

# E-Z PACKET 80

A Packet Terminal Program

For The TRS-80 Model I/III/4/4P

Version 1.1

Copyright (c) 1986 by The Martin Company

## COPYRIGHT NOTICE

Copyright (C) 1986 by The Martin Company. All rights reserved worldwide. This publication, in whole, or in part, may not be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of The Martin Company, P. O. Box 982 Marysville, Washington 98270 USA.

## DISCLAIMER

The Martin Company makes no representations or warranties with respect to the content hereof, and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, The Martin Company reserves the right to revise this publication and software and to make changes from time to time in the content of the manual and software without obligation of The Martin Company to notify any person of such revision or changes.

## DISTRIBUTION DECLARATION

\*\*\* USER NOTICE \*\*\* If you have received this program from another user and find it of value, your \$20 contribution will be appreciated. In return, an operation manual will be forwarded to you.

The Martin Company  
P. O. Box 982  
Marysville, WA 98270

Notice: Users of this program are granted a limited license to make copies of this program for trial use by others on a private non-commercial basis. This limited license does not include:

- (1) Distributing this program in connection with any other product
- (2) Making the program available for any consideration or "disk fee"
- (3) Posting the program for public access via telecommunications
- (4) Distributing the program in modified form

We encourage you to pass along this program to your friends. In doing so, please encourage them to contribute \$20 for the program. The manual is copyrighted and you are not authorized to copy and distribute the manual. We ask you to please cooperate and not let your "friends" copy the manual. We feel the program and manual are well worth the \$20 contribution. Your cooperation will be greatly appreciated.

Where used in this manual, the following are recognized as registered trademarks:

TRS-80 is a registered trademark of Tandy Corp.  
TRSDOS is a registered trademark of Tandy Corp.

\* \* \* Table of Contents \* \* \*

| Chapter |   | page |
|---------|---|------|
| I       | Introduction . . . . .                      | 1    |
|         | General Description . . . . .               | 1    |
|         | Hardware. . . . .                           | 2    |
|         | Software. . . . .                           | 3    |
|         | Bootng the Program . . . . .                | 4    |
|         | Exploring Around. . . . .                   | 5    |
| II      | Menu Screens . . . . .                      | 7    |
|         | Command Screens . . . . .                   | 7    |
|         | Edit Mode Buffer Selection Screen . . . . . | 13   |
|         | Edit Mode Command Screen. . . . .           | 15   |
|         | Control Code Screen . . . . .               | 16   |
| III     | Odds and Ends. . . . .                      | 18   |



## =====

## GENERAL DESCRIPTION

=====

E-Z PACKET 80 is a highly sophisticated, split-screen, terminal program that was designed to allow easy operation of packet Terminal Node Controllers (TNC's) from a TRS-80 Model I, III, 4 or 4P, using a built-in RS-232 port, or a ROM-116 RTTY interface unit to connect to the TNC. Full hardware and software handshaking is supported. Particular care has been taken to ensure virtually trouble free operation in both the attended and unattended modes.

E-Z PACKET 80 contains 10 programmable 140 character line buffers, that can be sent to the TNC by a control key function, a 200 character programmable initialization text buffer (ITEXT) that is automatically sent to the TNC when the program is booted, a 200 character programmable exit text (ETEXT) buffer that is sent to the TNC when exiting the program to DOS, a 200 character programmable connect text (CTEXT) buffer that will automatically be issued upon connection to another station when you are in disksave, and a MAIN text buffer that is approximately 23K in length in the Model I, 22K in the Model III and 30K in the Model 4. All buffers are accessible, through a built-in editor, for modification.

In addition, the program allows saving of incoming text to disk or to the MAIN text buffer, sending of the MAIN text buffer to the TNC, saving of the MAIN text buffer to disk, loading any buffer from disk, and displaying the buffers or a disk file to screen.

Some other features of the program are the ability to customize the program and save it to disk, (including creating a version of the program that will automatically boot up into a disksave mode, unattended, in the event of power failures), automatic connect status checking, a remotely triggerable acknowledgement message, full file transfer support in the 8-bit mode and an automatic screen turn-off to prevent burning the CRT in.

=====  
HARDWARE  
=====

E-Z PACKET 80 makes use of the handshaking control lines between the computer and the Terminal Node Controller (TNC). This requires a five (5) wire connection between the two units. In addition to the normal TXD, RXD and GND connections, the CTS (from the TNC to the computer) and the RTS (from the computer to the TNC) must be connected. The cable should have only the above five wires connected. Do not use a cable where all the pins are connected. If you do, you will probably burn up a few resistors in the TAPR-2 and TAPR-2 clones.

If the CTS and RTS control lines are not connected, data may be lost when using certain features of this program, such as looking at the menu, displaying buffers or disk files, or during disk access of any type. When these functions are in use, the program causes the RTS line to become false, thus holding the TNC data back from the computer. In this way, data from the TNC is not lost while these functions are being used (unless the TNC buffer fills up). The CTS line from the TNC to the computer is used to hold computer data back from the TNC, when the TNC buffer becomes full (which can occur when dumping the **MAIN** text buffer into it using the SEND command). The program will also respond to XON and XOFF commands from the TNC.

When the TRS-80 Model I, III or 4 has a built-in RS-232 card, the connections are accomplished by the use of a straight through cable, which has only pins 2, 3, 4, 5 and 7 connected. For the ROM-116 version, some research may be necessary, due to the fact that several different pin out configurations were made. Consult your ROM-116 manual when making the hookup. When you are finished, you should have the above pins connected between the computer and the TNC and you should measure approximately -10 VDC on pins 2 and 3, with no data flowing in or out, while reading approximately +10 VDC on pins 4 and 5. Pin 7 is GND and should always read 0 VDC. To check for the proper connection of the RTS line, bring up the MENU (refer to the instructions in the next chapter) of the program, and while it is being displayed, check check pin 4 on the TNC. It should read approximately -10 VDC. When the menu is not on display, the voltage should be around +10 VDC. To check the CTS connection, load the **MAIN** text buffer (refer to the next chapter) with a greater amount of text than your TNC has RAM for. Turn your transmitter off or disconnect the keyline and use the SEND command to send the data to the TNC. Be sure the TNC is in the CONV mode so that it will send the text unproto. Sometime before the buffer is empty (watch the counter to see when it stops) the TNC buffer should become full and switch the CTS line from +10 VDC to -10 VDC. Try this text once with the XFLOW command ON and once again with it OFF. If the counter pauses in both modes, then the CTS control line is functioning properly.

## =====

## SOFTWARE

=====

E-Z PACKET 80 is designed to operate under TRSDOS for the Model I, III, 4 and 4P computers. There are no versions for cassette operation and compatibility with other disk operating systems cannot be guaranteed. In fact, using a different operating system will most likely cause some functions, such as the DIR command, to not function properly. We can offer no support, including program modifications, to anyone who is having problems due to the use of a different operating system.

E-Z PACKET 80 is designed to be used with the TAPR (Tucson Amateur Packet Radio) style of command format. TNC's currently known to be using this style are the TAPR TNC-1, Heathkit HD-4040, AEA PKT-1, AEA PK-80 and the Kantronics Packet Communicator.

Certain functions in this program will give automatic commands to the TNC. For proper operation, the following values should be programmed into your TNC and then PERMed:

|         |   |         |         |   |      |
|---------|---|---------|---------|---|------|
| AWLEN   | - | 8 *     | SCREENL | - | 0    |
| CMDTIME | - | 1       | SENDPAC | - | \$0D |
| COMMAND | - | \$03    | START   | - | \$11 |
| CONMODE | - | CONVERS | STOP    | - | \$13 |
| DELETE  | - | OFF     | TXFLOW  | - | OFF  |
| ECHO    | - | OFF     | XFLOW   | - | ON   |
| FLOW    | - | OFF     | XOFF    | - | \$13 |
| NULLS   | - | 0 **    | XON     | - | \$11 |
| PARITY  | - | 4 ***   |         |   |      |

\* NOTE: This command is valid only with TAPR1, TAPR2 and clones. Kantronics TNC's do not recognize this command. Do not set AWLEN to 7!

\*\* NOTE: Although the program will communicate with the TNC at baud rates greater than 1200 baud, we suggest that you not do so. If you wish to use a baud rate greater than 1200 baud, you may notice characters missing on the left end of the screen lines (when the CR occurs). To correct this situation, increase the nulls from 0 to a value that results in proper copy. (This requires having NUCR ON in the TNC.) At baud rates higher than 1200 baud, characters will be lost when the screen turns back on after automatically turning off. There is nothing that can be done about this and that is why we suggest 1200 baud as the maximum baud rate.

\*\*\* NOTE: For TAPR, Kantronics, etc, this value is set to 4. With the TAPR2, the value is 0. Consult your TNC manual and select the number that turns parity off.

Most other TNC commands affect the TNC to radio link and do not directly affect the operation of this program.

## =====

## BOOTING THE PROGRAM

=====

**NOTE:** In the following information, it is sometimes requested that the user perform a control function, such as **CTRL-H**. In the Model 4, this is done by holding down the **CTRL** key, and while holding it down, simultaneously pressing another key, which in the example would be the **H**. In the Model I or the Model III, which do not have a **CTRL** key, this function is performed by the **CLEAR** key. Thus, in the example given, you would hold down the **CLEAR** key, and while holding it, simultaneously press the **H** key.

When booting up an uninitialized version of **E-Z PACKET 80**, i.e. one that has not been previously initialized by someone and then saved via the **PGMSAVE** command, you will be presented with the split-screen format of the program. In the Model 4 version, you will see 18 lines of receive text area, containing a sign-on banner, a status line splitting the screen, and on the bottom, 5 lines of text area which is used for editing, command functions and sending to the TNC. You might think of these 5 lines as the "worktable" of the program. In the Model I and the Model III, the screen appears the same in format, but contains 12 lines of receive area, 1 status line, and 3 lines of worktable.

A question will appear in the bottom portion, or work area, of the screen asking what baudrate is to be used. This is the baud rate you will select for RS-232 connection between the computer and the TNC. It does not affect the baudrate of the radio link data.

Once this question is answered, the program will go into **CONV** mode with the buffer **COPY** mode **ON**. **CONV** mode will be annunciated at the right side of the status line in the mode window. As soon as this happens, any text contained in the **ITEXT** buffer will be issued to the TNC. Upon first initialization, this buffer will be empty, of course, but after you have saved an initialized version of the program, you will have no doubt placed in the buffer a startup sequence of your own preference. Examples you might use will be given in Chapter 2.

**COPY** is automatically entered whether you are booting an initialized version of the program or an uninitialized one. The reason for this is that the TNC may have been saving text in its buffer, and when the RS-232 connection is enabled by the program, all of the saved data will be issued to the computer. To allow you to view all of the data, even if it was of a sufficient quantity to have scrolled off the screen, the text is saved in the **MAIN** text buffer, where it can be viewed with the **DISPLAY** command. **COPY** mode is toggled on and off by typing a **CTRL-H**.



```
=====
EXPLORING AROUND
=====
```

So! You've booted the program, seen the screen, counted the lines to see if there really are that many, answered the question about the baud rate, perhaps even hit the **CTRL-H** and seen the **COPY** annunciation disappear from the status line, and there you sit!

Well, let's take this opportunity to do a little exploring of the program before someone walks in and starts asking you questions about what you have. And relax! This program is very simple to operate, requiring very few keystrokes to accomplish any task.

To start with, reach out and hit the **BREAK** key. The mode window now says **(\*COM\*)** and the worktable area says:

```
COMMAND?
>
```

You will also notice a window on the status line to the left of the mode window that says ( 0 ). This is the buffer counter, and it will display the character count for whatever buffer you are currently working in. Since the **MAIN** text buffer was the last one entered, upon initialization, and no text came from the **TNC**, it reads 0. (Of course, if something did come from the **TNC** while you were in **COPY** mode, then the counter will reflect this.) Here is an important fact to remember: no matter what mode you are in when using this program, the **BREAK** key will always return you to the **COMMAND** mode. To get from the **COMMAND** mode to the **CONV** mode, simply hit the **ENTER** key. Go ahead and try it. When you are in the **CONV** mode, anything you type will be sent to the **TNC**, which can cause some startling results if you are not careful. While in **CONV** mode, type a **CTRL-C** and hit **ENTER**. The **TNC** should return a message of "cmd:" to the upper screen.

OK, now hit the **BREAK** key again. This time, type in the command:

```
MENU
```

and hit **<ENTER>**. On the upper screen, you will see the first page of the menu. It contains the Distribution Declaration explaining what you can and cannot do regarding copying and sharing this program. (NOTE: The Distribution Declaration can also be found at the front of this manual.) Read it carefully. On the Model 4, it is all on one screen, while on the Model I and III, due to the limitation of the 64 character screen width, it takes two pages to get it all in. Once you have viewed this message, it is taken away, never to be seen again, unless you reboot the program. The reason for this is rather obvious. We wish to make sure that it is read by each and every user of this program at least once, regardless of where they acquired the program. Scrolling through the **MENU** screens is accomplished by use of the space bar, as is prompted for in the bottom screen.

Hit the space bar and read each screen as you come to it. You will find two or three **COMMAND** screens, depending on your computer model, one **EDIT MODE BUFFER SELECTION** screen, one **FULL EDIT MODE COMMANDS** screen and one **CONTROL CODES** screen. One last push of the space bar will return you to the **COMMAND** mode. Now that you are back there, once again type in:

```
MENU
```

and hit `<ENTER>`. You are now viewing the first `COMMAND` screen again.

Look carefully at the format of the screens as you scroll through them with the space bar. Each command shows the keystrokes necessary for its accomplishment, and is followed by a short description to serve as a little "mental jog" so you won't have to keep running back to this manual. In addition, where input values are required in the program, the default settings are shown in parentheses. Don't be overwhelmed by the seemingly large amount of commands available. This program is not complicated, and most of the commands are used only once, to set operating parameters in the program, or very infrequently during normal operation. And after all, you did conquer the command structure of the TNC, didn't you? The TAPR TNC-1 has over 70 commands, while this program has only 30 commands, that operate the program, 11 editing commands (used only during editing), and 7 Control key functions.

In Chapter 2 the program commands, the edit modes and commands, and the control codes will each be fully explained with examples given. In no time at all, after studying them, in conjunction with some hands on experimentation, you will be able to create your own customized version of this program, and will then be in for hours of trouble free operating pleasure, whether in an attended or unattended mode.

## =====

## COMMAND SCREENS

=====

**CONV <ENTER>**

This command is used to enter the **CONV**ersation mode. This mode may also be entered by pressing the <ENTER> key. This is the normal mode to use for "live and direct" conversations, or to send commands to the TNC and is automatically entered when the program is first started up. Most keys pressed on the keyboard are sent to the TNC as you type them, without regard to line length or any formatting, such as word-wrap. Special function control keys are still functional in the program, allowing you to send the TNC the contents of the **ITEXT** buffer, or any of the 10 line buffers, by typing a **CTRL-@**, or a **CTRL-n** (n=any number 0 through 9). You will not get any indication on the screen when sending a line buffer. You will just have to trust us that what you had in there is going out. (From Screen 2 and Program 27-28-29-30)

All control keys, such as **CTRL-S** will be passed on to the TNC in the **CONV** mode, except for **CTRL-P**, which is used in the program to toggle the printer on and off, **CTRL-H** which is used to toggle the **COPY** mode on and off, and **CTRL-J** which is used to toggle the **FORMAT** mode on and off. If you have a requirement to send a **CTRL-H** to the TNC, this can be accomplished by use of the **BACK-ARROW**. Similarly, the **CTRL-J** can be sent with a **DOWN-ARROW**. This leaves **CTRL-P** as the only control key which cannot be sent to the TNC.

**LOAD FILE/EXT [:D] <ENTER>**

This command will load an ascii file, previously saved with the **SAVE** command, from disk into the **MAIN** text buffer.

EXAMPLE: **LOAD DAILYLOG/TXT:1 <ENTER>**

**SAVE FILE/EXT [:D] <ENTER>**

This command will save the contents of the **MAIN** text buffer to disk.

EXAMPLE: **SAVE DAILYLOG/TXT:1 <ENTER>**

**SEND <ENTER>**

This command will transmit the contents of the **MAIN** text buffer to the TNC. You may preload the **MAIN** text buffer to the limits earlier declared, which are approximately 23K for the Model I, 22K for the Model III, and 30K for the Model 4.

**EDIT <ENTER>**

This command will allow you to edit the contents of any buffer. See **EDIT MODE BUFFER SELECTION** for a description of buffer selection commands. Refer to **EDIT MODE COMMAND SCREEN** for editing commands available.

**DISPLAY <ENTER>**  
**DISPLAY FILE/EXT [:D] <ENTER>**

This command is used to display the contents of the **MAIN** text buffer. If **FILE/EXT** is also added, the program will load the contents of a disk file into the **MAIN** text buffer and display it on the upper screen. With this second option, any existing data in the **MAIN** text buffer is destroyed.

While the data is being displayed, whether from the **MAIN** text buffer or a disk file, pressing the number keys (1 through 9) will speed up or slow down the display (1 being the fastest). Pressing the space bar will stop the display and pressing it a second time will start it again. Pressing the **BREAK** key will terminate the display mode and leave the last loaded data in the **MAIN** text buffer. The message "Press any key" will be annunciated in the bottom, or worktable, area of the screen when the display of a file is completed.

**NOTE:** If you are displaying a file that was created with the **AUTOSAVE** command, the disk will be accessed repeatedly, as each segment of the file is displayed, in the order it was saved. The display is not finished until the message "Press any key" appears in the worktable area.

**EXAMPLES:** **DISPLAY <ENTER>**  
**DISPLAY DAILYLOG/TXT:1 <ENTER>**

**CONTDISP <ENTER>**

If the display of a disk file, which was saved with **AUTOSAVE**, was terminated by a **BREAK** key, prior to displaying all of the information in the file, **CONTDISP** will allow you to **CONTINUE DISPLAYING** where you left off. This command will **NOT** work if, during the interruption, you performed a **DISPLAY** of any other file at all. Attempting to use this command on any file not saved by the **AUTOSAVE** command will result in you only seeing the last few bytes of the file. The reason this command was created is that in the **AUTOSAVE** mode, the disk file created is actually composed of many "sub-files". Each sub-file has an EOF that tells the computer to go fetch the next part of the file. At the absolute end of the file, there is an EOF that tells the program that all of the file has been seen. If you have saved 50K of text from the Packet Network you are a part of, and are viewing it, you would not want to have to start all over, if you had to break out of the display to answer a connection, or for some other reason. This command allows you to get back to where you were and continue on with the display, so long as you didn't attempt to display any other files while you were away.

**AUTOSAVE FILE/EXT [:D] <ENTER>**

This command is used to put the program into the automatic disk save mode for unattended operation. All data received will be saved onto disk for later viewing with the **DISPLAY** and **CONTDISP** commands. While this mode is active, the contents of the **CTEXT** buffer will be transmitted when any station connects to you. Press **SHIFT-BREAK** to exit the **AUTOSAVE** mode.

**Note:** The program uses the TNC message "\*\*\* CONNECTED TO" message to determine when a

connection has been made to your TNC. If this message shows up in the text of anything you receive with the TNC monitor ON, the program will assume a connection has been made and will issue the buffer even though you are not actually connected to anyone. (This message might appear, for example, in the output of a mailbox or gateway when a connection has been made out of the L.A.N. by another station whom you are monitoring.) In this case, the buffer will go out in the TNC UNPROTO mode. Should you decide that you do not want to involuntarily participate in these unscheduled transmissions, it may be best to leave the CTEXT buffer blank, or use the ACKOFF command to prevent any automatic response from your station.

EXAMPLE: AUTOSAVE DAILYLOG/TXT:1 <ENTER>

PGMSAVE FILE/EXT [:D] [FILE/EXT [:D]] <ENTER>

This command will save the program in its initialized form to disk. After you have answered the baud rate question, at the initialization stage of the program, you can then place your own customized messages and connect sequences into the various buffers. Using this command will allow you to save the entire program with the baud rate present and the buffers already loaded. The program will then bypass the baud rate question and immediately enter the CONV mode with the buffer COPY turned on when it is booted.

If the [FILE/EXT] option is used, a program will be saved that will automatically boot up, then go into AUTOSAVE mode, using the FILE/EXT as the name of the file to save incoming data to. This is useful for power outages, where the computer can autoboot this program when the power is restored, while unattended. To fully understand this function, observe the following sequence of example commands:

- (1) Create a version of the program and save it to disk using the command:

PGMSAVE PACKET/CMD:0 DAILYLOG/TXT:1 <ENTER>

This will save an autoboot version of the program which when loaded will automatically enter the AUTOSAVE mode with a file called DAILYLOG/TXT being created on disk drive number 1.

- (2) From the COMMAND mode, type in DOS <ENTER> to return to TRSDOS. Now type in:

AUTO PACKET <ENTER>

This will cause the DOS to autoexecute the file PACKET/CMD anytime it is booted. Now press the RESET button on the computer and watch the results! The DOS will boot, then automatically load and run the program PACKET/CMD, and when it has loaded up, it will go into AUTOSAVE mode (the mode window will annunciate DISK) with incoming text being saved in the file DAILYLOG/TXT. So long as the system disk with the AUTO function on it remains in drive :0, this sequence will be repeated each time the DOS boots up, such as in the event of restoration from a power failure. This can be a very useful utility, as your normal station operation can be restored unattended.

NOTE: Holding a SHIFT key down during boot will abort the AUTOSAVE function.

EXAMPLES: PGMSAVE PACKET/CMD:0 <ENTER> (Non-autoboot version)  
PGMSAVE PACKET/CMD:0 DAILYLOG/TXT:1 <ENTER> (Autoboot version to AUTOSAVE)

**RCVFILE FILE/EXT [:D] <ENTER>**  
**SENDFILE FILE/EXT [:D] <ENTER>**

These commands are used to transfer a file from the disk of the sending station to the disk of the receiving station. Any file, regardless of its contents, may be transferred. In this program, though, all of the work of file transfers has been taken out, leaving you to only ask the program to get on with the job. When you select this mode, the TNC will automatically be conditioned for 8 bit transfers by being placed in the TRANSPARENT mode with the parity set to 4 and PACTIME set to AFTER 1. And, assuming you have set the values in the TNC asked for on page 3, when the transfer session is done, the program will exit this mode and send 3 CTRL-C's to the TNC, and then place it back in CONV mode.

You will only be able to transfer programs between yourself and other stations who use the file transfer protocol used in this program. At the present time, no other programs are known to be using this particular protocol, which is similar to the popular XMODEM protocol, but with differences to enable better error trapping over a radio link.

For those with inquisitive minds, who are wondering why there is another layer of CRC error checking built into a program, that feeds a device whose whole reason for being is to do error checking on the messages, we include an explanation. During the development of this program, it became apparent that errors could sneak into data transfers on the RS-232 side of the TNC. Not all errors can be attributed to bad handshaking between the TNC and the computer, either, as sometimes the data can become invalid due to R.F. energy from another transmitter getting into a Model I, for example. The job of the TNC is to error check the data on the radio side of the link, and no where else. Once that data is shipped out the RS-232 port to the computer, it is on its own. Having another level of error checking built in, does not slow the program down significantly, and does allow FULLY error free file transfers. If you are thinking in terms of ASCII file transfers, you might believe that 1 bad byte in 10K bytes is not such a bad deal; however, in a machine language file transfer, 1 bad byte in 10K bytes may render a program virtually useless, and you might never find the error either!

When sending a file, your screen will annunciate how many blocks (128 bytes/block) are in the file. The average rate of transfer, working station to station, (no digipeating, and no other traffic on the frequency) is 12 blocks per minute when both stations are able to work with TXDELAY = 5 and AXDELAY = 5. This works out to approximately 15K in 10 minutes.

**KILL FILE/EXT [:D] <ENTER>**

Delete a file from the disk.

Note: This is used on the Model 4 also. The normal Model 4 REMOVE command will not work.

**DIR [:D] <ENTER>**

Display disk directory. This command is not available on the Model I computer.

**DOS <ENTER>**

CTRL-S

This command is used to exit the program and return to the DOS. It is important to use this command rather than using the reset button. When the DOS command is used, the program sends the contents of the EXIT buffer (ETEXT) to the TNC. You should have this buffer set up with a "graceful" exit sequence to be sent, such as a CTRL-D to shut the TNC data output off. If the TNC buffer does not fill up with data during the time you do not have the computer servicing it, the data will be available for display when you again use the packet program. One suggestion for the ETEXT buffer, besides the CTRL-D, is to send a MON OFF command to the TNC, allowing it to only save connected messages in the buffer. This will allow maximum utilization of the short amount of buffer space in the TNC, and may save an important message for you. If you use the computer reset button, all data received while the packet program was not being used will be lost. LS

**BAUDSET <ENTER>**

This command is used to reset the computer to TNC link baud rate. After pressing <ENTER>, a list of possible baudrates will be given to choose from.

**VIDTIME n <ENTER> - Default 1 minute**

This command is used to select the time, in minutes, before the screen is turned off when in the DISCONNECTED mode when there has been no activity. The command has no effect if you are connected to another station, or if you are viewing the MENU. This feature is used to reduce the wear and tear on your CRT when nothing is happening (screen burn-in).

Any data received from the TNC, as well as pressing any key on the keyboard will bring the CRT back to life. If you find the screen blank and wish to bring it back to life to see what happened last, simply press the SHIFT or the CTRL key, as these keys will bring the display back without altering anything in the program.

The value of n can be anything from 0 to 255. If 0 is selected, the video will not turn off.

EXAMPLE: VIDTIME 10 <ENTER>

The above example selects a 10 minute timer before the screen goes blank from no activity.

**CONNTIME N <ENTER> - Default 5 minutes**

This command is used to select the time after which the program will test the data link when no input has been received from the TNC for "n" minutes. This timer only operates in the CONNECTED mode. After "n" minutes of no activity, a carriage return is sent to the TNC to test the link. If the link exists, an ACK will be received by the TNC and everything continues as before. However, if the link does not exist the TNC will time out on retries and disconnect on its own. This sequence may be altered by the RETRY command. Any value from 0 to 255 may be entered. If 0 is selected, the link will not be tested and the program will not attempt an automatic disconnect.

EXAMPLE: CONNTIME 1 <ENTER>

The above example will causes the link to be tested after 1 minute of no activity.

**RETRY n <ENTER>** - Default 5 retries

The **RETRY** command selects the number of times the program will test the link for connect status before disconnecting. If **RETRY=0** then the link will not be tested and the program will begin the disconnect sequence after the first **CONNTIME** timeout. If **RETRY** does not equal 0, then the program will begin the disconnect sequence after the link has been tested "n" times. For example, if **RETRY** was set to 1 and **CONNTIME** was set to 5, after 5 minutes the link would be tested and if it exists the program will disconnect after the next 5 minutes timeout from no activity (no data received and no keys pressed).

**NOTE:** PRESSING ANY KEY ON THE KEYBOARD OR RECEIVING ANY DATA WILL RESET ALL OF THE ABOVE DISCUSSED TIMERS.

**PNTLINE n <ENTER>** - Default 80

This command sets the maximum line length for the line printer. If the received data extends past this point, it will be wordwrapped and placed on the next line.

**LINE n <ENTER>** - Default 79

This command sets the maximum line length of data sent from the **MAIN** text buffer in **SEND** mode. (This line counter has no effect on direct typed data from the program **CONV** mode.)

**LFON <ENTER>**

This command enables the LF after CR switch for the line printer. When it is ON, a LF will be sent to the printer each time the program sends a CR.

**LOFF <ENTER>** - This is the default setting

This command disables the LF after CR switch for the line printer.

**ACKON <ENTER>** - This is the default setting

This command is used to enable the \*\*\* **ACKNOWLEDGED** and **CTEXT** message transmission. When **ACKON** is enabled, the program will automatically transmit any text in the **CTEXT** buffer upon being connected to. In addition, another special feature of the program allows for remote testing of your station, by a connected party, to determine if you have received the last packet. The remote command is triggered by the programs receipt of a **CTRL-W** whenever you are connected to someone. This is a handy feature for older versions of TNC's which do not have a means available to identify if the last packet has gone out or not. By simply ending their text with the **CTRL-W**, they (and you, if you are sending to a station running this program) will be able to determine when it is safe to disconnect and avoid the heartburn of watching the last half of a lengthy message go out UNPROTO in full view of the entire L.A.N.

**ACKOFF <ENTER>**



As you might suspect, this command disables the \*\*\* ACKNOWLEDGE and CTEXT message transmission.

=====

EDIT MODE BUFFER SELECTION SCREEN

=====

After selecting EDIT mode from the COMMAND mode, you will see a prompt on the bottom screen:

Press key for buffer to edit or press enter

At this point, if you press <ENTER> you will be returned to the COMMAND mode. You may also enter any of the buffers in an active edit state, by pressing a key or CTRL function, as shown on the EDIT MODE BUFFER SELECTION screen. The choices are:

M - Edit, or add to, the MAIN buffer.

C - Edit, or add to, the CTEXT buffer.

@ - Edit, or add to, the ETEXT buffer.

CONTROL @ - Edit, or add to, the ITEXT buffer.

CONTROL 0-9 - Edit, or add to, any of 10 line buffers.

#### MAIN TEXT BUFFER

The MAIN text buffer is from 22K to 30K long, depending on your computer model. (See Chapter 1) It can be loaded from the TNC in COPY mode, from a disk file, or from the keyboard. It can be sent from the COMMAND mode to the TNC using the SEND command. The MAIN text buffer can be saved to disk using the SAVE command.

#### CONNECT TEXT BUFFER

The CTEXT buffer is 200 characters in length. It can be loaded from a disk file or via the keyboard. It is only sent to the TNC when a station connects to you, AND you are in AUTOSAVE mode. It cannot be sent manually. A typical example of how this buffer could be loaded is:

This is WA7ZZZ in Whatchamahoozit, WA. This station is in  
AUTOSAVE. Please leave your message and disconnect.  
<ENTER>

Each station connecting to you while you were in AUTOSAVE would get this message and know that you were in unattended operation, but capable of storing messages to disk, so they wouldn't have to be held until later, or placed on the local mailbox.

### EXIT TEXT BUFFER

The ETEXT buffer is 200 characters in length and may be loaded from a disk file, or via the keyboard. This buffer is automatically issued to the TNC when the DOS command is used to exit the program to TRSDOS. This allows a means of programming a "shutdown" sequence for the TNC, so that incoming messages will be held in the TNC buffer while the computer is disconnected. A typical setup for this buffer might be:

CTRL-C MON OFF <ENTER>  
CTRL-S

Such a sequence would serve the purpose of setting the TNC to only listen to connected messages, but to hold them in the buffer, since the CTRL-S serves to shut off the data flow.

### INITIALIZATION TEXT BUFFER

The ITEXT buffer is 200 characters long and can be loaded from the keyboard or from a disk file. This buffer is issued to the TNC automatically when the program is booted up. It can also be forced to the TNC by typing a CTRL-Q. The purpose of this buffer is to provide a means of storing all of the non-permable data you use in the TNC, such as long lists of MTO and MFROM stations (or BUDLISTS in the newer TNC's), plus your BEACON text and your UNPROTO command. If you have to do a reset on your TNC, or have suffered a power outage (as described under PGMSAVE), you can reissue all of the data to the TNC in one keystroke.

**HINT:** Always start your ITEXT buffer with a CTRL-Q, to enable the RS-232 link, if you are booting the program after having exited to DOS previously. (It doesn't hurt to send it anyway.)

### LINE BUFFERS

There are 10 line buffers, 140 characters in length, which can be loaded from a disk file or from the keyboard. They are sent to the TNC by holding down the CTRL key and then hitting the buffer number you want sent. Editing is entered the same way. The best use of these buffers is to hold the connect sequences of your favorite stations so that you only have to hit one key to connect with them. This is particularly useful if you know some stations who can only be reached via multiple relay digipeating.

### ARROW KEYS

After you have selected EDIT mode, and if you have selected the MAIN text buffer to edit or add to, you will be presented with a question on the bottom screen:

NEW TEXT (Y=YES)?

If you answer "Y", the MAIN text buffer will be zeroed out and all previous contents lost. You will then be placed automatically in the ADD mode, and any typing you do will be placed in the buffer, with the buffer counter being updated accordingly. If you answer with "N", you will see any contents in the buffer scroll by on the bottom screen rapidly (QUITE rapidly!) and you will find yourself placed at the end of the file in the ADD mode. At this point, any typing you do will be ADDED to the end of the file and the

counter window will be increased.

Any other buffer selected will show its contents on the bottom screen. To exit from the buffer, leaving it unchanged, simply press the BREAK key. Use **SHIFT BREAK** to get to ADD mode.

After you have been placed in the ADD mode, there are two ARROW commands you can use:

#### **SHIFT UP ARROW**

Used to load a disk file into the buffer. To use, hold the SHIFT key down and then hit the UP ARROW. The bottom screen will show:

**FILENAME?**

>

If you enter the name of an ASCII disk file, it will be loaded into the buffer you are working with, after the last active byte in the buffer and the counter will be updated accordingly. Non-ASCII files will produce unpredictable results. ASCII files not created by this program (such as BASIC programs saved in ASCII) may be able to be sent. See the **ODDS AND ENDS** section for further information.

#### **SHIFT DOWN ARROW**

This arrow is the EDIT/ADD toggle. Pressing it once, when in the ADD mode will reformat the bottom screen and place the cursor at the beginning of the file for editing. Pressing it again will reformat the screen and place the cursor at the end of the file for ADDing text to the buffer. You may hop back and forth between EDIT and ADD at will. For commands available to you when in the EDIT mode, refer to the next section of this manual.

=====

#### **EDIT MODE COMMAND SCREEN**

=====

After you have entered a buffer, as described in the preceding section, you can use the SHIFT DOWN ARROW to toggle between the ADD and the EDIT mode. When in the EDIT mode, the following commands are available to provide full editing capabilities of the buffer you are working in.

|                    |                                    |
|--------------------|------------------------------------|
| <b>UP ARROW</b>    | Move the cursor up one line        |
| <b>DOWN ARROW</b>  | Move the cursor down one line      |
| <b>LEFT ARROW</b>  | Move the cursor left one space     |
| <b>RIGHT ARROW</b> | Move the cursor right 10 spaces    |
| <b>SPACE BAR</b>   | Move the cursor right one space    |
| <b>A</b>           | Delete all text from cursor to end |

|                  |  |
|------------------|--|
| B                | Delete all text from cursor to beginning                       |
| D                | Delete character under cursor                                  |
| I                | Select insert mode   |
| BREAK KEY        | Exit the edit mode and return to buffer selection mode screen. |
| SHIFT DOWN ARROW | Toggle between EDIT and ADD modes                              |

While in the INSERT mode:

|                |                                       |
|----------------|---------------------------------------|
| LEFT ARROW     | Delete character to left of cursor    |
| SHIFT UP ARROW | Exit the INSERT mode                  |
| BREAK          | Exit the INSERT mode                  |
| ANY OTHER KEY  | Insert that character into the buffer |

=====

CONTROL CODE SCREEN

=====

Certain control characters are used to toggle or select special functions in this program. Any control character may also be sent to the TNC for other functions, either by directly typing it, or by substitution, except for the CTRL-P. Refer to the description on page 7 for a full explanation.

|           |  |
|-----------|--|
| BREAK     | Exit current mode  |
| CONTROL-J | Select screen format ON/OFF. This command can be used to toss away the CR function for the screen. This allows for the incoming data to fill the entire screen, using wordwrapping, rather than having partial lines wherever a CR occurs. It can be confusing watching the screen in this mode, unless you are connected to someone. Whenever you are in the FORMAT mode, a window on the status line will annunciate the fact. |
| CONTROL-P | Select lineprinter output ON/OFF. A window on the status line will show PRINT whenever you are in the lineprinting mode.   |
| CONTROL-M | Issue Disconnect sequence to TNC (This will be a CTRL-C, followed by a D and a CR)   |
| CONTROL-H | Copy into MAIN text buffer (CONV mode). A window on the status will annunciate COPY whenever that mode is active.  |
| CONTROL-G | Sends the BELL character along with a <BEEP> for video stimulation. (This is handy if you or your friends don't have an audible beeper, such as a line printer, or a Model 4 with a sound  |

board.)

CONTROL 0-9

Send one of 10 line buffers to TNC

CONTROL-@

Send contents of ITEXT buffer to TNC

TNC COMMANDS: (FROM COMMAND MODE)

CTRL-Q

ENABLE RS-232

CTRL-S

DISABLE RS-232

CTRL-Y

CANCEL PACKET / CANCEL DISPLAY TO TERM. IN CONV. MODE

CTRL-C

ENTER TNC COMMAND MODE

CTRL-X

CANCEL LINE

CONS ON/OFF

DAY/TIME START ON/OFF

DAY/TIME

YYmmdd hhmm

DISPLAY

ALL PACKET PARAMETERS

## =====

## ODDS AND ENDS

=====

We have now arrived at the end of the manual. This is the place where most authors put the "Index", the "Addendum" or the "Appendix". Those are the sections that explain the wire connections on a TRS-80 RS-232 DB25 connector, or tell you how the keyboard is layed out, or give you something you REALLY need, like a list of all the ASCII codes. We will not bother with those subjects, on the assumption that you have purchased a computer-related book in the last few years. There are really not enough pages in this manual to require an "Index" for fast look-up of a particular subject or command, and so we shall dispense with that too.

When the first two chapters were finished and formatted, there remained all over the ham shack here, little bits of paper, cards and envelopes with notes scribbled on them saying things like, "Be sure and mention who worked on the program" and "Explain the disksave sequence better". Those items, which are the "ODDS AND ENDS" of this manual, will be dealt with in this chapter. With that explanation out of the way, let us address this miscellany.

**THE PROGRAM AUTHOR**

This program was written by Craig Larsen, WA7HTN, the same ham who wrote the software for the ROM-116. Those of you who have been privileged to own one of those units will recognize the same style of programming in this program. It is functional, well organized and aimed at "ease of operation", with as many functions automated as possible.

**THE HELPERS (TESTERS AND TORTURERS)**

The program was extensively tested and debugged by 3 other hams in the area, each of whom possessed a different model, or version of a model, of TRS-80 computer.

The Model I version, with an RS-232 interface card, and also using the ROM-116, was tested by KB7DY. Dennis also tested the program in a Model 4 with 128K of memory. A TAPR TNC-1 was used in the test.

The Model 4P, with 128K of memory, was tested by N7CUR. Bill has the TAPR TNC-1, and a Kantronics Packet Communicator with Version 1.6 software in it.

The Model 4, with 64K of memory, was tested by WA7HDI. Joe has the Kantronics Packet Communicator with Version 1.5 software.

**SENDING ASCII FILES**

It was stated in the section on ARROW KEYS that ASCII files, not created by this program, may be able to be sent to the TNC. An example of this, and the trouble you can run into, is a BASIC program. You may have need to send a BASIC program to a friend, who does not have this program, and thus cannot do direct file transfers with you. In such a case, you will want to load your BASIC program, under BASIC, and then resave it in ASCII. Before doing so, we recommend that you go to the end of the program and add several lines of REM statements. The reason for this, is that BASIC does not "end mark" the files the

same way this program does. Trying to load such a file will cause a disk error number 29 in most cases. When you look at the **MAIN** text buffer to see what is in there, you may see the file loaded up to your **REM** statements and no more. In this condition, it is OK to send. You may also see the whole file, followed by a lot of graphic characters. In this case, use the editor to delete the junk off the end, and then send it. At the other end, a file will be received which, when loaded, will cause a "DIRECT STATEMENT IN FILE" error. Listing the program will show that it is all there, and that the error is caused by an improper terminator. Saving the file at the receiving end will restore the proper terminators and it will load properly from then on.

If you encounter difficulties transferring files to someone else who is running E-Z PACKET 80, you can bet you are doing something wrong. Make sure you have set the word length to eight and have selected no parity. There are several hundred people using this program and transferring files everyday. It seems highly unlikely that your program is the only one that has something wrong with it.

### THE BREAK KEY

Earlier in the manual, you were told that "the **BREAK** key will always return you to the **COMMAND** mode". This is a LIE, albeit an innocent one. In actuality, the **BREAK** key will always **LEAD** you back to the **COMMAND** mode. It happens that if you are in the **EDIT** mode and in the process of actively editing a buffer, it can take you as many as 3 pushes of the **BREAK** key to get back to **COMMAND** mode! (Sorry 'bout that.)

### DISKSAVES IN THE AUTOSAVE MODE

The manual is quite clear on the fact that, in the **AUTOSAVE** mode, data from the TNC is saved to the disk. What is not made clear is exactly how this is done. When you are in **AUTOSAVE** mode, everything sent to the program by the TNC will be stacked up in the **MAIN** text buffer. At the end of one minute, the data saved so far will be stored on the disk provided there are more than 256 bytes. The exception to this is when you have been connected to by someone else. In this instance, the save will take place, regardless of how many bytes there are, upon receipt, from the TNC, of the sequence "\*\*\* DISCONNECTED". So, now you know why you keep hearing the disk whirling around for short bursts of time when you are in **AUTOSAVE**.

### PRINTER PORT

The lineprinter must be connected to the parallel line printer port on the TRS-80. There is no other way.

### BORING FILTER

The program will print to screen, or save to disk (in the case of you having selected **AUTOSAVE** mode), every single thing sent to it by the TNC. If you want to save, and scroll through later, 100 % of the traffic on your L.A.N., just set your MTO (or equivalent) to **ALL** and the program will dutifully show it all to you. However, if you want to cut down on the disk activity or not have to wade through all that stuff, you will have to use the MTO or MFROM (or equivalent) commands in the TNC as a "BORING FILTER". No provisions are made in the program for SELCALs, or any other RTTY-like operation.

**BEEP BEEP**

Somehow, in the section on the <BEEP>, a fact got lost in the shuffle. Here it is: In the Model 4 version of this program, when someone connects to you, the sound board (if you possess one) will sound an audio alert. Additionally, it will beep at each receipt of a CTRL-G (BELL character).



DATE: 02/10/86

SUBJECT: E-Z PACKET 80 Program changes

VERSION: 1.1, 3.1 and 4.1

The above versions of E-Z PACKET 80 have the following additions to the program:

(1) Some TNC software requires the escape character (ESC) as part of the command structure. The escape character can now be generated from the keyboard by using the SHIFT LEFT-ARROW. This is done by holding down the shift key and pressing the LEFT-ARROW key. On the Model IV, function key F1 will also generate the escape character.

(2) The SHIFT @ had been added for use with the multi-connect feature provided by the TAPR-2. After pressing the SHIFT @, you will then see the vertical line (|) on the screen. You may then enter the stream switch letter you desire.



DATE: 06/09/86

SUBJECT: E-Z PACKET 80 Program changes

VERSION: 1.2, 3.2 and 4.2

The above versions of E-Z PACKET 80 have the following additions to the program:

(1) The file transfer routines have been improved. These changes will not be apparent to the user, but will make file transfer more efficient.

(2) Wordwrapping has now been added to the conversation mode. This will result in cleaner appearing text when text is being sent to a packet mailbox.

(3) When a line buffer is sent, the contents will now appear in the lower portion of the screen. This will take the mystery out of knowing whether or not the contents of the buffer was really sent. This was done to assist those who have little faith.