait one second, type a <Ctrl+C>, within one second type a second <Ctrl+C>, within one second type a third <Ctrl+C>, WAIT one second, cmd: prompt should appear. If the terminal program has the capability, a modem break can be sent to escape Transparent Mode.

See also: command, trans, break

CMSG {ON | OFF | DISC | PBBS}

default OFF

When OFF, the custom connect text stored in CTEXT will not be sent to the connecting station upon receiving a connect request. When ON, the custom CTEXT string will be sent. When CMSG is set to DISC, the custom CTEXT string will be sent to the connecting station, and then the TNC will disconnect from that station. When set to PBBS, the custom CTEXT string will be sent to the connecting station, and then the connection will automatically be transferred to the PBBS. If the PBBS is not available, the TNC will disconnect from the station.

See also: ctext, pbbs

COMMAND n

(n = \$00 - \$FF)

default \$03 <Ctrl+C>

This command is used to change the Command Mode entry character. When COMMAND is set to the default value, typing a <Ctrl+C> causes the TNC to return to Command Mode from packet Convers Mode. See CMDTIME for returning to Command Mode from Transparent Mode.

CONLIST [ON | OFF] [NONE | {+|-}callsign | callsign1,callsign2...]

default OFF

CONLIST is used to determine which stations (callsigns) may use this station for ANY purpose, including digipeating. When ON, the TNC will recognize only those packets received with a callsign that appears in the CONLIST's list of callsigns; and connections cannot be made to any station that is not in the CONLIST. When OFF, the TNC will process all packets.

The maximum number of callsigns allowed in CONLIST is 10. Up to 10 callsigns may be entered at once, using the format shown after the command name, and the new list of from 1 to 10 callsigns will replace whatever list was there before. The CONLIST command can be used to add a single callsign (+callsign), so long as there is room for the new callsign on the list (if there is not, the TNC responds with "EH?" and returns the command prompt). Similarly, a single callsign can be removed from the list (using -callsign). And to remove all items from the current list, enter CONLIST NONE.

A callsign entered without an SSID will match any SSID of that callsign. To match a specific SSID only, enter the callsign with that SSID: for example, callsign-n, where n = 0-15.

CONMODE {CONVERS | TRANS}

default CONVERS

This command controls the mode the TNC will be placed in AUTOMATICALLY after a connect if NOMODE is OFF. The connect may result either from a connect request received or a connect request originated by a CONNECT command. If the TNC is already in Convers or Transparent Mode when the connection is completed, the mode will not be changed. If part of a command line was entered when the connection is completed, the mode change will not take place until the next command is completed or cancel the line input.

See also: canline, connect, convers, nomode, trans

❖ CONNECT call1 [VIA call2,call3,...,call9]

immediate

call1 = callsign of station to be connected to.

call2,...,call9 = optional stations to be digipeated through. A maximum of 8 digipeater addresses (callsigns or aliases) can be specified. This is referred to as a path.

Each callsign may also have an optional Secondary Station Identifier (SSID) specified as -n, where n = 1 - 15. The digipeat callsigns are specified in the order in which they are to relay transmitted packets. The mode set by CONMODE will be entered upon successful connect, if NOMODE is OFF. If no response to the Connect request occurs after RETRY attempts, the command is aborted. A timeout message is printed on the display and the TNC remains in the Command Mode. The station being connected to (call1) may receive the connect request but is unable to accept connects, in which case a busy message will be printed to the screen and the TNC will stay in Command Mode. Connect requests may only be initiated in the Command Mode and the connect will be established on the current stream.

If a connect is in progress, or already established, the path may be changed by simply re-issuing the CONNECT command with the desired path. This must be done on the same stream as the original connect.

CAUTION, packets en-route between this station and the reconnected station may be lost.

If CONNECT is entered with no parameters, the status of the current stream is displayed.

See also: conmode, conok, dbldisc, maxusers, nomode, retry, ring, streamsw, xmitok

CONOK {ON | OFF}

default ON

When ON, connect requests from other TNCs will be automatically acknowledged and a <UA> packet will be sent. The standard connect message, with stream ID if appropriate, will be output to the terminal and the mode specified by CONMODE will be entered on the I/O stream if not connected to another station and NOMODE is OFF.

When OFF, connect requests from other TNCs will not be acknowledged and a <DM> packet will be sent to the requesting station. The message “connect request: (callsign of whoever is trying to connect)” will be output to the terminal if INTERFACE is TERMINAL or NEWUSER.

When CONOK is OFF, connection can still be made to the mailbox.

When operating with multiple connects allowed, the connection will take place on the next available stream. Connect requests in excess of the number allowed by the USERS command will receive a <DM> response and the “connect request: (call)” message will be output to the terminal if INTERFACE is TERMINAL or NEWUSER.

See also: conmode, connect, intface, maxusers, monitor, nomode, users

CONPERM {ON | OFF}

default OFF

Setting CONPERM to ON forces the connection on the current stream to become permanent and causes an attempt to reconnect when the KPC is restarted. CONPERMED connections are shown with /P in the status display, which is generated by the STAT command.

See also: status

❖ **CONVERS**

immediate

CONVERS has no options. It is an immediate command and will cause entry into Conversational Mode from Command Mode on the current I/O stream. Any current link connections are not affected.

- Hint: K is the same as CONVERS, for quicker entry.

See also: k, command!

CPACTIME {ON | OFF}

default OFF

When OFF and in the Convers Mode, packets are sent when the SENDPAC character is entered or when PACLEN is achieved. When ON and in the Convers Mode, packets are sent at periodic intervals determined by PACTIME. Characters are sent periodically as in Transparent Mode, but the local editing and echoing features of Convers Mode are enabled.

See also: convers, cr, paclen, pactime, sendpac, trans

CR {ON | OFF}

default ON

When ON the SENDPAC character (normally carriage return) is appended to all packets sent in Convers Mode except when PACLEN is exceeded. Setting CR ON and SENDPAC \$0D results in a natural conversation mode. Each line is sent when a "CR" is entered and arrives at its destination with the "CR" appended to the end of the line. To avoid overprinting, AUTOLF may need to be ON at the receiving end.

See also: autolf, lfadd, sendpac

CRSUP {ON | OFF}

default OFF

When ON, this command suppresses every other carriage return (when no data is between them) in the received data before sending the data to the terminal.

When CRSUP is OFF, all carriage return characters are left in the received data as it is sent to the terminal.

See also: autolf, lfsup

CSTAMP {ON | OFF}

default OFF

When ON, the daytime stamp is printed with all “*** CONNECTED TO” and “*** DISCONNECTED” messages on the terminal.

See also: connect, daytime, disconnect, mstamp

CTEXT text

(0 - 128 characters)

default (blank)

Enter any combination of characters and spaces up to maximum length of 128. Entering a single ‘%’ will clear CTEXT. This entry specifies the text of the first packet to be sent in response to an accepted connect request provided that the parameter CMSG is not OFF.

See also: cmsg, connect

CTRL [A | B] {n | ON | OFF | LONG | MUTE m}

(n=1-20)

immediate

This command activates or deactivates the specified output line (A or B) to the radio port as indicated. If A or B output is not specified, both outputs are affected. For example, to pulse the A output (CTLA on the schematic) 3 times, the command “CTRL A 3” would be entered. The number of times to pulse the output (n), or to turn the output ON, or OFF is an option. Each pulse on the output line lasts approximately 100 ms (0.1 s). Specifying LONG results in one pulse of the output line, lasting approximately 1.5 s. These output lines are open drain outputs and are pulled to ground when active. For more information, see the section on “Connecting to the Radio” in this manual.

With MUTE, the control line can be used to mute voice repeater outputs. When MUTE is specified, the output control line will be a filtered version of the RCV led, with about a 10*m ms attack time. If the RCV led is active for m consecutive samples at 10 ms intervals, the CTRL line will activate; if the RCV led is off at any sample, the CTRL line will open.

Do not allow more than 200 mA of current in this circuit.

CWID [EVERY | AFTER] n

(n = 0 - 255)

default EVERY 0

Each increment specifies 1-minute intervals. A value of 0 turns the ID OFF. Setting a value greater than 0 activates the ID under the conditions specified. If the optional keyword “Every” is used, an ID will be sent every n minutes. If set to “After”, an ID will be sent ONCE after the specified interval with no channel activity.

The string specified by the CWIDTEXT command will be sent in Morse Code, using AFSK tones.

Some countries require all stations to ID in Morse code periodically.

See also: mycall, cwidtext

CWIDTEXT text

(0-15 characters)

default DE mycall

This command sets the text to be transmitted during an automatic CWID. The text will be transmitted periodically as set by the CWID command.

See also: cwid

DAMA {ON | OFF}

default OFF

When ON the TNC can operate as a DAMA slave station once connection is established with a DAMA master station. When OFF the TNC operates in standard Packet mode (i.e., CSMA—Carrier Sense Multiple Access).

See also: damachck

DAMACHCK n

(n = 0 to 255)

default 18

This command sets a DAMA timeout timer in 10-second increments. If the TNC is connected to a DAMA master station (that is, operating in DAMA slave mode) and the master does not poll the TNC for data before the DAMACHCK timer expires, the TNC will revert to CSMA operation and disconnect from the DAMA master.

This command was added to allow the user to regain CSMA connect capability in the event the DAMA master fails to poll because of a node failure – lightning strike, power outage, etc.

See also: dama

DAYSTR text

default mm/dd/yyyy hh:mm:ss

The DAYSTR command is used to set the FORMAT of date/time display. DO NOT enter an actual date or time. Simply enter the form of the date/time desired, using the lower case letters m, d, h, y, and s as described below.

The format entered is used for all time stamps, including the PBBS, KA-Node, Mheard list, etc.

The lower-case characters m, d, y, h, and s have special meaning to this command and will be replaced with data from the software clock. The lower-case m will be replaced with the minutes the first time it appears after a lower-case h. If h, m, y, d, or s is specified as a single character, the corresponding date/time element will be displayed as a single digit if the value is less than 10. Entering two characters will force a two-digit display for values under 10. If the month is entered as three ms, it will be displayed as the first three characters of the month name (JAN). Other text may be entered such things as the local time zone.

- Remember ALL lower-case m, d, y, h, and s will be replaced by actual values.

Some examples of setting the DAYSTR command and the resulting time/date display are:

DAYSTR setting	Displayed Date/Time
mm/dd/yyyy hh:mm:ss	01/16/2026 12:14:22
d.m.y h:mm:ss	16.1.02 12:14:22
d.mm.yy h:mm	16.01.02 12:14

mmm d 20yy h:mm CST
TIME hh:mm DATE: mmm dd, 20yy

Jan 7 2026 12:14 CST
TIME 12:14 DATE Jan 07, 2026

- This command is used to define how date and time are to be displayed; it is not used to enter an actual date and time. If an actual date and time is entered with the DAYSTR command, the displayed date/time will appear to stop.

See also: daytime

❖ **DAYTIME yymmddhhmm[ss]**

DAYTIME is used to enter a date and time (after which time and date are formatted for display by the specifications given in DAYSTR).

When date and time are entered, using the input format specified in the command and explained below, the values entered are used to set the software clock/calendar. The software clock/calendar is then used for MHEARD and NDHEARD logging, as well as for CSTAMP and MSTAMP functions.

When entering the daytime digits, enter them in pure number sequence with no spaces, dashes, or slashes. Notice that there are two digits each for entering a year, month, day, hour, minute, and (optionally) second. Also, notice the need to use the order given here, which has nothing to do with the format (given in DATSTR) for displaying date and time values. For example, to enter 2026, January 16, at 22:30:00 hours, the value of the DAYTIME parameter would be 260102223000. In this case, seconds were entered, so the string is 12 characters long (2 characters for each of six pieces of information entered).

If DAYTIME is entered with no parameter, the daytime is displayed using the format defined by the DAYSTR command.

- Hint: If the time appears to be stuck, check the current value of DAYSTR, to be sure that the DAYSTR setting is being used to format the display of time and date, and not (in error) to attempt to supply an actual date and time. If this is not the case, fix the problem and see if time and date displays work correctly.

See also: cstamp, daytweak, daystr, mheard, mstamp

DAYTWEAK n

(n = 0 - 15)

default 8

This parameter is used to adjust the software clock, if needed, for accurate time keeping. Increasing the parameter will slow the clock, decreasing the parameter will speed up the clock. Each count corresponds to 0.64 seconds increase or decrease per 24 hours. Ambient temperature will affect the clock to some degree.

DAYTWEAK has no effect on the (optional) battery-backed clock (BBC); that clock is set only by using the DAYTIME command. The software clock stops when a soft reset or powering off occurs, and upon power on the software clock is set to the value of the BBC, if installed.

DBLDISC {ON | OFF}

default OFF

When OFF, only one disconnect command (D) needs to be given to terminate an unsuccessful connect attempt. If connected, the normal disconnect sequence will occur.

When ON, a normal disconnect sequence will always occur. A second D command is required to force a local disconnect independent of the retry counter.

See also: disconnect

DELETE n

(n = \$00 - \$FF)

default \$08 <Ctrl+H>

This command sets the character to be used as the delete character. When this character is typed, the last input character is deleted. The most common settings are \$08 (backspace) and \$7F (delete).

See also: bkondel

DIGIPEAT {ON | OFF | UIONLY}

default ON

When ON, any packet received that has MYCALL, MYALIAS, or MYNODE (if KA-Node is active) in the digipeat list of its address field will be retransmitted. Each station included in the digipeat list relays the packet in the order specified in the address field. Digipeating takes place concurrently with other TNC operations and

does not interfere with normal connected operation of the station. To disable digipeat operations (via MYCALL, MYALIAS, or MYNODE) turn this command OFF.

If UIONLY is selected, only UI type frames will be digipeated.

See also: hid, myalias, mycall, mynode

❖ **DISCONNE [MYPBBS | MYNODE x]**

(x=KA-Node circuit)

immediate

When given without options, “disconnect” will initiate an immediate disconnect request on the current I/O stream. A successful disconnect results in the display of “*** DISCONNECTED”. If the RETRY count is exceeded while waiting for the connected station to acknowledge, the TNC moves to the disconnected state on that stream. Entering a second Disconnect command before RETRY has expired will result in an immediate disconnect but may leave the other station thinking it is still connected. Disconnect messages are not displayed when the TNC is in Transparent Mode. Other commands may be entered while the disconnect is in progress.

DISCONNE MYPBBS

Issue this command if to cause the personal mailbox to issue a disconnect to the user of the mailbox. D MYPBBS is what should be entered, do not type the call entered in the mypbbs command.

DISCONNE MYNODE x

(x = KA-Node circuit)

x may be any of the KA-Node circuits in use, designated by A, B, C, etc. This command will cause the node to disconnect the stations linked through the node on the circuit specified. MYNODE does not refer to the call entered in the mynode command but is the actual characters to type.

See also: dbldisc, newmode, retry, status

❖ **DISPLAY [{ A | C | G | I | L | M | N | P | T }]**

immediate

This command causes the TNC to display a list of all the parameters in the TNC. A display of only a selected group of parameters can be requested by specifying the appropriate class identifier for that group. When using the DISPLAY command

with a subclass be sure to use a space between the DISPLAY command and the subclass. Subclasses of related parameters are:

ASync	asynchronous port parameters (TNC to computer)
CHAR	special TNC characters
GPS	parameters related to GPS operations
ID	ID parameters
LINK	parameters affecting packet link (TNC to TNC)
MONITOR	monitor parameters
NET	parameters affecting K-Net
PBBS	mailbox commands
TIMING	timing parameters

Individual parameter values can be displayed by entering DISPLAY and the command name, followed by "CR".

See also: Display Listings section!

❖ **DWAIT n**

(n = 0 - 255)

default 0

DWAIT defines a delay to be used to avoid collisions with digipeated packets. The value entered for "n" is the number of 10 ms intervals, after last hearing data on the channel, for the TNC XE to wait to begin its own key-up sequence.

This value should be established and agreed on by all members of the local area network. The best value will be determined by experimentation but will be a function of the key-up time (TXDELAY). This feature is made available to help alleviate the drastic reduction of throughput, which occurs on a channel when digipeated packets suffer collisions. Digipeated packets are not retried by the digipeater but must be restarted by the originating station. If all stations specify DWAIT, and the right value is chosen, the digipeater will capture the frequency every time it has data to send since digipeated packets are sent without this delay.

Observations have proven that a better algorithm for avoiding collisions between end-user stations, while still allowing digipeaters the high-priority access they require is achieved using Persistence and Slottime to determine proper transmit intervals, and setting DWAIT to 0.

See also: persist, slottime.

❖ **ECHO {ON | OFF}**

default ON

When ON, characters received from the computer by the TNC are echoed back and displayed. If double print of characters entered at the keyboard is being echoed back to the terminal, set this command to OFF. This corresponds to the setting in the terminal program for duplex (half or full). If the terminal program is set for full-duplex set ECHO ON. If it is set for half-duplex (some call it echo) then set ECHO in the TNC to OFF. Regardless of the setting of this command, the TNC will not echo an X-OFF or X-ON character to the terminal when it receives a STOP or START character. Echo is disabled in Transparent Mode.

See also: bkondel, flow!

ESCAPE {ON | OFF}

default OFF

This command specifies the character, which will be sent to the terminal when an escape character (\$1B) is received in a packet. When OFF, \$1B is sent. This is useful if the terminal interprets ESC characters as screen positioning commands (ANSI). When ON, the escape character is sent as a dollar sign (\$).

FILTER {ON | OFF}

default OFF

When ON, this command will inhibit the printing of control characters (hex \$00 - \$1F) which may be present in monitored packets. This will be useful when monitoring channel traffic, which includes binary file transfers, or higher-level protocols (networks talking to each other). Control characters, which may be embedded in those packets, can have strange and unpredictable effects on the monitoring TNC. All control characters except carriage return (\$0D) and line feed (\$0A) will be filtered. This command DOES NOT affect receipt of control characters in packets received from a “connected” station when MONITOR or MCON is OFF.

See also: monitor

FLOW {ON | OFF}

default ON

When FLOW is ON, any character entered from the terminal will halt output to the terminal until the current packet or command is completed (by SENDPAC, PACLEN, or PACTIME). Canceling the current input to the TNC or typing the REDISPLAY-line character will also cause output to resume. FLOW will keep the display of received data from interfering with data entry.

When FLOW is OFF, received data will be “interleaved” with keyboard entry. If using a split screen terminal program, FLOW and ECHO should be set OFF to allow received data to be displayed while entering characters into the TNC’s type-ahead buffer.

See also: canline, canpac, cpactime, echo, paclen, redisplay, sendpac

FRACK n

(n = 1 - 15)

default 4

Each increment specifies 1-second intervals. After transmitting a packet requiring acknowledgment, the TNC waits FRACK seconds before incrementing the retry counter and sending the packet again. If the retry count (specified by the RETRY command) is exceeded, the current operation is aborted. If the packet address includes digipeaters, the time between retries is adjusted to $FRACK * ((2 * m) + 1)$ where m is the number of digipeater stations specified. When the retried packet is sent, a random wait time is also added to avoid lockups where two units repeatedly collide with each other.

The FRACK timer begins when PTT is released (the packet has been sent) and is suspended when data carrier from the radio is present, or when this station is transmitting.

See also: connect, resptime, retry

FX25RX {ON | OFF}

default OFF

This command enables the error checking of received packets with FX.25 encapsulation.

FX25TX (0 | 16 | 32 | 64)

default 0

This command enables the FX.25 forward error correction to transmitted AX.25 packets, when set to non-0 (zero). The number of “check bytes” selected, offers correction to lost or corrupted data inside the packet. 16 check bytes can correct up to 8 damaged bytes, 32 can correct up to 16, and 64 can correct up 32.

GPSHEAD n string (n = 1-4)

(string up to 8 chars)

default 1 \$GPGGA (string 2, 3, and 4 = <blank>)

This command determines which GPS NMEA sentences will be stored in the LT buffers. n (1-4) determines which buffer will be used to store the data, and string is a NMEA sentence header.

When the header is received from the GPS unit, the KPC-3XE will store up to 128 characters in the associated LT buffer.

- To clear one of the four strings, enter LT and the buffer number n (n = 1-4), and nothing else, then press the ENTER key.

See also: blt, lt, ltp, ltrack

GPSINIT string

(string up to 128 characters)

default (blank)

Note: This command will have no effect in the KPC with the USB computer interface. A GPS device cannot be connected the USB version TNC.

This command establishes a string which will be sent to the attached GPS unit upon power-up (i.e., initial text sent to terminal in GPS Mode). This may be useful to configure a GPS unit to provide only required output data. Some GPS units may require more than one sentence; consult the GPS unit manual. To send more than one, enter a <Ctrl+N> at the end of each sentence This will cause a CR/LF sequence to be sent to the GPS unit. A single % will clear the string.

See also: gpshead, intface

GPSPORT [baud] [NORMAL | INVERTED] [CHECKSUM | NOCHECK]

default 0 NORMAL CHECKSUM

This command allows input from a GPS unit, using the XCD input on the DB9 radio port.

If 'baud' is non-zero, the XCD pin in the radio port can be used as a GPS input during regular operation. 'baud' may be set to 0, 300, 600, 1200, 2400, or 4800. If NORMAL is specified, the sense (polarity) of the XCD pin is RS232; the XCD pin would be expecting RS232 input or 0-5 V input of the same polarity. If INVERTED is specified, the XCD pin would be expecting polarity opposite of RS232. Most stand-alone GPS units use RS232 or 0-5 V output of the same polarity. If CHECKSUM is specified, the firmware will pass along only GPS sentences, which have valid checksums, and also GPS sentences, which do not have a checksum. Not all GPS units will append checksums to the GPS sentences that they send; some GPS sentences require a checksum, while others do not.

When using GPSPORT, note that the TNC cannot send any initialization string to the GPS unit; it must pre-initialized before connecting it to the XCD pin.

Note: The GPSPORT is a firmware UART. For best results, set 'baud' as low as possible for the application; when setting 'baud' to 4800, the CHECKSUM parameter option should be used, to discard errors.

Note: When GPSPORT is zero (0), the XCD pin reverts to its original function, inhibiting PTT. If INTERFACE is set to KISS, XKISS, or MODEM, GPSPORT should be set to zero.

GPSTIME { OFF | VALID | ON } { GGA | GLL | RMC | ZDA }

default VALID RMC

If GPSTIME is set to OFF, the TNC software clock is not set when NMEA strings are received. If set to VALID, the clock is set but only if the valid bit in the string specified is set. If set to ON, the time is acquired from the string specified.

Note:

- In any event, any string must have a valid checksum.
- Time is not updated if the time received in a string is the same as the last time received.
- GGA and GLL strings do not include a date.
- The ZDA string does not include a "valid" bit.

- Some GPS units do not include a “valid” bit with the GLL string.

HBAUD n

(n = 300, 400, 600, 1200)

default 1200

This baud specifies the radio port baud rate of data exchange with other TNCs. The value of HBAUD has NO relationship to the computer port baud rate specified with ABAUD. In order to communicate with other packet stations, the baud must be the same at each end of the link. As a general rule, 300 baud is used on frequencies below 28 MHz, 1200 baud is used on frequencies above 28 MHz. FCC rules currently limit the maximum baud to 300 when operating below 28 MHz. Although the KPC-3XE can operate 300 baud, the modem tones will still use a 1000 Hz shift whereas most HF packets use a 200 Hz shift.

HEADERLN {ON | OFF}

default ON

When ON a carriage return is output to the terminal between the header and text of monitored packets. This causes the packet header and time stamp (if on) to be displayed on one line, with the packet text displayed below it on the next line. When receiving packets only from local packet connects (MONITOR and/or MCON OFF) this parameter does not apply. When OFF the data will be on the same line as the header.

See also: cstamp, mcon, monitor, mstamp

❖ HELP [command]

immediate

Entering the word “HELP” alone, without any argument, will generate a display listing all commands available.

When the name of a command is entered also (e.g., HELP CONVERS), a brief description of that command will be displayed.

A wild card also can be used; entering “Help C*” will display brief descriptions of all commands beginning with C.

HELP can also be called by entering “?”.

HID {ON | OFF}

default ON

When ON, an ID packet will be sent every 9.5 minutes, provided that packets are being digipeated through this station, routed through the KA-Node, or into the PBBS. This command should be ON if digipeating, KA-Node or pbbs is enabled. If OFF, periodic identification packets will not be sent.

See also: digipeat, id, myalias, mynode, mypbbs, numnodes, pbbs

HTEXT text

default blank

The HTEXT command is used by the PBBS to provide hierarchical routing information. Use this command to enter the hierarchical portion of this stations packet address. Do not enter the TNCs MYCALL callsign or the first period.

A hierarchical address consists of state, country, and continent codes, separated by periods. Some areas (in large states particularly) may also use sub-state designators. For instance, a station in Rhode Island might use a hierarchical address of RI.USA.NOAM. This means Rhode Island (RI) which is in the United States (USA) which is in North America (NOAM). In this case, HTEXT would be set to RI.USA.NOAM

When unsure of the local hierarchical address, contact a local packet BBS sysop and ask what the proper addressing should be. Entering HTEXT %, will clear any text previously entered.

Note: The PBBS will not be able to forward or reverse forward messages unless HTEXT is set.

ID

immediate

When this command is entered an identification packet will be transmitted. This command may be used to ensure that a station identification is the last transmission before taking the station off the air.

The ID packet is an unnumbered information <UI> packet whose data consists of this station's identification as set in its MYCALL and a listing of other modes, which are active.

The MYCALL will be appended with "/R" if DIGIPEAT is ON, MYALIAS callsign appended with "/D", MYNODE callsign appended with "/N", and MYPBBS callsign appended with "/B". This packet will be addressed to "ID" and digipeated via any addresses specified in the UNPROTO command.

See also: hid, unproto

❖ **INTERFACE**

{TERMINAL|NEWUSER|BBBS|GPS|HOST|KISS|XKISS|MODEM|MODEMK|NET|POLL|TUP}

default TERMINAL

When set to TERMINAL, the full command set of the TNC is available.

When set to NEWUSER, the TNC will operate with a limited command set available to the user.

When set to BBS, the TNC deletes certain messages (i.e. *** connect request, *** FRMR, etc.) for greater compatibility with full-service BBS programs such as WØRLI, WA7MBL, CBBS, etc.

When INTERFACE is set to KISS mode of operation, users interact with the TNC using software that implements the KISS code specified by Phil Karn (for more information, see the KISS Operation section of this manual).

When INTERFACE is set to XKISS mode of operation, users interact with the G8BPQ extended KISS protocol (for more information, see the KISS Operation section of this manual).

When INTERFACE is set to HOST, the TNC will talk to the attached computer using the Kantronics HOST mode and users communicate with the TNC using "host mode" software (for more information, see the HOST Operation section of this manual).

When INTERFACE is set to GPS, the TNC will enter GPS mode upon power up. Data from the serial port will be parsed according to the GPSHEAD command, text will be placed into the LT buffers, and beacons will be transmitted according to the setting of the BLT commands. Note that this setting will not apply to the TNCs with a USB computer connection.

When INTERFACE is set to MODEM, the KPC-3XE mirrors at the serial port @ 1200 baud what it receives at the radio port (for more information, see the “modem mode” in the Modes of Operation chapter).

To have the TNC exit GPS Mode, connect a PC or terminal to it and issue three <Ctrl+C> characters (see COMMAND for details).

For instructions on exiting KISS, XKISS, HOST and MODEM modes, see the appropriate sections of the “other modes of operation” chapter of this manual.

NOTE: After changing the setting of this command to KISS, XKISS, HOST, GPS, or MODEM a soft reset is required for the new mode to take effect (see RESET for details).

K

immediate

This single letter command is synonymous with CONVERS. It is included as a single-keystroke convenience for entering Convers Mode.

See also: convers

KNTIMER n

(n = 0 - 255)

default 15

If there is no activity (data) on a KA-Node circuit for n minutes, the KA-Node will disconnect both the input and output sides of the KA-Node circuit. Setting KNTIMER to 0 disables this feature.

See also: rnrtime

LCOK {ON | OFF}

default ON

When ON, no character translation occurs in the TNC. If OFF, lower case characters will be translated to upper case before being output to the terminal from the TNC. This case translation is disabled in Packet Transparent Mode.

LEDS {ON | OFF}

default ON

To conserve power, the front panel LEDs can be disabled by setting LEDS to OFF.

LFADD {ON | OFF}

default OFF

When ON, a linefeed will be appended to every carriage return received from the keyboard before being transmitted. When OFF, no line feeds will be added to the transmitted signal.

LFSUP {ON | OFF}

default OFF

When ON, this command suppresses any line-feed characters received from the other station, relying on the terminal program to properly advance to the next line. When OFF, the line-feed characters received from the other station are not suppressed but are sent to the terminal as received.

See also: autolf, crsup

LGETCHAR \$xx

Default \$05 (CTRL-E)

LGETCHAR specifies a hot key functional in command or converse mode. This key causes the unformatted strings from the LTEXT buffers to be output to the computer interface port. This key performs the same function as the command "LT RESTART". Set it to 0 to disable it.

LLIST [ON | OFF] [NONE | {+|-}callsign | callsign1,callsign2...]

default OFF

LLIST is used to determine which stations (callsigns) may NOT use this station for ANY purpose, including digipeating. When LLIST is ON, the TNC will NOT recognize those packets received with any callsign that appears in the LLIST's

(lid) list of callsigns. In addition, when LLIST is ON, connections cannot be made to any station that is on that list.

The maximum number of callsigns allowed in LLIST is 10. Up to 10 callsigns may be entered at once, using the format shown after the command name, and the new list of from 1 to 10 callsigns will replace whatever list was there before. The LLIST command can be used to add a single callsign (+callsign), so long as there is room for the new callsign on the list (if there is not, the TNC responds with "EH?" and returns to the command prompt). Similarly, a single callsign (-callsign) can be removed from the list. And to remove all items from the current list, enter LLIST NONE.

A callsign entered without an SSID will match any SSID of that callsign. To match a specific SSID only, enter the callsign with that SSID: for example, callsign-n, where n = 0-15.

LT {n [text] | RESTART}

(n = 1-4) (text up to 128 chars)

default blank (for n = 1 - 4)

This command, used with GPS activity, fills the specified LOCATION Text (LT) buffer with "text". n specifies which buffer to use (1-4). If GPSHEAD is set for this buffer, and if the INTERFACE command is set to GPS, the buffer contents will be updated automatically by the GPS data.

To clear one of the 4 LT strings, enter "LT n%", where "n" is the number of the LT string clear.

When the command "LT RESTART" is executed, the LT buffers are output in an unformatted manner to the computer interface port, and the BLT timers are restarted. "Unformatted" means that only non-empty LT strings are output and that there are no preceding numbers indicating which strings are being output.

See also: blt, gpshead, ltp, ltrack

LTP n dest [via call1[,call2,...]]

(n =1-4)

default GPS (for n = 1-4)

This command sets the destination callsign and the digipeaters used to transmit the LT strings. n specifies which string (1-4). The destination callsign defaults to GPS; however, to change it to APRS, LOCATE, POSIT, or some other destination. Up to 8 digipeater callsigns may be specified.

See also: lt

LTRACK n [LT1] [LT2] [LT3] [LT4] [SYSOP] [TIME]

default 0

This command allocates memory for a special tracking buffer used to store LT messages. n defines the amount of memory (Kbytes) set aside for the LT messages (limited by available RAM). Specifying the LT1 through LT4 parameters determines which messages will be stored in the tracking buffer. If SYSOP is specified, only a user with SYSOP privileges may access the tracking buffer, and if TIME is specified, a time stamp will be included, in front of the LT message.

Specified LTs are put in the LTRACK buffer based on the BLT settings.

The LTRACK buffer is accessed by connecting to the mailbox (MYPBBS) and using the mailbox's List Tracking buffer Command (LTR). To see the syntax of this command, type HELP at the BBS prompt.

To change LTRACK, enter the entire command name, LTRACK, not just the short form (LTR).

See also: ltp, blt, GPS operation section

MALL {ON | OFF}

default ON

When ON, monitored packets include connected and unconnected packets between other stations. If OFF, only other station's unconnected packets (UI frames) will not be displayed.

This is a useful arrangement when stations are talking as a group in an unconnected configuration.

See also: monitor

MAXFRAME n

(n = 1 - 7)

default 4

MAXFRAME sets an upper limit on the number of unacknowledged information packets, which can be outstanding at any one time. The TNC will send MAXFRAME number of packets in a single transmission, if they are available.

See also: paclen

MAXUSERS n

(n = 1 - 26)

default 10

This command causes the TNC to allocate the memory required for the maximum number of simultaneous connections to be allowed. Each connection uses a different stream. In order to direct data to a different stream, use the STREAMSW character. All streams may be used for outgoing packets, but USERS sets the number that may be used for incoming connections.

Changing the value of MAXUSERS will cause the TNC to perform a “soft reset”. Note that the value of MAXUSERS cannot be changed while in connected state.

Note: Although the entire command word (MAXUSERS) is needed to change the current value of MAXUSERS, it can be entered in the short form, MAXU, to just see the current value of MAXUSERS.

See also: status, streamsw, users

MBEACON {ON | OFF}

default ON

This command determines whether packets addressed to Beacon or ID will be displayed on the screen.

To not display Beacon or ID packets heard, turn this command OFF.

See also: beacon, id, monitor

MCOM {ON | OFF}

default OFF

Supervisory (control) packets are not monitored unless MCOM is set ON and MON is ON. In addition, if this station is in connected state, control packets are not monitored unless MCON is also ON. Control packets, when displayed, are

distinguished from information packets by the “<>” and “<<>>” character sets appended to the packet header. With these characters, a letter is included to denote the type of control packet received, e.g. <<c>> - a connect request packet. The types of control packets are listed below.

<C>	Connect request
<D>	Disconnect request
<DM>	Disconnected Mode
<UA>	Unnumbered Acknowledge

In addition, the following bracketed information will be added to the Information packets as appropriate:

<UI>	Unconnected Information frame
<Is>	Information frame (connected); s = send sequence number

See also: ax25l2v2, monitor, mresp

For details on control packets, see the book *AX.25 Amateur Packet-Radio Link-Layer Protocol Version 2.0 October 1984*, available from the ARRL.

MCON {ON | OFF}

default OFF

When OFF, and connected, the TNC will display only those packets from a local connection. Any header information displayed will be determined by the settings of STREAMEV and STREAMCA. When OFF, and not connected, all eligible packets (as determined by other monitor commands) will be displayed. When ON, all eligible packets will be displayed, whether connected or unconnected.

See also: monitor, streamca, streamev

❖ MHEARD [SHORT | LONG | CLEAR]

immediate

This command causes display of a list of stations heard. An asterisk, *, indicates that the station was heard through a digipeater. The date/time the station was last heard is also displayed.

If the S option is used, i.e. MHEARD S, then only the callsigns of the stations heard will be displayed.

If the L option is selected, all callsigns contained in the received packet including the digipeater paths, are displayed. For example:

```
NØGRG > ID 10/16/88 14:31:30
```

```
VIA TOP,KCI,WØXI
```

Here, this station heard NØGRG transmitting an ID packet. NØGRG was also using the digipeating path TOP, KCI, WØXI. If this station heard NØGRG via one of these other stations, an asterisk would show by the call or alias of the last digipeater heard. When listed short, the asterisk would show beside NØGRG.

- Whenever PASSALL is ON, packets will not be checked for errors and no additions will be made to the MHEARD list.

If the C option is used, the list is cleared.

See also: daytime, passall

MHEADER {ON | OFF}

default ON

When ON, the headers are displayed for all monitored packets. When OFF, headers are not displayed and only data is output to the terminal. Since only I and UI frames have data, only these frames are displayed.

- To avoid confusion when MHEADER is OFF, use BUDLIST to restrict the monitored data.

See also: monitor, mcon, mbeacon, mall, suplist, budlist

❖ MONITOR {ON | OFF}

default ON

When ON, unconnected packets will be monitored unless prohibited by SUPLIST, BUDLIST, CONLIST, or LLIST. This will also allow monitoring of other packets if permitted by the other monitor commands. The MONITOR command acts as a master switch for the MALL, MCOM, MCON, MRESP, and MRPT commands. The

addresses in the packet are displayed along with the data portion of the packet. Callsigns (to and from fields) are separated by a ">"; and the Secondary Station Identifier (SSID) is displayed if it is other than 0. If any data is contained in the monitored packet which does not follow the AX.25 protocol, it is displayed in curly braces on the header line. All monitor functions are disabled in the Transparent Mode.

When OFF, only data from those stations connected locally will be displayed, no matter how other monitor commands are set. Any header information displayed will be determined by the settings of STREAMCA and STREAMEV.

See also: budlist, conlist, headerln, llist, mall, mbeacon, mcom, mcon, mresp, mrpt, mstamp, pid, streamca, streamev, suplist

MRESP {ON | OFF}

default OFF

When ON, monitored packets include the following AX.25 response packets, if MONITOR and MCOM are ON. If connected, MCON must also be on. The bracketed information will appear at the end of the header information. The "<" and "<<" characters are used to bracket and denote packets received as version 1 or version 2, respectively. For example, "<<RR1>>" denotes a version 2 packet. In addition, upper case characters are used to designate commands (polls) and lower-case characters are used to denote responses for RR, REJ, and RNR. For example, "<<rr1>>" is a response in version 2.

<FRMR>	Frame Reject
<REJr>	Reject, r = received sequence number
<RNRr>	Device busy, r = received sequence number
<RRr>	Receive Ready, r = received sequence number

In addition, the following bracketed information will be added to the Information packets as appropriate:

<lsr>	Information frame (connected); s = send sequence number, r = received sequence number
-------	---

See also: ax25l2v2, mcom, monitor

For more information, see the book *AX.25 Amateur Packet-Radio Link-Layer Protocol Version 2.0 October 1984*, which may be obtained from the ARRL.

MRPT {ON | OFF}

default ON

This command affects the way monitored packets are displayed. If ON, the entire digipeat list is displayed for monitored packets, and the station that relayed the packet is indicated with an asterisk. The MONITOR command must be ON for this command to work, and if connected MCON must also be ON. If OFF, only the originating station and the destination station callsigns are displayed for monitored packets.

See also: monitor

MSTAMP {ON | OFF}

default OFF

This command enables time stamping of monitored packets. The date and time information is then available for use for automatic logging of packet activity or other computer applications. The date and time are set initially by the DAYTIME command, and the date format is determined by the DAYSTR command. The MONITOR command must be ON for this command to work, and if connected MCON must also be ON.

See also: cstamp, daytime, monitor

MXMIT {ON | OFF}

default ON

When ON, transmitted packets will be displayed as monitored data on the terminal. Repeated packets will be displayed as they are sent over the radio. The frames to be displayed will be controlled by the MONITOR, MCOM, MCON, and MRESP commands, and will obey the settings of TRACE, MSTAMP, HEADERLN, 8BITCONV and FILTER commands.

See also: monitor

MYALIAS xxxxxx-n**(n = 0-15)**

default (blank)

Setting this command to a callsign or character string enables digipeating by using the MYALIAS. Enter up to six characters (plus optional SSID) which are different than those used for MYCALL, MYNODE, MYPBBS, or MYREMOTE. For example: LAW-3 may be entered as the MYALIAS, which would be easier to remember for stations wishing to digipeat through a station in Lawrence, enroute to a station more distant. Digipeating by an alias name in this TNC can be disabled with the command MYALIAS %.

See also: hid

❖ MYCALL xxxxxx-n**(n = 0-15)**

This command tells the TNC what its callsign is.

The extension n is called a Secondary Station Identifier (SSID) and is defaulted as 0 and is hidden but may be any number from 0 to 15. All packets originated by the TNC will contain this callsign in the FROM address field. Any packets received by the TNC with this callsign in the TO address field or digipeat fields will be responded to appropriately (connect, disconnect, ack, digipeat, etc.).

- When the TNC is first turned on out of the box, or after a hard reset, it asks for a callsign - there is no default. The callsign entered for MYCALL is automatically used to compute other IDs too (MYPBBS, MYNODE, and CWIDTEXT), adding an SSID for each. Changing MYCALL at the "cmd:" prompt does not affect MYPBBS, MYNODE, or CWIDTEXT

See also: cwidtext, digipeat, id, myalias, mynode, mypbbs, myremote

MYDGPS xxxxxx-n**(n = 0-15)**

default (blank)

This command accommodates differential GPS operation. In GPS mode only, if a UI packet is received addressed to the call in MYDGPS, the data from the UI packet is output on the computer interface port.

MYDROP n**(n = 0-15)**

default 0

This command sets the KISS address of the radio port. A KISS frame with the upper nibble of the command byte set to this value will address this TNC.

See also: intface, KISS operation section

MYNODE xxxxxx-n**(n = 0-15)**

default mycall-7

Setting this command to a callsign or character string enables the KA-Node in this TNC. Enter up to six characters (plus optional SSID) which are different than those used for MYCALL, MYALIAS, MYPBBS, or MYREMOTE.

The Ka-Node callsign will be included when it is enabled (when NUMNODES is set to a non-zero value.)

The KA-Node is disabled when MYNODE is set to the same as MYCALL, or when NUMNODES is set to 0.

See also: digipeat, ndwild, numnodes; KA-Node section

❖ MYPBBS xxxxxx-n**(n = 0-15)**

default mycall-1

This command is used to set the callsign (address) of this TNCs Personal Packet Mailbox. Entering a callsign with or without an SSID, -n, defines the address. The six characters and optional SSID must be different than those entered for MYCALL, MYALIAS, MYNODE, or MYREMOTE.

See also: pbbs

MYREMOTE xxxxxx-n**(n = 0-15)**

default (blank)

This command sets the callsign used for remote access to this TNC. Enter up to six characters (plus optional SSID) which are different than those used for MYCALL, MYALIAS, MYNODE, or MYPBBS. In addition, the RTEXT must be

programmed with a text, number, or combination string. A station that connects will be sent a series of numbers that must be decoded according to RTEXT.

See also: rtext; remote access section

NDHEARD [SHORT | LONG | CLEAR]

immediate

This command allows the operator to display a list of nodes whose ID packets have been heard by the TNC. The list includes KA-Nodes as well as K-Net, TheNet, NET/ROM and G8BPQ nodes.

K-Net, TheNet, NET/ROM and G8BPQ nodes are identified as:

ALIAS (CALLSIGN)

and Kantronics' KA-Nodes will be identified as:

MYNODE (MYCALL)

An asterisk "*" indicates that the station was heard through a digipeater. The date/time the station was last heard is also displayed. If the S option is used, i.e. NDHEARD S, then only the callsigns of the stations heard will be displayed. If the L option is selected, all callsigns contained in the received packet are displayed (see MHEARD for display format). If the C option is used, the list of nodes is cleared.

See also: mynode

NDWILD {ON | OFF}

default OFF

When OFF, the KA-Node will only recognize connect requests directed to the MYNODE call. When ON, connect requests to any SSID of the MYNODE call will be recognized as connects to the KA-Node, if that SSID is not being used for any other ID in the TNC.

See also: myalias, mycall, mynode, mypbbs, myremote

NETALIAS xxxxxx**(up to 6 alpha-numeric characters)**

default {blank}

The NETALIAS is an optional mnemonic that usually provides information about the location of the node. It exists only for the convenience of users. Network nodes automatically convert the NETALIAS to the associated NETCALL when setting up a circuit for a user. Changing or entering NETALIAS will not cause a soft reset.

NETBUFFS n**(n = 1 - 255)**

Default 32

The NETBUFFS command sets the number of node buffers available in the node. If the NETBUFFS reach zero, node operation may become unstable. Changing NETBUFFS will cause a soft reset as memory is reallocated. Each NETBUFF allocated uses 320 bytes of RAM.

NETCALL xxxxxx[-n]**(up to 6 alpha-numeric characters plus optional SSID)**

default {blank}

The NETCALL is the callsign of the K-Net node and can have an optional Secondary Station Identifier (SSID) extension of 1 to 15. The NETCALL must be entered before the K-Net can be operational and must be different from any other callsign used in this TNC.

Entering a % as the NETCALL (NETCALL %) will clear both the nodes and routes tables, but not change any node-related parameters. Entering or changing the NETCALL will cause a soft reset as memory is reallocated.

NETCIRCS n**(n = 1 - 64)**

default 5

This command limits the number of Network Circuits available in the node. A *circuit* is a connection between two nodes. The *circuit* will contain an *uplink* (originated by a user or server), a *crosslink* (the other network node), and usually a *downlink* (to another user or server). The *downlink* will not exist until requested by the originating *uplink* user. The *circuit* will exist until either the destination user or server or the originating user or server disconnects. The *circuit* will also be

discontinued if one of the nodes determines that data can no longer be sent due to a retry time-out (L4N2), or if the inactivity timer (L4TI) expires. Changing the NETCIRCS will cause a soft reset as memory is reallocated. Each NETCIRC allocated uses 50 bytes of RAM.

NETDESTS n

(n = 1 - 255)

default 25

Setting the NETDESTS determines the maximum number of destination nodes that can be stored in the K-Net node list. This list is sent to users when the NODES command is given to the K-Net node and when the node does a nodes broadcast. Setting NETDESTS to a very large number (i.e. greater than 50 or so) is not a good idea since this can create long node broadcasts and long transmission times to users in response to a NODES command. Setting NETDESTS to a very low value can result with a desired node getting "bumped" out of the nodes table. After the K-Net node has been in operation for a while, it should be able to determine how many of the destination nodes that are listed in its nodes table are *reliable* paths. The value entered for NETDESTS should be greater than this number, allowing room for other nodes as they become available. In any case, never allow the node destination table to become so large that it is filled with unusable and unreliable destination nodes. Very large nodes tables (>50 or so) are simply baffling to new users, and most of the time are unusable to even experienced "node hoppers".

Although setting NETDESTS will place a "hard limit" on the number of destination nodes that can appear in the nodes table, proper setting of the K-Net QUALITY and MINQUAL commands will allow the number of destination nodes listed in its nodes table to "float" with changing band conditions and assure that reliable destination nodes are still listed. Changing NETDESTS will cause a soft reset as memory is reallocated. Each NETDEST allocated requires 31 bytes of RAM.

NETLINKS n

(n = 1 - 64)

default 10

The NETLINKS command sets the maximum number of *uplinks*, *downlinks*, and *crosslinks* available to the node. An *uplink* is a user that connects directly to the K-Net node without using another network node. A *downlink* is a connection from the K-Net node directly to a user. A *crosslink* is a connection between two nodes. Changing NETLINKS will cause a soft reset as memory is reallocated. Each NETLINK allocated requires 130 bytes of RAM.

NETROUTE n

(n = 1 - 32)

default 5

This command sets the maximum number of neighbor nodes that can be listed with the ROUTES command. A neighbor node is a network node that this K-Net node can hear "direct", without the benefit of being relayed by another network node. As with the NETDESTS parameter, if the setting of NETROUTE is too low, not all of this nodes neighbors will be able to be listed. Setting NETROUTE to a higher number will allow local node activity to still be noted, and still have some room in reserve to handle itinerant nodes and nodes that will show up during a band opening. Changing NETROUTE will cause a soft reset as memory is reallocated. Each NETROUTE allocated requires 39 bytes of RAM.

NETUSERS n

(n = 0 - 26)

default 5

NETUSERS sets the maximum number of *uplinks* and *downlinks* from the node. This command places a limit on how many users can access or be accessed by the node. The NETUSERS command will not cause a soft reset when changed or entered.

NEWMODE {ON | OFF}

default ON

When ON, the TNC will return to Command Mode if the station on the current I/O stream disconnects. The TNC will not return to Command Mode if the station disconnecting is on a different stream. When OFF, a disconnect will not cause the TNC to change modes.

See also: connect, disconnect, status

NOMODE {ON | OFF}

default OFF

When OFF and a connection takes place, the TNC will change to whatever mode is specified in CONMODE. When ON, the TNC stays in Command Mode after connecting to another station; that is, it does not immediately change to Convers or Transparent Mode.

See also: conmode, connect

NTEXT text

(up to 128 characters)

default (blank)

This entry specifies customized text to be sent with the initial KA-Node sign-on message (when a remote station connects to a KA-Node). Enter any combination of characters and spaces up to a maximum length of 128. Entering a single '%' will clear NTEXT.

See also: mynode, numnodes

NUMNODES n

(n = 0-26, depending on available RAM)

default 0

This command is used to set the number of allowable circuits through the KA-Node. For example, to allow up to 3 simultaneous circuits through the node, set NUMNODES 3.

- Changing the value of NUMNODES will cause a soft reset.

Approximately 4 Kbytes of RAM is used for each circuit. The amount of RAM available will depend on how much RAM has been used for the PBBS, MAXUSERS, and MYREMOTE parameters. When setting n larger than available RAM will allow, a "Not enough RAM" message will be returned. The amount of RAM required for the PBBS (personal bulletin board) is set first, then set the desired number of Ka-Node circuits.

Note: While the full command name is required to change the value of NUMNODES, it can be entered in the short-form NU, to see the current value of NUMNODES.

See also: mynode; KA-Node section

PACLEN n

(n = 0 - 255)

default 128

This command specifies the maximum length of the data portion of a packet. The TNC will automatically send a packet when the number of input bytes reaches n.

This value is used in both Convers and Transparent Modes. A value of 0 means 256 bytes.

See also: maxframe

PACTIME [EVERY | AFTER] n

(n = 0 - 255)

default After 10

This command is used to pace the formation (and transmission) rate of packets in Transparent mode or in Convers mode if CPACTIME is ON. In addition, a PACTIME timer is necessary in transparent mode, since the SENDPAC character is ignored and treated as data. The timer is not started until at least one byte is entered for a packet; hence, a zero data length packet is never formed and sent.

When AFTER is specified, bytes are packaged (if any available) when input from the terminal stops for n*100 ms or when PACLEN is reached. When EVERY is specified, input bytes are packaged (if available) and queued for transmission every n*100 ms or when PACLEN is reached.

See also: cpactime, trans

PASS n

(n = \$00 - \$FF)

default \$16 <Ctrl+V>

This command selects the ASCII character used to allow the passage of any character in Converse Mode.

For example, if a COMMAND character (<Ctrl+C>) is to be sent as part of packet data, it can be done so by preceding it with the PASS character. The command character will be sent rather than returning the TNC to Command Mode. In Transparent Mode all characters are passed, there are no special functions except the one combination to get out of transparent mode. However, see TRFLOW.

PASSALL {ON | OFF}

default OFF

This command enables the attempt to display corrupted packets when PASSALL is ON. If a packet contains beginning and ending flags, the TNC will attempt to

decode the address field as well as the data field and display the packet as specified by other commands such as MONITOR. MHEARD and NDHEARD logging are disabled when PASSALL is ON.

- When PASSALL is OFF, packets will be displayed only if the CRC (error checking) is correct and as specified by other commands such as MONITOR.

❖ **PBBS [n | STATUS]**

(maximum value of n depends on available RAM)

default 100

Setting n greater than 0 allocates memory and activates the Personal Mailbox in the TNC. The amount of memory allocated will be n kilobytes and may be limited by other functions that require memory (e.g. NUMNODES and MAXUSERS). Changing the size of the PBBS memory allocation will not affect the contents of the mailbox (messages will be preserved). If attempting to set the PBBS smaller than is required for existing messages the message "Messages would be lost" will be displayed and no change will be made to the mailbox size. In this case to reduce the size of the mailbox some messages must first be deleted (killed) the PBBS be set to 0, which will delete all messages; then set the new size. Using the PBBS n command with n equal to the current size will renumber the messages in the mailbox beginning with message number 1. If n is a different size, the messages will not be renumbered. This command causes a soft reset if n is different from its previous value.

Note: To change the value of PBBS the whole command name must be entered, but to just see the current value, only the short-form name needs to be entered (PB).

See also: cmsg, mypbbs, pbheader, pblo, pbperson, ptext; PBBS section

PBFORWRD [bbSCALL [VIA call1,call2,...call8]] [EVERY | AFTER n]

default NONE EVERY 0

This command will cause the PBBS to attempt to initiate a forward of any eligible mail to another BBS periodically. Any message in the PBBS which contains an @BBS field and is not being HELD (has a status of H) or has not previously been FORWARDED (has a status of F) is eligible for forwarding. If the keyword "EVERY" is used, the PBBS will attempt to forward once every n hours. If the keyword "AFTER" is specified, the PBBS will attempt to forward whenever a user

disconnects from the PBBS, and every n hours after that. Setting the time interval will cause the PBBS to attempt to forward immediately.

After the PBBS has initiated a forward, it will also request a reverse forward if PBREVERS is ON. Note that full-service BBS systems must be configured to allow this station to reverse forward mail to this PBBS.

Note: The PBBS will not forward or reverse forward unless HTEXT is set.

See also: pbrevers, htext

PBHEADER {ON | OFF}

default ON

When ON the routing headers received from a full service BBS will be stored in the PBBS mailbox. When OFF, these headers are not stored in the mailbox, allowing messages to require less space. The routing headers are those lines normally seen in messages beginning with R:. Note that the PBBS will ignore all lines beginning with R: until it sees the first line that does not have R: in column one. From that point on, all of the messages will be stored, even if a line begins with an R:.

See also: PBBS section

PBHOLD {ON | OFF}

default ON

When ON, any message received over the radio will automatically be held by this PBBS for SYSOP review. The message may then be released for forwarding by editing its H status flag (for example, to change the HELD status of message number 4: enter E 4 H at the PBBS prompt).

When set to OFF, messages received over the radio are not held but may immediately be forwarded. (Note that any message addressed TO or @ the TNC MYCALL or MYPBBS call will be held regardless of the setting of PBHOLD.)

PBKILLFW {ON | OFF}

default OFF

When OFF, Private and Traffic messages will not be deleted (killed) after they have been forwarded but will be marked with an F flag (forwarded).

When ON, Private and Traffic messages will be deleted from the PBBS after they have been successfully forwarded.

PBLIST [ON | OFF] [NONE | {+|-}callsign | callsign1,callsign2...]

default OFF

The front panel of this TNC has a mail-status indicator that blinks when there is unread mail in its PBBS addressed to the TNCs callsign (MYCALL) or MYPBBS. Several PBBS commands (LISTMINE, READMINE, and KILLMINE) operate on mail addressed to MYCALL or MYPBBS.

PBLIST allows the entry of up to ten callsigns to the set of callsigns included as “mine” for use with the “mail” indicator and the PBBS commands (LISTMINE, READMINE, and KILLMINE). The PBLIST command has two parameters: one for creating, changing, or deleting the list of callsigns on the PBLIST and the other for turning ON or OFF the use of the callsigns on the current PBLIST.

- SSID's attached to callsigns are always ignored in the mailbox.

The maximum number of callsigns allowed in PBLIST is 10. Up to 10 callsigns may be entered at once, using the format shown after the command name, and the new list of from 1 to 10 callsigns will replace whatever list was there before. The PBLIST command can be used to add a single callsign (+callsign), so long as there is room for the new callsign on the list (if there is not, the TNC responds with “EH?” and returns its command prompt). Similarly, a single callsign (-callsign) can be removed from the list. And to remove all items from the current list, enter PBLIST NONE.

When the PBLIST is changed, the new list of callsigns takes effect immediately for use with the PBBS commands (LISTMINE, READMINE, and KILLMINE). But the new list will not be used with the mail-status indicator until after the next disconnect from the PBBS or the next reset.

When PBLIST is ON, the callsigns on the current PBLIST list are included as “mine” for use with the “mail” indicator and the PBBS commands (LISTMINE, READMINE, and KILLMINE). MYCALL and MYPBBS are also included as “mine” when PBLIST is ON.

When PBLIST is OFF, only MYCALL and MYPBBS are included as “mine” for use with the “mail” indicator and the PBBS commands (LISTMINE, READMINE, and KILLMINE).

PBLO [OLD | NEW] [FIXED | VARIABLE]

default NEW VARIABLE

When set to OLD the PBBS will list messages to the user from oldest to newest (i.e. ascending numerical order). When set to NEW, the newest message will be listed first. When the second parameter is set to FIXED, the user cannot change the listing order. When the second parameter is set to VARIABLE, the user may change the order in which messages will be listed by using the LO command within the PBBS.

See also: PBBS section

PBMAIL {ON | OFF}

default OFF

When ON, a “YOU HAVE MAIL” message will be sent to the terminal when new and un-read messages are available for MYCALL.

PBPERSON {ON | OFF}

default OFF

When set to OFF, the personal mailbox will allow messages to be sent to any callsign

When set to ON, only messages addressed to the MYCALL, MYPBBS or PBLIST callsigns will be accepted over the radio, but a message entered from the terminal or by the SYSOP may be addressed to anyone.

See also: mycall, mypbbs, pbbs

PBREVERS {ON | OFF}

default ON

When set to ON, the PBBS will request a reverse forward from another BBS after it has finished forwarding any mail to the other BBS. This does not affect the ability of another BBS to reverse forward from your PBBS, but only affects the action taken by the PBBS after it has initiated a forward (PBFORWRD command).

See also: pbforwr

PBSIZE n **(n = 0 - 65535)**

default 0

This command controls the maximum size of new messages entered into the PBBS. A value of zero means there is no maximum.

PBUsers n **(n = 1 - 10)**

default 1

This command controls the maximum number of connects allowed to the PBBS. This command causes a soft reset if n is different from its previous value.

PERSIST n **(n = 0 - 255)**

default 63

This command, used with SLOTTIME, implements an algorithm for channel access to send packets. The algorithm used to determine whether or not to transmit using the PERSIST/SLOTTIME method has been shown to be more efficient than the DWAIT method used by most standard AX.25 packet stations. The result of using the persistence algorithm is increased throughput under most channel conditions.

The value of "n" is used to determine if a packet will be sent after SLOTTIME expires. For example, let's assume a PERSIST setting of 63 and a SLOTTIME setting of 10. This slottime setting corresponds to 100 milliseconds. When the TNC detects that the channel is clear and available (no carrier is detected), it starts a timer (SLOTTIME). When the timer expires, (100 ms in our case), the TNC generates a random number between 0 and 255. If the generated number is equal to or less than the PERSIST value, the TNC keys up the transmitter and sends the data packet. With our setting of 63 the odds of this occurring after the first slottime are 1 in 4. (Actually the probability is PERSIST plus 1 divided by 256.) If the algorithm generated random number is greater than PERSIST, the

timer is restarted and will have to expire again before a new random number is generated. This is repeated until the TNC gains channel access and sends its packet of information.

Making SLOTTIME smaller will cause the TNC to generate the random number more frequently, whereas raising the PERSIST value will give a better chance (improve the odds) of transmitting the data. Through careful choice of these values, it is possible to improve data throughput while at the same time permitting shared channel usage by other packet stations. The persistence algorithm has been added on top of the DWAIT algorithm.

See also: slottime

PHEARD [CLEAR]

immediate

This command will display a list of the last 10 stations that have connected to the PBBS in this TNC. The list will show the callsign of the connecting station and the last time they connected and disconnected.

To clear the PHEARD list of all current entries, enter PHEARD CLEAR.

See also: passall

PID {ON | OFF}

default OFF

When OFF, only those packets with a protocol ID of \$F0 (pure AX.25) are displayed. When ON, all packets are displayed. Some of the information in non-AX.25 packets (for example: TCP/IP, NET/ROM or TheNet) can cause some computers to lock up. Net/Rom, TheNet and G8BPQ nodes have a PID of \$CF, TCP/IP uses \$CC and \$CD, and standard AX.25 is \$F0.

PMODE {CMD | CONV | TRANS}

default CMD

The PMODE command controls the mode interface mode this TNC will be in when it is first powered up or reset. When set to CMD, this TNC will display its sign-on message followed by the command prompt. When set to CONV, the unit will be in

the Convers Mode. Setting PMODE TRANS will set this TNC to power up in Transparent Mode. Exiting Transparent Mode with three <Ctrl+C>s will return this TNC to Command Mode.

PTEXT text

(up to 128 characters)

default (blank)

This entry specifies the customized text sent to a station that connects to this TNCs PBBS (personal mailbox). Enter any combination of characters and spaces up to a maximum length of 128. Entering a single “%” will clear PTEXT. Don’t use the “>” character in PTEXT, as this is reserved by BBS systems for their prompt.

See also: pbbs

RANGE a-b/c-d/e-f/g-h/i-j

default 0-255/0-255/0-255/0-255/0-255

The 5 A/D inputs are normally read by the microprocessor as a decimal value from 0 to 255, representing a voltage from 0 to +3.3 V on the pin. The RANGE command will mathematically convert that decimal value to any offset and scale desired. If an input is to be expressed as a voltage with 2 decimal precision, for example, set the range for that analog pin as 0-5.00. If a temperature sensor is connected to an A/D input pin, which has an output of 0-3.3 V for temperatures of –40 to 100, set the range for that input to –40 – 100.0. A pressure sensor that gives 0-4 V for 1 PSI would use a range of 0-1.25.

See also: analog, telemetry

REDISPLA n

(n = \$00 - \$FF)

default \$12 <Ctrl+R>

This command is used to change the REDISPLAY input editing character. The parameter n is the ASCII code for the character to be entered in order to REDISPLAY the packet or command currently being entered.

This character causes the TNC to redisplay the data entered for a packet to send, but a packet has not yet been transmitted. When the REDISPLAY-packet character is entered, the following things happen: First, type-in flow control is released (if FLOW was enabled). This displays any incoming packets that are

pending. Then a \ (backslash) character is displayed, and the packet or command that was begun is redisplayed on the next line. If any character has been deleted or re-typed, only the final form of the packet will be shown. Then, packet data entry can be continued. Incoming packets will continue to be displayed until the next character of your current message is entered.

Use this character when typing a message in Convers Mode and a packet comes in. An incoming message can then be seen, before sending the packet, without canceling input.

See also: bkondel, canline, canpac, flow

RELINK {ON | OFF}

default OFF

When OFF, the TNC operating with AX25L2V2 ON does not attempt to automatically reconnect. When ON, the TNC operating with AX25L2V2 ON will attempt to automatically reconnect after RETRY is exceeded.

The KA-Node and the PBBS will never attempt to reconnect regardless of the setting of this command. If using AX.25 Level 2 Version 1 (AX25L2V2 OFF) this command has no effect.

See also: ax25l2v2, retry, tries

❖ RESET

immediate

This command is used to perform a soft reset. Contents of the mailbox (PBBS) are kept, and the NDHEARD and MHEARD logs are not cleared. Any existing connections, which are not conpermed, will be lost, even though the other station thinks this connection still exists. This TNC will display its initial sign-on message.

See also: intface, maxusers, myremote, numnodes, pbbs, restore

RESTORE DEFAULT

immediate

When RESTORE D is given, this TNC will revert to its factory default settings, perform its AUTOBAUD routine, and erase PBBS memory (deleting all messages and non-default PBBS parameter settings). This clears this TNCs RAM, the as having performed a Hard Reset.

See also: reset

RETRY n

(n = 0 - 15)

default 10

This command specifies the number of packet retries. Packets are re-transmitted n times before an operation is aborted. The time between retries (when no acknowledgment is received) is specified by the command FRACK.

See also: ax25l2v2, frack, relink, tries

RING {ON | OFF}

default ON

When ON, three bell characters (\$07) are sent to the terminal with each "*** CONNECTED TO" message (when another station initiates the connect).

RNRTIME n

(n = 0 - 255)

default 0

RNRTIME is set in 10-second increments. If a connection stays in a remote device busy state (continues to receive RNR frames) for RNRTIME, the TNC will disconnect. If a KA-Node connection stays in a remote device busy state for RNRTIME the KA-Node will disconnect the input and output sides of the KA-Node circuit. Setting RNRTIME to 0 disables this function.

See also: mresp

RPRINT text

(text up to about 250 characters)

immediate

This is an immediate command which sends the "text" string to the device attached to the computer interface port. It is intended to allow a remote SYSOP

(connected to this TNCs MYREMOTE callsign) to send a string to a terminal attached to the computer interface port of this TNC.

RTEXT text

(text up to about 250 characters)

default (blank)

This command sets the password string for use when accessing the MYREMOTE or when performing SYSOP functions in the PBBS remotely.

When connecting to the MYREMOTE or attempt to enter the SYSOP mode of the PBBS (or the network node, when enabled), the TNC will send three sets of numbers, one of which must be properly decoded (matching position number with string character) and returned as the password.

See the “Remote Access” section for more information.

SCREENL n

(n = 0 - 255)

default 0

This value is used to format what is sent to terminal. A “CR” sequence is sent to the terminal at the end of a line when n characters have been printed. A value of zero inhibits this action.

See also: autolf

SENDPAC n

(n = \$00 - \$FF)

default \$0D <Ctrl+M> (CR or ENTER)

This command specifies a character that will force a packet to be sent in Convers Mode. In the Convers Mode, packets are sent when the SENDPAC character is entered or when PACLEN is achieved.

See also: cpactime, cr

SLOTTIME n

(n = 0 - 255)

default 10

n specifies the amount of time, in 10 millisecond increments, between successive tries of the persistence algorithm.

See also: persist

START n

(n = \$00 - \$FF)

default \$11 <Ctrl+Q>

This command specifies the character sent by the computer to the TNC to restart output from the TNC. If set to \$00 only hardware flow control will be used. For software flow control, set this parameter to the character the computer will send to restart data flow.

See also: stop, xflow, xoff, xon

❖ STATUS [LONG]

immediate

Entry of the STATUS command will display the number of bytes available (i.e., free bytes) in the computer interface input buffer, the current I/O stream, and any streams having a connected status.

Entering LONG will cause the TNC to display the status of all streams allowed by MAXUSERS, PBBS, NUMNODES, and MYREMOTE.

The current KA-NODE input and output (IO) stream is also indicated. A pound sign '#' (octothorpe) indicates that there is unacknowledged data in the buffers for that stream. The number immediately following the # is the number of bytes outstanding. The number in parentheses is the number of packets not yet acknowledged.

The following is an example of a display (using STATUS LONG). A, B, C, through J (i.e., MAXUSERS = 10) indicate the stream. "A" stream is connected to KEØSM and has 50 bytes in 2 packets that are unacknowledged. The current stream (IO) is the "B" stream, which is connected to NØKN. KBØNYK is connected on circuit "A" of the KA-Node. All other streams, node circuits, the BBS, and the remote access are disconnected. Appending "/P" to the callsign indicates that the connection is permanent (see CONPERM).

```
cmd: status long
FREE BYTES 1661
```

A stream - #50(2) CONNECTED to KEØSM
B stream - IO CONNECTED to NØKN/P
C stream - DISCONNECTED
J stream - DISCONNECTED
BBS DISCONNECTED
REM DISCONNECTED
Ain CONNECTED to KBØNYK
Aout DISCONNECTED
Bin DISCONNECTED
Bout DISCONNECTED

See also: conperm, maxusers, myremote, numnodes, pbbs, streamsw

STOP n

(n = \$00 - \$FF)

default \$13 <Ctrl+S>

This command specifies the character sent by the computer to the TNC to stop output from the TNC. If set to \$00 only hardware flow control will be used. For software flow control, set this parameter to the character the computer will send to stop data flow.

See also: start, xflow, xoff, xon

STREAMCA {ON | OFF}

default OFF

When set to ON, receiving locally connected packets (MONITOR OFF and/or MCON OFF), enable the display of the callsign of the connected-to station following the stream identifier of the connection (controlled by STREAMEV). This is especially useful when operating with multiple connections allowed.

See also: mcon, monitor, streamev

STREAMEV {ON | OFF}

default OFF

When OFF, the stream indicator is displayed only when a change in streams occurs. When ON, the stream indicator will be displayed with every incoming

packet. This command takes effect only when receiving locally connected packets and MCON is OFF.

See also: mcon, monitor, streamca, streamsw

STREAMSW n

(n = \$00 - \$FF)

default \$7C (I)

This command selects the character to be used to signify that a new “stream” or connection channel is being addressed. To change streams, type this streamswitch character followed immediately by the stream designator. The stream designator is an alphabetic character A through Z, or a through z, limited by the value of MAXUSERS.

- If STREAMSW is set to the dollar sign (\$24), parameter values must be entered as decimal numbers. Alternatively, precede the \$ with the PASS character to enter hex numbers.

The character selected can be PASSed in the Convers Mode by using the special PASS character and will always be passed as data in the Transparent Mode. When operating in Transparent Mode and need to change streams, first return to the Command Mode.

See also: maxusers, pass, status

SUPLIST [ON|OFF] [NONE | {+|-}call | call1,call2,..]

where call={callsign | {<|>}callsign |

callsign{>|<>}callsign}

default OFF NONE

SUPLIST is used to determine which received packets will not be displayed, i.e. suppressed. When OFF or NONE, SUPLIST will not prevent the display of packets, even if it has callsigns in its list.

The maximum number of callsigns allowed in SUPLIST is 10. Up to 10 callsigns may be entered at once, using the format shown after the command name, and the new list of from 1 to 10 callsigns will replace whatever list was there before. The SUPLIST command can be used to add a single callsign (+callsign), so long as there is room for the new callsign on the list (if there is not, the TNC responds

with “EH?” and returns to the command prompt). Similarly, a single callsign (-callsign) can be removed from the list. To remove all items from the current list, enter SUPLIST NONE.

A callsign entered without an SSID will match any SSID of that callsign. To match a specific SSID only, enter the callsign with that SSID: for example, callsign-n, where n = 0-15.

Selective display is enabled when one or two callsigns are combined with a “>” and/or “<” character. For example, entering “>callsign” will disable display of packets TO that callsign (assuming SUPLIST is ON), entering “<callsign” will disable display of packets from the callsign, entering “callsign1>callsign2” will block display of packets from callsign1 to callsign2, and entering “callsign1<>callsign2” will block display of packets from callsign1 to callsign2 and vice versa.

- Note that entering call>call or call<>call counts as two of the 10 maximum allowed callsigns.

See also: monitor

SWP u,d,t

default 17,17,108

This command sets the parameters used for software carrier detect. The first number (u) is used to increment (up count) a counter when a valid mark/space or space/mark transition occurs in the received signal (i.e. transition occurs at the beginning of a bit time). The second number (d) is a penalty subtracted from the counter (down count) when a transition occurs in the middle of a bit time. The t value is the threshold value - when the counter total reaches this value, the carrier detect will be set true.

The values need to be separated by commas as they are entered (as shown). Also as shown, spaces between values are not necessary—but entering spaces after the commas is OK.

Once carrier detect is active, the counter must drop to 0 before carrier detect is again made false.

See also: cd

TELEMETRY n

(n = 0 - 255)

default 0

A telemetry beacon is sent every $n \times 10$ seconds. If $n = 0$, the beacon is disabled. The telemetry beacon is a UI frame sent to BEACON. If there is a current UNPROTO digipeat path, it will be used to transmit the telemetry beacon.

The data in the UI frame is in the form: "T#nnn,an0,an1,an2,an3,an5,bbbbbbbb", where:

- "nnn" is a decimal count for the telemetry beacon and is incremented each time the beacon is sent. When first used, or after the unit is reset, the beacon counter begins at 001 (T#001). After reaching 999 (T#999), the counter cycles around to 000 (T#000).
- "an0", "an1", "an2", "an3", and "an4" are the decimal readings on the corresponding analog inputs of the microprocessor.
 - **Note:** APRS expects leading zeroes for the analog readings. RANGE must be set to 0-255 to beacon samples in APRS format.
- "bbbbbbbb" is the binary reading of the 8 analog input pins on microprocessor, where "b" = "1" means a Logic High on the pin and "b" = "0" means a Logic Low on the pin. The first "b" in the string of 8 binary digits is the most significant bit of PORT E and the last "b" is the least significant bit of PORT E.

See the documentation on APRS for more information on the use of telemetry along with GPS.

See also: analog, range

TRACE {ON | OFF}

default OFF

When ON, all received frames are displayed in their entirety, in hexadecimal, including all header information. All packets, which are eligible for monitoring also, will be displayed in normal text.

TRANS

immediate

This command causes immediate exit from Command Mode into Transparent Mode. The current link state is not affected. There are no special editing characters, all characters are sent out as received. To get out of Transparent, send the TNC a modem break if BREAK is ON, or see CMDTIME for a special keyboard sequence.

See also: break, cmdtime, command

TRFLOW {ON | OFF}

default OFF

This command allows the TNC to respond to software flow control from the computer while in the Transparent Mode. When TRFLOW is OFF, software flow control is not used in the Transparent Mode. Hardware flow control will be expected from the computer by the TNC. The computer program needs to be set to use hardware flow control.

When ON, software flow control is enabled and the START and STOP characters are sent by the computer to the TNC to control the flow of data. When START and STOP are set to \$00, hardware flow control must be used. If not zero, the TNC will respond to the computer's START and STOP characters and remain transparent to other characters from the terminal or computer for flow control.

When START and STOP are set for software flow control (normally <Ctrl+Q> and <Ctrl+S>) all characters can be received in Transparent Mode (including the START and STOP characters) by setting TRFLOW ON and TXFLOW OFF. However, the START and STOP characters cannot be transmitted, since the TNC will interpret them as flow control.

See also: trans, txflow, xflow

TRIES [n]

(n = 0 - 15)

The TRIES command will display and optionally set the number of attempts which have been made to re-send a packet (on the current stream) which failed to reach its destination. For instance, if RETRY is set to 10, TRIES will show how many attempts have already been made to pass the data. For example, if TRIES were to show 8, "TRIES 3" would reset the counter to make the TNC believe that it had

only tried 3 times so far, thus allowing 7 more attempts before the RETRY limit is exceeded.

See also: retry

❖ **TXDELAY n**

(n = 0 - 255)

default 30

This command sets the transmitter key-up delay as 10*n ms. This setting establishes the time delay between the application of push-to-talk and the start of packet data to the transmitter. Flags (characters to begin packet) are sent during the delay.

This command needs to be set long enough to allow the transmitter time to come to full power before data is sent.

If set too short, the beginning of the transmitted packet will not be heard and will be discarded by other stations. If set too long, additional flags at the beginning (heard as a repetitive sound) waste air time. Also, it may be necessary to increase TXDELAY to allow other receiving stations sufficient time to detect transmitted packets.

TXFLOW {ON | OFF}

default OFF

This command allows the TNC to send software flow control (XON and XOFF) to stop and restart the flow of data from the computer while in the Transparent Mode. When TXFLOW is OFF, hardware flow control must be used between the computer and TNC. RTS and CTS must be connected between the TNC and computer for hardware flow control. When TXFLOW is ON, software flow control between the TNC and computer in Transparent Mode will depend on the setting of XFLOW. XFLOW ON enables software flow control, XFLOW OFF disables it. When software flow control is enabled, the TNC will send the XON and XOFF characters to the computer to control data flow.

When set for software flow control, all characters can be sent in Transparent Mode (including the XON and XOFF characters) by setting TXFLOW ON, XFLOW ON and TRFLOW OFF. However, START and STOP characters won't be displayed, since the terminal program should interpret them as flow control.

See also: trans, trflow, xflow

TXTAIL n**(n = 0 - 100)**

default 12

This command sets the transmitter key-OFF delay as n ms. This setting establishes the time delay between the last HDCL flag (end of packet data) and release of push-to-talk to the transmitter.

It needs to be set long enough to give the transmitter time to have transmitted the complete packet before the PTT control signal is released.

If set too short, the ptt may be released too soon, and the end of a packet may not be transmitted. If this is the case, other stations will never be able to decode the transmitted packet.

This command was added specifically when the TNC is used with radios that have a mismatch between the transmit audio, and PTT. This can cause loss of packet data, when PTT is released before the end of packet audio has been transmitted.

When this is the case, increase this setting to 20 or 30.

UIDIGI ON [+|-] call1[,call2[,call3[,call4]]]

default OFF NONE

Up to 4 call signs can be specified for special digipeater duty. If any of the UIDIGI calls appears in the to-be-digipeated field of a UI packet, and if MYCALL does not appear in the source field or any of the has-been-digipeated fields, the UIDIGI call in the to-be-digipeated field will be replaced by MYCALL with the H bit set and the packet will be digipeated.

See also: dwait, persist, uidwait, unproto

UIDUPE n**(n = 0-255)**

default 0 (disabled)

Timer for UI duplicate checking. Use of this “do not duplicate within n seconds” feature helps control the number of UI frames digipeated.

See also: uiflood and uitrace.

UIDWAIT [ON | OFF]

default OFF

When UIDWAIT is OFF, “special” digipeat packets (those formed by UIDIGI, UIFLOOD, or UITRACE only) have their usual channel access; there is no wait DWAIT or slottime added before transmission once the channel is clear. However, if UIDWAIT is set ON, the packets waiting to be digipeated will be subject to the same wait times as not-to-be-digipeated packets awaiting transmission.

- By subjecting “special” to-be-digipeated packets to a delay determined by slottime and persist, it is more likely that to-be-digipeated packets of two or more stations in the same vicinity would not collide. This may be good if one wants to guarantee that a digipeated packet will “make it out” of its neighborhood but bad if one wishes to limit the number of times a packet is re-digipeated, such as in APRS applications.

See also: dwait, persist, uidwait

UIFLOOD name[,time[, {FIRST | ID | NOID}]] (name = 5 char max)(time = 0-255)

default disabled,30,NOID

When a UI frame is received with a call in the to-be-digipeated field of the form ‘name’ x-y where x is a number (1-7) appended to ‘name’ and y is an SSID (1-7), the SSID is decremented and the UI frame is digipeated without setting the H bit. When the packet is digipeated, a checksum is formed over the source, destination, and data fields of the packet. This checksum is kept for time seconds (0-255). If an incoming UI packet is eligible for digipeating as above, but its checksum matches one of those being saved, the packet is discarded (not digipeated). The buffer holds a maximum of 64 checksums. If the optional parameter ID is selected, the MYCALL call sign is inserted in an additional digipeater address field with its H bit set.

See also: uidwait

UITRACE name, n (name = 5 char max)(n = 0-255)

default disabled, 30

When a UI frame is received with a call in the to-be-digipeated field of the form ‘name’ x-y where x is a number (1-7) appended to ‘name’ and y is an SSID (1-7),

and MYCALL does not appear in the source field or any of the has-been-digipeated fields, MYCALL with the H bit set is inserted before the to-be-digipeated field, the SSID of the to-be-digipeated field is decremented, and the UI frame is digipeated without setting the H bit of the to-be-digipeated field. The value of n sets a delay time between duplicate digipeats. Use of this “do not duplicate within n seconds” feature helps control the number of UI frames digipeated.

- If the packet should already have 8 digipeater fields, MYCALL is not inserted.

See also: `dwait`, `persist`, `uidwait`

UNPROTO {call1 [VIA call2,call3..call9] | NONE}

default CQ

call1 = destination address (this is just a “dummy” address, as no connection takes place, people often put their name or CQ here)

call2 ... call9 = optional stations to be digipeated through. A maximum of 8 digipeat addresses (callsigns or aliases) can be specified. This is referred to as a path.

Each callsign may also have an optional Secondary Station Identifier (SSID) specified as `-n`, where `n = 1 - 15`. The digipeat callsigns are specified in the order in which they are to relay transmitted packets. This command is used to set the digipeat and destination address fields for packets sent in the unconnected protocol (unproto) mode. Unproto packets do not receive an acknowledgment and are not retried. They are sent as Unnumbered I-frames <UI>.

The digipeater list is also used when transmitting BEACON and ID packets. If UNPROTO is “NONE”, no unconnected information type packets will be sent except for BEACON and ID.

Unconnected information type packets heard from other stations can be monitored when MONITOR is set to ON. To show those packets while in connected state, MCON must also be set to ON.

See also: `beacon`, `id`, `monitor`, `mrpt`, `xmitok`

USERS n (n = 0 - 26)

default 1

This command specifies the channels (streams) which may be available to incoming connect requests. For example, if USERS = 5 then an incoming connect request will connect to the lowest channel A - E, if any of these channels are in the unconnected state. If none of the 5 channels are available (all of them are connected), a <DM> packet will be sent back to the requesting station and the message "*** connect request: (call)" will be output to the terminal, if permitted. If USERS is set to 0 connection requests from other stations will receive no response. If an attempt is made to set USERS to a number higher than the setting of MAXUSERS, the extra is ignored and the message "USERS LIMITED BY MAXUSERS" will be displayed.

See also: intface, maxusers, streamsw

❖ **VERSION**

immediate

This command causes this TNC to display its current bios/firmware version along with the name of the unit.

XFLOW {ON | OFF}

default ON

When ON software flow control will be implemented according to the settings of START, STOP, XON, XOFF. For normal software flow control set XFLOW ON, START \$11, STOP \$13, XON \$11, XOFF \$13. The TNC expects the computer or terminal to respond to the flow control characters XON and XOFF, and the TNC responds to the START and STOP characters from the computer. When XFLOW OFF, the TNC will only use and recognize hardware flow control lines (CTS and RTS) to start or stop data. The RS232 cable must be wired appropriately. If the software flow control characters are set to \$00, software flow control is not possible.

In the Transparent Mode, flow control is also determined by the settings of TRFLOW and TXFLOW.

See also: start, stop, trflow, txflow, xoff, xon

XKCHKSUM {ON | OFF}

default OFF

This command only applies when INTERFACE is set to XKISS. The value of XKCHKSUM controls whether the XKISS CHECKSUM mode is active.

See also: xkpolled

XKPOLLED {ON | OFF}

default OFF

This command only applies when INTERFACE is set to XKISS. XKPOLLED controls whether the XKISS POLLED mode is active.

See also: xkchksum

XMITLVL n

(n = 0-502)

default 100

This command sets the output level of the TNCs transmit audio to a radio.

When adjusting this setting, it may be more convenient to use the CAL command.

See the CALIBRATION/EQUALIZATION section for more information.

XMITOK {ON | OFF}

default ON

When ON, transmitting functions are enabled. If turned OFF, transmitting is inhibited while all other functions of the TNC are unchanged.

XQFF n

(n = \$00 - \$FF)

default \$13 <Ctrl+S>

This command selects the character sent by the TNC to the computer to stop input from the computer. If set to \$00, hardware flow control must be used. For

software flow control set this parameter to the character the computer expects to see to stop sending data to the TNC.

See also: xflow, xon

XON n

(n = \$00 - \$FF)

default \$11 <Ctrl+Q>

This command selects the character sent by the TNC to the computer to restart input from that device. If set to \$00, hardware flow control must be used. For software flow control set this parameter to the character the computer expects to see to restart sending data to the TNC.

See also: xflow, xoff

Appendix A: Advanced Installation

This appendix provides additional information about installation and configuration beyond that given in the chapter on **“Installing the TNC”**. We assume that chapter has already been read.

Precautions

This TNC is grounded through its connections to your transceiver, computer, and power supply. Make sure the transceiver and other equipment connected to this TNC has equal ground potential. Follow the grounding instructions in your transceiver manual.

This TNC contains a Lithium battery for its internal SRAM and Real-Time clock. Note that this battery type may leak or explode if shorted, recharged, exposed to fire or high temperature, or inserted incorrectly.

Cable Wiring

Caution: Make sure the power to the transceiver, computer, and the TNC is OFF before connecting any cables.

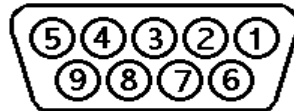
This TNC uses a standard USB type-A to USB type-B cable (like is used to connect a printer to the computer).

Software settings

When using a standard terminal program to communicate with this TNC, its serial port configuration must be set to use 8 data bits, no parity, and one stop bit. The computer interface bauds supported by this TNC are 300, 600, 1200, 2400, 4800, 9600, 19200, and 38400. In general, use the fastest baud the terminal program and hardware supports. The TNC defaults with ECHO ON, so you should set the terminal program for “full-duplex” operation.

The TNC Radio port

The radio port connector on the rear panel is a 9-pin (female) DB style (DSUB-9) connector, with pin layout as shown below:



Pin names;

1 TXA	Transmit audio (AFSK out)
2 XCD	External carrier detect
3 PTT	Push-to-Talk
4 CTLB (AN1)	Control line B (AN1 Analog input)*
5 RXA	Receive audio (AFSK in)
6 GND	Ground
7 EXT-IN	External input for Power/Reset*
8 CTLA (AN1)	Control line A (AN1 Analog input)*
9 GND/RESET	Ground (may be configured as external reset)*

* See the jumper section for details.

Pin Descriptions

Pin 1: TXA - Connects to the microphone input of a transmitter and provides the audio signal to be transmitted.

Pin 2: XCD - This pin is normally used as an external means of supplying carrier detect but can also be used as a “hardware” transmit inhibit. Alternately, it can be used for data from GPS or weather station devices (see GPSPORT).

Pin 3: PTT – Will be pulled to ground the TNC needs to transmit. This is an open-drain circuit switched by an FET (not to exceed 50 V dc or 200 mA).

Pin 4: CTRL-B - This pin is the CONTROL B output which may be used as a control line to a radio or other devices. It is an open-drain circuit switched by an FET. It a ground will be applied on this pin when controlled by the CTRL command (not to exceed 50 V dc or 200 mA). This pin can also be used as AN1 (Analog Input). See the Jumper section for details.

Pin 5: RXA - Receive audio input. This pin would normally be connected to the external speaker jack on a radio but may connect to the receive audio output pin on a microphone connector (if available), or to a receive audio output from a radios accessory port.

Pin 6: GND - This pin connects to the ptt ground of a radio mic connector.

Pin 7: EXT-V - This pin is not normally connected. Using an internal jumper inside, it may be configured as a supply voltage input or as an external reset. See the Jumper section for details.

Pin 8: CTRL-A - This pin is the CONTROL A output, which may be used as a control line to a radio or other devices. It is an open-drain circuit switched by an FET. A ground will be applied on this pin when controlled by the CTRL command (not to exceed 50 V dc or 200 mA). This pin can also be used as AN0 (Analog Input). See the Jumper section for details.

Pin 9: GND/RESET - This pin is configured as a ground as shipped from the factory. It may be re-configured as an external reset or as no function, by an internal jumper. See the Jumper section for details.

Interfacing Hand-Held Radios

This section is for those who need more information than that given in the “Installation” chapter.

Many transceivers, especially most hand-held models, obtain Push-To-Talk keying by completing a circuit between the mic input and PTT ground. A direct PTT input to the mic input line of units with this type electret condenser microphone is not usable without some type of isolation.

If planning to operate with a hand-held transceiver, an isolation circuit inside is available by the setting of configuration jumper labeled NOR/HT. To use the isolation circuit for connection to an HT, set the jumper from center pin to the pin marked HT. When using a different type radio (one that is not an HT), return that jumper to center pin to the pin marked NOR.

Interfacing to hand-held type radios can be done without performing this jumper modification by incorporating the same type of isolation circuitry inside the interface cable.

Ground return and speaker audio are both supplied through the external speaker jack of your hand-held.

Check for the exact requirements of your hand-held radio. The following guidelines may be useful:

- A 0.1 μ F capacitor (non-polarized) usually is required in the transmit audio line from the TNC to any handheld transceiver (HT) radio. Even if a capacitor is not required for connection to a particular radio, it can still be installed, and the transmit audio (packets) will pass through to the radio.
- ICOM HT radios key the PTT by providing a low impedance path from the mic input to ground. To accomplish this, simply install a resistor (approximately 3.9 k Ω seems to be a good value) in series with the PTT wire from this TNC, and connect this to the mic input along with the AFSK line.
- The ICOM 2A style wiring is most common, and can be used with some Alinco, Azden, and standard HTs.
- YAESU radios are similar but may use a mono plug and a different value (approximately 2.2K resistor).
- The wiring for Yaesu HTs can also be used with Radio Shack HTs.
- Most KENWOOD HT radios key the PTT line by connecting the sleeve of the mic connector to the sleeve of the earpiece connector. This means that you will not need a resistor in the PTT wire from the KPC-3XE, simply connect the PTT wire to the sleeve of the mic connector. Another point to watch: most of the KENWOOD HTs (2600 and later) use a three-pin mic connector. The AFSK from this TNC should therefore connect to the RING and not the TIP of the mic connector.
- Leave J9 in normal (not HT) when using Kenwood HTs.

- Verify the correct pinout of the HT first, to determine the proper type cable wiring.

Optional Connections to DSUB-9

Pin 7 of the DSUB-9 connector may be configured to allow power input or to perform a soft reset.

Pin 9 of the DSUB-9 may be configured as either a ground, or as a soft reset input.

Applying Power

To power this TNC from Pin 7 of the DSUB-9 connector, you must connect the jumper labeled VIN/RESET, center pin to the pin marked VIN.

Reset

To perform a soft reset from Pin 7, connect the jumper marked VIN/RESET, center pin to the pin marked RESET. Momentarily applying ground to pin 7 will then perform a soft reset. If no pins of the VIN/RESET are connected, pin 7 has no function.

To perform a soft reset from pin 9, connect the jumper marked RESET/GND center pin to the pin marked RESET. Momentarily applying a ground to pin 9 will then perform a soft reset. If the jumper center pin is connected to the pin marked GND instead, pin 9 can be used as a ground connection. If no pins are connected on the RESET/GND jumper, pin 9 has no function.

See the jumpers section for more information about their settings.

Appendix B: Advanced Information

Assembly and Disassembly

Note: Observe proper static protection procedures when handling the unit, and when its cover is removed.

Should access be required to the circuit board inside to reposition jumpers or for other purposes, disassemble as follows:

1. Turn off power and remove all cables from the unit.
2. Using a small Phillips screwdriver, remove the two (2) case screws (located on each side of the case) completely.
3. Remove the top cover from the unit. NOTE: In most cases, no further disassembly of the unit is necessary. All the internal jumpers are accessible at this point. This will need to be done to replace the memory backup battery, to change jumper settings, to perform a Hard Reset, or to install a replacement bios. Only if there is a need to make solder connections to the PC board (such as utilizing the extra analog inputs), should further disassembly be needed. When necessary, continue as follows:
4. Remove the hex nuts on the rear panel to secure the DSUB-9 connector to the case.
5. Remove the two (2) screws securing the front panel to the case.
6. Remove the front panel.
7. Pull the PC board forward to remove the PC board from the case.

To reassemble, reverse the procedure above.

Hard Reset/Self-Test

The hard reset process is provided to re-initialize the unit to its factory default values. This process may become necessary should operational problems be encountered. To start this procedure, power the unit OFF, then connect the pins of this jumper together. When next powered ON, the Self-Test will start, and the results of the test will be output to the terminal at 9600 baud. If the terminal is

not set to 1200 baud, the reset will still occur, but the display readout on the terminal may be garbled or absent.

Perform a Hard Reset, the procedure follows:

1. Make sure the unit is powered OFF.
2. Follow the disassembly procedure above:
3. Locate the Hard Reset jumper. Jumpers are appropriately labeled on the PC board.
4. Place the jumper on both pins. Reconnect the computer interface cable and power supply.
5. Power the unit ON.
6. With the terminal program set at 9600 baud, the Self-Test will start, and the test results from the TNC should be displayed as:

```
SELFTEST WILL CLEAR SETTINGS SAVED IN RAM
READY? (Y/N)
```

If the reply is Y, the test continues;

```
TESTING SRAM
```

```
.....
.....
.....
```

```
RAM OK
REMOVE TEST JUMPER
```

7. Turn power off.
8. Return the Hard Reset jumper to its normal position (one pin only).
9. Reassemble the unit in the reverse of the disassembly procedure and return to operation.

Note: After a hard reset, the unit will restart in autobaud mode, and all parameters are at factory default values previous mailbox messages are lost.

Calibration/Equalization

The CALIBRAT command is used to assist in adjusting the data transmitter drive level and aiding in adjusting the TNCs receive equalization jumper. The procedures listed below are a guideline for using the calibration command to adjust for optimum performance.

When CALibrate command is issued, the following options will appear on the terminal screen:

```
CALIBRATE MODE  HBAUD 1200

M send mark
R receive
S send space
T send square wave
- or + adjust XMITLVL while transmitting
X exit
XMITLVL 100
```

Pressing M will transmit a constant MARK (signal) until another key is pressed.

Pressing R will measure a received square wave.

Pressing S will transmit a constant SPACE until another key is pressed.

Pressing T will transmit a square wave (space/mark) until another key is pressed.

Pressing X will exit calibrate mode.

Drive level.

The drive level is set by a digital pot, providing a range of 1 mV to 2 V p-p. Default drive level is about 50 mV, corresponding to a XMITLVL command setting of 100. After entering the calibrate mode (CAL) the T option is used to initiate the transmission of a square wave. Then adjust the AFSK drive level by pressing the '-' or '+' key (and holding it down) until the drive level needed is achieved. It should be set so that the transmitted signal results in 2.5 to 3.5 kHz deviation from the radio.

Equalization.

If another Kantronics TNC user is nearby, that station can use their TNC in Calibrate mode to transmit a calibrate square wave. In this TNC (while in its CALibrate mode) choose the R option measure the status of receive equalization

(the ratio of the MARK/SPACE tones from that station), each time the R command is sent, the ratio of the space/mark tones will be re-calculated.

For best performance, these numbers should be about equal. A 40/60 to 60/40 ratio is acceptable. Change jumper EQ if needed. Default is no equalization, since it's normally not needed.

Note: When using a High Impedance Data output from the radio, set the jumper labeled INPUT Z on one pin only (OFF = 10 k Ω , ON = 620 Ω).

PTT (Push-to-Talk) Watchdog Timer

The KPC-3XE has a built-in PTT Watchdog Timer. The PTT output will be disabled if the TNC has been continuously transmitting for more than approximately 2.5 min.

To disable the Watchdog timer, connect the pins of jumper labeled PTT WATCHDOG together.

Microprocessor Watchdog Timer

The TNC includes a watchdog timer within the microprocessor. In the unlikely event that your TNC firmware gets into an infinite loop, the watchdog timer will detect the failure and force a reset of the TNC. When this happens, its power up sign-on message will be displayed.

A/D Converter

Two A/D converter inputs are available to be sampled when entering the Analog command. These inputs can be used to measure dc external voltages from 0 V to 3.3 V dc with up to 8-bit accuracy.

Note: If measuring voltages greater than +3.3 V dc, the signals need to be attenuated with an external voltage divider pair of resistors. (A pair of resistors, such as 100K over 33K should work for a 13.8 volt source.

So, if the jumpers for the A/D inputs are connected, the A/D inputs and control line outputs can be used at the same time. Note: If a CTRL command has been set to ON, the A/D unput that shares that connection will read as 0 (zero). If the A/D input jumpers are set to OPEN (not connected), those inputs are not connected externally.

- Note: There are three more analog input on the circuit board, but there are no external connections available for them. To use them, a connector of some type would need to be added on the rear panel. The same limit applies their use, to use a voltage divider resistor pair to limit the maximum voltage at the circuit board to no more than 3.6V.

Jumpers

Functions/descriptions

Jumper Name Function.

J21	ANO
J20	AN1
NOR/HT	ISOLATION1200 EQ RECEIVE INPUT EQUALIZATIONINPUT Z RECEIVE INPUT IMPEDANCEVIN/RESET DSUB-9 PIN7 OPTION
RESET/GND	DSUB PIN9 OPTION
WATCHDOG	PTT WATCHDOG TIMER
TEST/RESET	SELF-TEST/HARD RESET
LOAD	BIOS LOAD MODE

J20: AN0 (default closed) When closed, allows AN0 to be sampled through pin 8 on the DSUB-9 jack.

J21: AN1 (default closed) When closed, allows AN1 to be sampled through pin 4 on the DSUB-9 jack.

Input Z: (default open) This jumper sets the impedance load connected the receive audio output of the receiver. When closed, sets receive input impedance to 600 Ω . When open, input impedance is set to 10 k Ω .

1200 EQ: (default closed) This jumper sets equalization of the receive audio input. When closed, no equalization is selected, and when open, partial equalization is used. See the calibration/equalization section for details.

GND/RESET (default center pin to GND) This three-pin jumper connects to DSUB-9 pin. When placed on the center pin and GND pin, pin 9 of the DSUB-9 connector is connected to ground. When placed on the center pin and the RESET pin, pin 9 is connected to the reset circuit. With this setting, a soft reset happens by applying a momentary ground to pin 9.

VIN/RESET (default OPEN) This three-pin jumper either supply dc power to the unit or to perform a soft reset without powering down the unit. See the section on Connector Configuration for more details. NOR/HT (default NOR) When placed on the center pin and the pin marked NOR, this configures the AFSK output line for

use with base station radios. When placed on the center pin and the pin marked HT, this configures the isolation circuit, where the AFSK output line and PTT line are combined for use with for HT radios.

Watchdog (default OPEN) This jumper enables/disables the PTT watchdog timer. When placed on one pin, the watchdog timer is active; when placed on both pins, the timer is disabled.

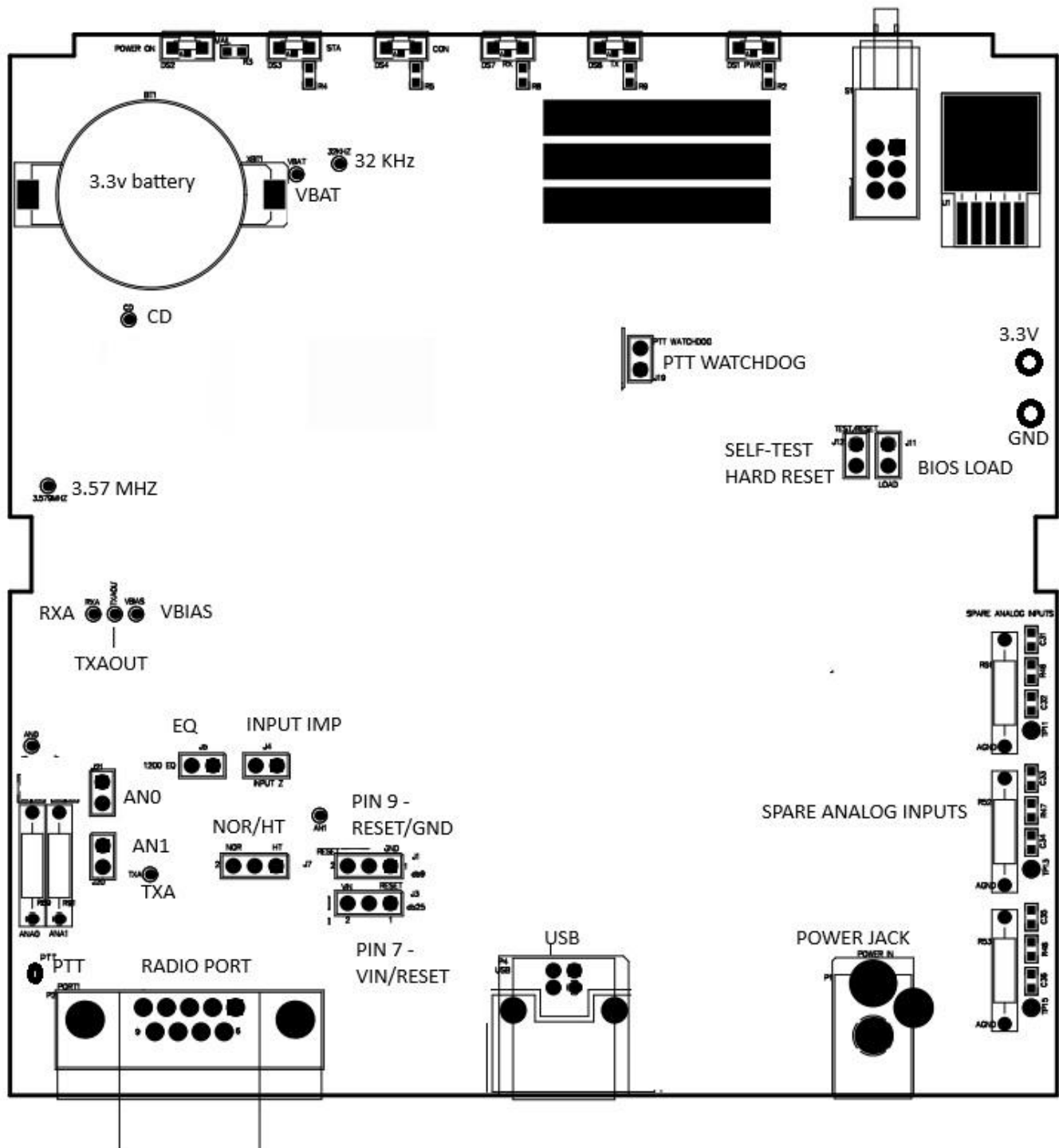
TEST/RESET (default OPEN) When open no action occurs. When closed (after turning unit power off), the processor will perform a “hard” reset (and internal self-test) once power is turned back on. All command parameters will be set to factory default values.

LOAD (default OPEN) When open, no action occurs. When closed (after turning unit power off), when next powered ON, the unit will be ready to accept a replacement bios file to be loaded into flash memory. This function is provided to easily update the unit.

Test Points

32KHz	-real-time clock crystal frequency
VBAT	backup battery voltage
CD	rx carrier detect
3.3v	regulator output voltage
3.57MHz	modem reference crystal frequency
RXA	receive audio
TXAOUT	transmit audio output from modem IC
VBIAS	bias voltage from modem IC

Board Layout (jumpers/test points)



Appendix C: Options

Low Power Operation

The KPC-3XE is efficient, normally drawing less than 40mA.

lower power consumption is possible.

The front panel LEDs can be turned off, by setting the command LEDS OFF.

Use INTERNAL carrier detection. Set the command CD to INTERNAL. This will require using squelched receiver audio from the radio. When SOFTWARE carrier detect is used, the TNC continuously monitors the received audio to look for packet data (and uses a bit more processing power).

Replacing the Lithium Battery

A lithium battery cell inside is used to maintain settings changes and other data, and to power the real-time clock, when the TNC is powered off.

To replace that battery, power off the TNC, remove the top cover, and carefully lift the battery out of the battery holder. Use an insulated (or non-metallic) tool to assist in removal of the battery. When installing a replacement battery (CR2032 or equivalent), be sure its positive terminal is facing up. Just press it into the battery holder.

NOTE: When the lithium battery is removed, all stored parameters will be lost, and the unit will revert to the factory default parameters when next powered on.

Appendix D: In Case of Difficulty

The KPC-3XE is manufactured to very stringent quality standards. If the installation procedures outlined in this manual have been followed, it is highly unlikely that a failure will be encountered. When having difficulty, use the procedures described in this section to assist in isolating and correcting the problem.

Does Not “Sign-On” to Computer

1. Carefully recheck cabling between the computers USB port and the TNC.
2. Check the terminal program to be certain it is configured with the correct communications parameters (com port number, baud, parity).
3. Try a “Hard Reset” using the Hard Reset jumper. (Operate the terminal program at 9600 baud when performing a hard reset to see the self-test result message output.)

Unable to Make a “Connect”

1. Issue a connect request and observe the TNC’s XMIT LED. If the XMIT LED illuminates, check to ensure that the TNC to radio interface cable is connected securely to both the TNC and to the radio port. If the XMIT LED does not illuminate, check to be sure the XMITOK command is turned ON. If the connection to the radio uses a fixed level un-squelched receive audio output, set the TNC command CD to SOFTWARE.
2. Observe the radio to determine if it is being switched to the “Transmit” condition when the TNC indicates it is transmitting. If not, recheck wiring of the interface cable to be sure PTT and ground are connected correctly.
3. If possible, monitor transmitted packet signals with another radio. If the transmitter is keying to “Transmit” but weak or no audio is monitored, increase TNCs AFSK output as necessary using the XMITLVL command. (See the AFSK Output Level section.)

Cannot Transmit

1. Check the XMITOK command – it must be ON to allow transmitting.

2. If using a “fixed level” (i.e., the RCV LED stays on all the time) Receive Audio from the radio, which the volume and squelch do not change, set the TNCs CD parameter to SOFTWARE. The TNC will not transmit while its receive led is lighted.
3. Check the interface cable, if it has a wire connected to the TNCs XCD pin, and the radio has asserted that line low (to ground level), the TNCs transmit will be inhibited.

Cannot Return to Command Mode

The single most common cause of this is that the STOP character (and usually XOFF) have been inadvertently set to the same as the COMMAND character. This is usually caused by using the dollar sign (\$) as a streamswitch. When using the \$ as a streamswitch character, be aware that HEX values cannot be entered without PASSing the dollar sign. Symptoms for this usually are that characters can be sent to the TNC XE fine in Command Mode, and can be sent to others on the air, but just can't get the TNC back to Command Mode. With most PC terminal programs, pressing <Ctrl+C> will display the heart character, but the cmd: prompt won't be returned.

Appendix E: Additional Information

Specifications

Size: 0.8" x 5.2" x 5.2" (21 mm x 133 mm x 133 mm)

Weight: 11 oz (0.32 kg)

Power Requirements:

6 V dc to 25 V dc, less than 30 to 50 mA (LEDS on, unit active)

6 V dc to 25 V dc, about 20 to 30 mA (LEDS off, unit inactive)

Power Plug Polarity: Center pin positive

Watch Dog Timer: Approximately 2-1/2 minutes

External Carrier Detect (XCD): Pulldown to ground

External Reset: Pulldown to ground

PTT Output: Open drain, +50 V dc max, 200 mA max

Audio Output:

Continuously adjustable: 1 mV p-p to 2 V p-p

Output Impedance (ac coupled): 600 Ω

Modulation:

1200 bps FSK, full duplex

Bell 202 (1200 Hz/2200 Hz)

Audio Input:

Minimum Input Sensitivity: 5 mV p-p

Dynamic Range: 70 dB

Input Impedance (unbalanced): 10 k Ω (600 Ω with Jumper input Z connected)

Max Input Voltage: ± 12 V dc; 35 V p-p sinusoidal

Modes of Operation Packet (AX.25/FX.25), KISS, XKISS, Host, GPS, LINESUB, MODEM, NET, POLL, TUP

Other Features: PBBS, KA-NODE, Remote Access, and, K-Net

Messages from the TNC

The following are brief explanations of the most important messages that the TNC may send to the terminal.

*****(callsign) busy**

The other packet station to which an attempt to connect (callsign) has been made, is unable to accept connects and responded with a busy signal.

Already connected on stream n

An attempt has been made to connect to someone a station that is already connected on another stream. The STATUS command will show all active connections, and on what stream (i.e., stream "n").

BBS BUSY

An attempt has been made to connect to the internal PBBS, but the PBBS does not have any available streams (PBUSERS) to allow a connect.

CALIBRATE MODE HBAUD 1200:

M send mark

R receive

S send space

T send square wave

- or + adjust XMITLVL while transmitting

X exit

XMITLVL 100

This list of Calibrate Mode functions, allow commands to be entered; M to generate a MARK tone, R to receive a Kantronics Calibrate signal, S to generate a SPACE tone, T to transmit a Kantronics Calibrate signal, - or + to adjust XMITLVL while transmitting, or X to exit calibrate mode and return to the TNCs Command Mode.

Can't DISCONNECT

The currently selected stream is not in connected state so therefore cannot disconnect. This message will be followed by the current stream letter and a "Link state is:" message, described later in this section.

Can't RECONNECT

An attempt has been made to reconnect to a station (by issuing a CONNECT command) but the callsign entered is not the same as the

station already connected. Reconnect can only be made to the station already connected to on this stream.

CHECKSUM ERROR

This message indicates that the bios/firmware in this TNC may be damaged. This message may be seen when performing a hard reset, or any time a soft reset is performed (including initial power up), and a bios error is detected.

CHECKSUM OK

When performing a hard reset, this message indicates that the bios/firmware has passed the internal checksum test.

cmd:

This is the displayed prompt when the TNC is in its Command Mode. Any characters entered after the TNC prints "cmd:" will be used as command input and not packet data.

Command not available in NEWUSER mode

This message indicates that an attempt has been made to change one of the commands in the TNC, but that command is not currently available. This occurs when the INTERFACE command is set to NEWUSER. NEWUSER mode is a limited command set. To enable access to all commands, set INTERFACE to TERMINAL.

***** connect request:**

A remote packet station has attempted to connect, but there is not a valid stream available for the connection. The remote station will be sent a busy message packet. See the USERS and MAXUSERS commands for setting more streams and allowing more connects at one time if desired. Also be sure CONOK is ON.

***** CONNECTED to call [VIA digi1....digi8]**

A packet connection has taken place. This can happen after issuing a connect request or an accepted connect request coming in from a remote station. "call" will be the callsign entered in the remote stations MYCALL and if a path was used it will be shown.

***** DISCONNECTED**

The packet connection no longer exists.

EH?

This is the TNC's generalized "I don't understand" message. A dollar sign (\$) is used to point to the offending character in the previous entry. It will also appear if a required input item is missing, e.g.:

```
C KV7B V
          $
EH?
```

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

ENTER YOUR CALLSIGN=>

This message is displayed after an * has been sent during the autobaud routine, Type in a callsign. The TNC will use this callsign to properly implement its protocol. The callsign entered will be placed in the MYCALL parameter and used to generate the MYNODE and MYPBBS callsigns, and the CWIDTEXT.

xxxx FREE BYTES

This message indicates how many additional characters may be entered into the TNC packet buffer before the buffer is full.

*****FRMR received:**

A frame reject packet has been received for an unknown reason. The information field of this packet will display in hexadecimal value. This display may be useful in determining why the connected receiving station rejected a packet.

*****FRMR sent:**

A frame reject packet has been sent due to a detected error in protocol. Three bytes (6 hexadecimal characters) are displayed to assist in determining the reason for the reject.

Note: FRMR errors may indicate a callsign conflict. This can mean that more than one TNC (or TNC function) is set to use the exact same callsign.

Input ignored

Since the TNC command interpreter was kept small and simple, sometimes it will change parameters before it completes parsing some of the more involved commands. In some cases, options at the beginning of the command will have been acted on before a syntax error near the end of the line is reached. When this occurs, "Input ignored" is used to show what

part of the line was ignored. The dollar sign points to the boundary: characters to the left were used; the character pointed to and those to the right were not, i.e., the line was parsed as if a <CR> was entered at the \$. For example:

```
BUDLIST QST,KBØNYK NØKN
                        $
Input ignored
```

Because the comma is missing, the command is interpreted as if it were BUDLIST QST,KBØNYK; the NØKN is ignored.

INVALID STREAM

An attempt has made to change to a stream that is not valid. Stream designators must be letters A - Z. MAXUSERS sets the upper limit on valid streams.

**KANTRONICS PACKET COMMUNICATOR KPC3XE VERSION X,XXXXX
(C) COPYRIGHT 2025 BY KANTRONICS INC. ALL RIGHTS RESERVED.
DUPLICATION PROHIBITED WITHOUT PERMISSION OF KANTRONICS.**

A message such as this (the currently installed bios version will be shown) appears when the TNC is first turned on and after any soft reset, or when changing the setting of commands the set memory allocation, such as; NUMNODES, PBBS, MAXUSERS or MYREMOTE commands, or after issuing the RESET command.

Link state is:

This message is output in response to the CONNECT and DISCONNECT commands if the state of the link does not permit the requested action. It is prefaced by "Can't CONNECT" or "Can't DISCONNECT" as appropriate and will be followed by the current link state. A CONNECT command with no options will display the link state of the current stream.

Current link states are:

Both devices busy

Both TNCs involved in the connection are unable to accept any more data.

CONNECTED to (callsign v path)

This TNC is currently in connected state to the indicated station, using the path given.

CONNECT in progress

This TNC is attempting to establish a connection.

Device busy

This TNC is unable to accept any more data from the remote station at this time.

DISC in progress

This TNC is attempting to disconnect from another station.

DISCONNECTED

No connection exists on the current stream.

FRMR in progress

This TNC has detected an error in the protocol. This is normally caused by two TNCs using the same callsign, resulting in both of them trying to respond to the same message (causing a packet sequence error).

Remote device busy

The remote TNC is unable to receive any more data from the radio at this time.

Waiting ACK and device busy

This TNC has sent a packet to another station and is waiting for the acknowledgment, but it is not able to accept any data from the radio at this time.

Waiting ACK and remote busy

This TNC has sent a packet to another station and is waiting for the acknowledgment, and the remote TNC is not able to accept any data from the radio at this time.

Waiting ACK and both devices busy

This TNC has sent a packet to another station and is waiting for the acknowledgment, but neither this TNC nor the other station's TNC is able to accept any data from the radio at this time.

Waiting acknowledgment

A packet of data has been to another station, and TNC is waiting for the acknowledgment.

MESSAGES WOULD BE LOST

This message indicates that an attempt has been made to change the PBBS size too small to hold all of the existing messages in the mailbox. The size is not changed, and therefore no messages are lost. To make the mailbox smaller, some messages must first be deleted, or set the PBBS to 0, which will erase all messages, then re-enter a new PBBS size.

NO KNOWN NODES

An NDHEARD list has been requested and the TNC has not heard any other nodes (e.g., KA-Nodes).

NOT ENOUGH RAM

This message indicates an attempt has been made to set the NUMNODES, MAXUSERS, or PBBS to a value, which requires more RAM memory than is currently available. It may also occur when trying to enable the MYREMOTE and there isn't enough memory left.

Not while connected

Some parameters cannot be changed if the TNC is in connected state to another station. This message is printed if such an attempt is made.

**PBBS MESSAGE BUFFER NOT VALID!
TRYING TO RECOVER**

When powering on the TNC, a check is made of the pointers relating to the PBBS. It appears that one or more of these pointers may not be correct. An attempt is made to correct them, but it is possible that some or all messages may be lost.

PRESS (*) TO SET BAUD

The TNC needs an asterisk (*) character from the terminal within 2 s of seeing this message. The autobaud routine will then detect what baud the terminal is using and will set its own interface baud to match.

RAM OK

This message appears when a hard reset is performed and indicates that the TNC has tested its SRAM and has found no problems.

RAM ERROR

If this message appears when performing a hard reset, the TNC has detected a problem with its SRAM.

*****retry count exceeded**

***** DISCONNECTED**

The number of tries set by the RETRY command has been exceeded.
Therefore, the connection has been broken.

S00

This message from the TNC indicates that it is in the HOST interface mode and has just performed a soft reset has happened. To exit Host interface mode, the TNC must receive the HOST escape characters from the terminal. See the Host mode section of the manual for details. Two FEND characters precede this message and one FEND character follows this message, but they may not be visible (depending on the terminal program).

Value out of range

If the syntax of an entered command is legal, but the value specified is too large or too small for this command, this message is returned. A \$ is used to point to the bad value.

Was

Whenever one of the parameters is changed, the previous value is usually displayed.

Example:

```
cmd:AX25 OFF
AX25L2V2 was ON
```

YOU HAVE MAIL

A new message addressed to the TNCs MYCALL has just been stored in the PBBS.

ASCII Chart

Ctrl	Dec	Hex	Code	Dec	Hex	Code	Dec	Hex	Code	Dec	Hex	Code
@	0	00	NUL	32	20	SP	64	40	@	96	60	`
A	1	01	SOH	33	21	!	65	41	A	97	61	a
B	2	02	STX	34	22	"	66	42	B	98	62	b
C	3	03	ETX	35	23	#	67	43	C	99	63	c
D	4	04	EOT	36	24	\$	68	44	D	100	64	d
E	5	05	ENQ	37	25	%	69	45	E	101	65	e
F	6	06	ACK	38	26	&	70	46	F	102	66	f
G	7	07	BEL	39	27	'	71	47	G	103	67	g
H	8	08	BS	40	28	(	72	48	H	104	68	h
I	9	09	HT	41	29	)	73	49	I	105	69	i
J	10	0A	LF	42	2A	*	74	4A	J	106	6A	j
K	11	0B	VT	43	2B	+	75	4B	K	107	6B	k
L	12	0C	FF	44	2C	,	76	4C	L	108	6C	l
M	13	0D	CR	45	2D	.	77	4D	M	109	6D	m
N	14	0E	SO	46	2E	:	78	4E	N	110	6E	n
O	15	0F	SI	47	2F	/	79	4F	O	111	6F	o
P	16	10	DLE	48	30	0	80	50	P	112	70	p
Q	17	11	DC1	49	31	1	81	51	Q	113	71	q
R	18	12	DC2	50	32	2	82	52	R	114	72	r
S	19	13	DC3	51	33	3	83	53	S	115	73	s
T	20	14	DC4	52	34	4	84	54	T	116	74	t
U	21	15	NAK	53	35	5	85	55	U	117	75	u
V	22	16	SYN	54	36	6	86	56	V	118	76	v
W	23	17	ETB	55	37	7	87	57	W	119	77	w
X	24	18	CAN	56	38	8	88	58	X	120	78	x
Y	25	19	EM	57	39	9	89	59	Y	121	79	y
Z	26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
[	27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
\	28	1C	FS	60	3C	<	92	5C	\	124	7C	
]	29	1D	GS	61	3D	=	93	5D	]	125	7D	}
^	30	1E	RS	62	3E	>	94	5E	^	126	7E	~
_	31	1F	US	63	3F	?	95	5F	_	127	7F	DEL