MULTI--USE DESIGN CONSIDERATIONS FOR THE TAPR TNC

Harold E. Price, NK6K
2110 Farrell Ave #14
Redondo Beach, CA 90278

Abstract

The Amateur Packet Radio Terminal Node Controller (TNC) built by TAPR Inc. is designed to meet the needs of a wide range of users, from the technical experimenter to the "appliance operator" end user. This paper discusses the human engineering factors which went into the design of the TNC's external software interfaces that enable it to serve a heterogeneous user base.

Background

The first Terminal Node Controller widely distributed for amateur use was produced by the Vancouver Area Digital Communications Group, It was mainly distributed in bare-board form, and aside from the specialized HDLC communications chip, was able to be stuffed with readily available components. Once the user supplied a modem and power supply he had a device which required only a terminal and a radio to get on the air and send packets. The capability supplied by the VADCG TNC to newcomers in the field of packet radio far surpassed anything else available at the time, and has earned it its place in the Amateur Radio Hall of Fame.

The TAPR TNC was designed to go beyond the capabilities offered by the previous TNCs. Taking advantage of advances in technology, including the reduction in price of denser memories, the TNC hardware offered the following features:

- 24K ROM
- Non-volatile RAM
- 6K RAM
- Onboard power supply with an off board transformer included.
- Onboard modem
- The TNC would be offered assembled and tested.

The increased program space made possible increased software functionality. The increased level of pre-packaging would make the board attractive to an additional class of users, the "appliance operators". This term applies to that group of people who buy a product and expect to plug it in and use it. The term is sometimes used in a derogatory sense but that is not the intent here. This class of TNC user is not interested in the underlying concepts and does not wish to tinker with the innards of the TNC. They accept packet radio as a proven technology and want to get on with the business of being end users, setting up higher level networks, establishing mailbox systems, building local area nets with intelligent host nodes and file servers, or simply using the TNC as error free RTTY. The TMC is a secondary item, simply a means to an end.

This type of user would not be happy if he was required to modify source code and recompile to change node address, call signs, and other operational parameters, On the other end of the spectrum is the experimenter, those individuals interested in tinkering with the lower levels. Many areas of amateur packet radio remain open issues, What are the best timing algorithms? What are the correct retry counts, timeout values, keyup delays, packet lengths and other details associated with the delivery of HDLC frames from one TNC to another? The functions offered by the TAPR TNC to s0lve these basic compatibility problems are discussed in the following sections.

Conflicting Goals

To illustrate the types of problems involved, the following items are presented, each with its set of conflicting viewpoints.

Degree of Transparency

In applications where the TNC is treated as a means for connecting two devices, computer to computer or remote terminal to computer, the TNC should be totally invisible. It can be viewed as a simple modem, receiving characters from data terminal equipment (DTE) and sending it down a phone line, and receiving data from the line and sending it to the DTE. The TNC has no personality of its own,

In other applications, the TNC is itself the smart device at the end of a communications link, using a terminal only as an IO device under its control. The TNC performs local input editing and output formatting. It attempts to preserve visual fidelity on the terminal device, i.e., not mixing the echo of data being entered for transmission with data received from the link,

Command set

Infrequent (casual) users require a simple command set with straightforward syntax and easy-to-remember english-like commands. Changing TNC major operating modes should require a small number (one) of commands.

Expert users or those experimenting with operational parameters require a rich command set. This class of user also prefers commands needing only a small number of keystrokes as the command entry rate is higher.

Tuning Knobs

This is closely related to command set, in that the more things there are to adjust, the more commands will be required to adjust them.

To use a radio analogy, some hams are happy with ON/OFF, push to talk, and frequency tuning, others require variable bandwidth, IF shift (both high side and low side), notch, RF attenuation, multiple tuning rates, ten VFOs and snooze alarm.

In addition, experimenters want access to all parameters which control forming, sending, and receiving packets.

These are examples of the type of conflicting interests generated by a diverse user base. Availability of a 24K program space on the TNC coupled with the means to develop software in a high level language (Pascal) permitted the three-person software team to provide a single integrated software package which meets everyone's needs. The major features of this design are discussed below.

Definition of Terms

Several of the shorthand terms used in the remainder of this paper are defined here.

User interface. The asynchronous data path to the TNC, the path used to exchange data which is formatted and transmitted over the air. Configuration commands are also given to the TNC through this path.

Link interface. The bit stream (HDLC) path which connects directly to the RF transceiver. Rackets (frames) are sent over this path.

Connected. A TNC is "connected" when it has exchanged the proper sequence of packets over the link interface with another TNC such that the two TNCs are connected at the AX.25 (or VADCG) level 2 layer. Data sent by the TNC will be in the form of sequenced (numbered) information frames. Connected does not refer to any physical connections.

Unconnected. The TNC is not "connected" to any other TNC. Data sent by it on the link is in the form of unsequenced information frames.

Data transfer mode. Characters received through the user interface are assumed to be data for eventual transmission through the link when the TNC is in a data transfer mode. When not in this mode data input is assumed to be a configuration command.

Operating modes

The major point of difference between users of the TNC is the degree of transparency of the user interface when the TNC is in a data transfer mode. Put more sirnply, is the TNC something one speaks TO or speaks THROUGH? As stated previously, sorne users want to treat the TNC as a smart terminal, others view it as a smart modem, to be heard but not seen, Because of the large number of differences in basic TNC functions required to satisfy these two needs, the TNC software supplies two data transfer modes. Entering one 0f these modes has the effect of changing the values of a large number Of configuration parameters at once. The data transfer modes are the CONVERSATIONAL mode and the TRANSPARENT mode, and are abbreviated to "convers" and "trans" .

Changing modes causes a radical change in behavior of the interface. Attributes of the TNC in CONVERS mode are:

1) Characters input are scanned for special meaning to implement f'eatures discussed below.

2) Creation of packets is directly under user control. A special character is used to say "take the data entered previously, make it into a packet, and send it as soon as possible".

3) Data can be edited until released to be made into a packet. Characters can be struck out, a line erased, or the entire packet can be erased.

4) Data input is echoed locally, i.e., by the TNC.

5) Data received from the link is formatted before being output.

6) Flow control is available via use of XON,'XOFF characters. Flow control is the process where data buffer or CRT display overflow is avoided by stopping the flow of data, The XOFF character is a request to halt the flow of data, XON is permission to resume t'he flow through the user interface. Case translation is performed, line feeds can be inserted in the data stream following carriage returns, carriage returns are inserted when the output line length is exceeded.

The attributes of the TNC when in the TRANS mode are:

1) All characters are transferred to the link without modification.

2) Creation of packets is only under the indirect control of the user. Packets are formed by time or by data length. Data length control causes packets to be formed after "n" characters have been entered through the user interface. Time control will build a packet every "n" seconds or after "n" seconds have passed since the last character was received from the user interface.

3) Local echoing and local editing are not available.

4) Data received through the link is transferred to the user interface as received, without modification or addition of control characters.

5) Flow control is not controlled by data characters, but is managed by use of the standard RS-232 handshake lines.

These two data transfer modes allow the TNC to be compatible with a wide range of applications. The behaviors described are present by default, they may be modified at will by the user. Functions can be added to TRANS mode, or removed from CONVERS mode, supplying a variety of hybrid semi-transparent or translucent modes.

A third operating mode is present in the TNC, COMMAND mode. Command mode is active when the TNC is first powered up, and can be entered from either of the other modes. COMMAND mode is used to change the configuration and operating parameters of the TNC. Changing the TNC from mode to mode does not effect the state of the link, i.e., whether or not the TNC is connected. This allows two personal computer owners to talk back and forth in CONVERS mode, and then enter TRANS mode to let their computers transfer files.

A problem arises with respect to the total transparent mode. Short of a hardware reset, how is the command mode entered from the transparent mode? The method used to leave convers mode and enter command mode, a special control character, can not be used. Instead, a small compromise is made. After a user defined period (guard time) has passed where no characters have been received from the user interface the TNC will watch for a mode escape character. If this character is entered three times with no intervening characters, and another guard interval passes with no additional input, the TNC enters the command mode. A proper selection of the guard time and escape character will provide a high probability that reception of the sequence is a valid request for command mode entry. This procedure did not originate with the TAPR TNC and is a common practice for intelligent modems.

<u>Parameters</u>

There are many facets of the user and link interfaces that experimenter will want to twiddle with and the end user will want to ignore, or at most change only once. Examples of user interface items are:

- Editing characters such as character delete, line delete, and packet delete.
- Flow control characters.
- Packet creation times in TRANS mode.
- Output presentation control such as <lf> after <cr>, number of nulls after <cr>, and terminal line length.

Examples of link control items are:

- Transmitter and repeater keyup delays.
- Number of times to retry a frame, frame acknowledge time
- Packet data size
- Maximum contiguous frames sent or maximum number of outstanding unacknowledged frames
- Station address or call sign.

Previous amateur packet systems kept these parameters in ROM and supplied no way to modify them at run-time. The TAPR TNC software embodies a design methodology where all such values are kept in RAM while the TNC is running, and therefore can be modified by user command. The values are initialized from ROM when the TNC is first installed. The large number of parameters available to the user (see table 1) would require a very tedious startup procedure every time power was applied as not all users would be happy with the default values supplied in ROM. To avoid this unpleasantness, the TAPR TNC uses non-volatile RAM to store parameters when the TNC is turned off. A dip switch on the TNC selects either the ROM copy or the non-volatile RAM copy for initialization of the RAM parameters, After an initial boot from ROM, the TNC is then configured to use the non-volatile RAM for subsequent booting.

This combination of software and hardware design permits hands-off operation for some users, one-time configuration for end-users, and endless opportunity for change by experimenters.

Commands

The desire to have many user accessible parameters conflicts with the desire to have a small number of simple commands. Since a major design goal was to externalize as many operational parameters as possible, it was soon realized that the TNC would have several commands. The Beta test release of the TNC software has 66 separate commands.

Only four commands, the commands to connect and disconnect the link and the mode change commands, are used in normal operation of the TNC. Most of the other commands are used to change parameters which have default values pre-assigned when the TNC is first powered on. These values are satisfactory for most applications and will never be changed. If the parameters are changed, however, the user can issue a command that causes the changed values to be stored in the non-volatile RAM, replacing the default values.

The command syntax is kept simple, consisting only of the name of the parameter to be changed and the new value. A DISPLAY command is present which lists all parameters and their values. While not a replacement for a HELP facility, the list of parameters names serves as a memory jogger and will cut down on trips through the manual.

To reduce the number of keystrokes needed, all commands can be abbreviated to a smaller number of characters that still uniquely identify the parameter,

Documentation

As much care was taken with the design and implementation of the manual as was taken with the design and implementation of the hardware and software. The documentation must serve the same diverse user base the that TNC does. Because of the need to provide both tutorial and detailed information, the manual for the TNC is approximately 140 pages long. The manual is structured to supply the right level of detail at the right time. The first few pages deal directly with the tasks required to get the TNC on the air, including a detailed radio interface section. This detail is required because incorrect wiring of the TNC to RF interface can result in damage to the TNC and/or the RF gear. The introductory material is keep as short as possible however, and by page 28 the user will be on the air sending packets.

Subsequent sections of the manual provide tutorial information on the hardware and on the protocols used. Other sections provide detailed information on the software command set and on the TNC hardware. A full set of schematics is provided for hardware experimenters. Also included in appendicies are the specifications for the communications protocols used on the TNC. These supply sufficient detail for others to implement compatible software.

The manual is structured to supply the level of detail required by experimenters while at the same time not scaring off the casual user.

Summary

Great care was taken during the design and implementat ion of the TNC so-ftware to provide a simple interface for end users while not limiting the activities of more advanced users. Th e TAPR TNC is currently in a Beta Test phase with 180 users spread out in more than 16 local groups. It is expected that new commands will be needed while some current commands will prove to be superfluous and can be removed. The large percentage of high level code and human engineered design should make this a relatively painless task.

Acknowledgements

Many thanks go to Margret Morrison, KV7D, and Dave Henderson, KD4NL, who along with the author designed and implemented the TAPR AX.25 software, and to Dan Morrison, KV7B, for his design inputs.

Command	Type	Function
ABAUD	U	Asynchronous 'baud rate
ABIT	U	Asynchronous stop bits
AUTOLF	U	Follow received <cr> with <lf>
AWLEN	U	Asynchronous word length
AX25	L	AX.25 or VADCG protocol
AXDELAY	L	Keyup delay time for audio repeater
AXHANG	L	Audio repeater dropout (hang) time
BEACON	L	Beacon mode
BKONDEL	U	Echo <bs> when <del> entered
BTEXT	L	Beacon text
CALIBRAT	C	Enter hardware calibration routine
CANLINE	U	Cancel line character
CANPAC	U	Cancel packet character
CMDTIME	U	Command mode entry from TRANS mode timer
COMMAND	U	Command entry from CONVERS mode character
CONMODE	D	Default data transfer mode
CONNECT	L	Connect TNC to another TNC
CONOK	L	Permits connections in unattended operation
CONVERS	D	Enter CONVERSation mode
CPACTIME	U	Create packets by time in CONVERS mode
CR	U	Append a <cr> to the end of each packet in CONVERS mode
DEBUG	C	Enter the debugger
DELETE	U	Specifies which of <del> or <bs> deletes characters
DIGIPEAT	L	Enables AX.25 digipeating function
DISC	L	Disconnects the TNC from the link
DISPLAY	C	Displays all configuration parameters
DWAIT	L	Digipeater wait time
ECHO	U	Local echo in not in transparent mode
ESCAPE	U	Character echoed when <esc> is entered
FLOW	U	I/O data flow in CONVERS mode
FRACK	L	Frame acknowledge timeout
HBAUD	L	HDLC (link) baud rate
ID	L	Send CW id
LCOK	U	Lower case translation
MAXFRAME	L	Maximum unacknowledged frames
MONALL	L	Link monitor command
MONCON	L	Link monitor command
MONFROM	L	Link monitor command
MONITOR	L	Link monitor command
MONT0	L	Link monitor command
MYCALL	L	Call sign (address) of this TNC in AX.25 mode
MYVADR	L	VADCG protocol address byte
NULLS	U	Number of nulls added after <cr> in CONVERS mode
PACLEN	L	Maximum packet data length
PACTIME	U	Packet creation time for TRANS mode
PARITY	U	Parity of asynchronous user interface
PASS	U	Send next character verbatim in CONVERS mode
PERM	C	Put current parameter setting in non-volatile RAM.
RESET	C	Software reset
RETRY	L	Frame retry count
SCREENL	U	Screen width for CONVERS auto-wrap feature
SENDPAC	U	Create packet character for CONVERS mode
START	U	Flow control character
STOP	U	Flow control character
TRACE	L	Link diagnostic command
TRANS	D	Enter TRANSparent mode
TXDELAY	L	Transmitter keyup delay
TXFLOW	U	Use character flow control in TRANS mode
UNPROTO	L	Unconnected address field
VDIGIPEA	L	Enables VADCG digipeater function
VRPT	L	Direct VADCG frames sen by this TNC to a digipeater
XFLOW	U	XON/OFF or hardware flow control in CONVERS mode
XMITOK	L	Enable transmit functions
XOFF	U	Flow control character
XON	U	Flow control character

L - Link commands   U - User interface
C - TNC control   D - Data transfer mode

Table 1. TNC commands