Omniterm™

PROFESSIONAL COMMUNICATIONS SOFTWARE

OMNITERM, the smart terminal program for TRS-80 microcomputers, wins top ratings from major product reviewers.

"OMNITERM is a breakthrough in terminal programs."

"OMNITERM has become the standard for TRS-80 communications software."

"The most versatile of its kind."

"OMNITERM is a must for anyone who wants to get the most out of their TRS-80 communications."

"OMNITERM has all the features you'll find in the most expensive terminal programs today."

- 5 Star Award
  - 80 Micro
- Software Review
  - 80 Micro
- 80 Microcomputing
- Byte
- Infoworld
- Microcomputing

System requirements

Models I and III:

-
-
-
-
-
-

Models II and 12:

-
-
-
-

OMNITERM
Version 1.10 for TRS-80 Models II/12
Catalog Number 90-0124

Omniterm is the most flexible, powerful terminal program you can buy.

Omniterm is the advanced terminal program that lets your TRS-80 communicate with practically all of the world's computers...mainframe, minicomputer, or microcomputer. With Omniterm, you can conduct two-way communications with any asynchronous ASCII based-computer system.

Omniterm can put you in touch with your company's mainframe, for instance. With your branch offices, clients, customers, and suppliers. With friends or associates with personal computers. With time-sharing computers. With bulleting board systems and university computers. And with videotex services such as CompuServe, Delphi and Source.

Omniterm comes complete with

- Conversion utilities
- Practical text editor
- Sample setting files
- Comprehensive user manual with full index and glossary

Plus every advanced feature

- Christensen XMODEM file transfer protocol (up to 9600 baud)
  - — Automatic transfer of unlimited length text or binary files
  - — Automatic error detection and correction
- XON/XOFF support
- Character pauses
- Line pauses
- Full RS-232 control
  - — Set baud rates (maximum 2400 Models I and III; maximum 4800 Models 11 and 12)
  - — Even, odd, or no parity
  - — 5—8 data bits
  - — 1, 1½, or 2 stop bits
  - — RTS and DTR control
  - — View CTS, DSR, CD, RI
- Carriage return and line feed handling
- Seven translation tables
  - — Translate any or all 256 characters
  - — Complete compatibility of input and output devices
- Run your printer
- Full cursor control
- Rag-free screen reformatting
- Scroll-back text review
- Autodial and sign-on
- On-line changes
- Save and restore settings
- Half or full duplex
- Echoing

Lindbergh Systems, Inc.

[redacted]

Source: TCA818
Delphi: LINDBERGH
CompuServe: 70310, 267

Omniterm
PROFESSIONAL COMMUNICATIONS SOFTWARE

Lindbergh Systems, Inc.
P.O. Box 3604
Ann Arbor, MI 48104

Dear Customer,

Thank you for purchasing our software. We have tried to design a program that is both powerful, yet easy to understand and use. The User's Manual explains the features and some of the uses of the program and we ask that you read it carefully.

We also understand that communications is a confusing and difficult undertaking for the uninitiated. Should you experience difficulties that are not addressed in the Troubleshooting Guide (Appendix M) or elsewhere in the manual, call Lindbergh Systems, Inc. for assistance. Be sure to note all the details of your configuration (what computer, modem, version of software, type of remote computer you are calling) and the nature of the problem before you call.

Our support number is (313) 971-9733

You must be a registered owner to qualify for support so please read the software license that is printed on the inside cover of the User's Manual and sign and return the enclosed Owner Agreement and Registration card.

OMNITERM Model II/12 Version 1.10 Addendum
Changes to OMNITERM Release 4 Manual of 05/07/84
For TRS-80 Model II/12 with 64K and 1 Disk minimum
Friday, May 11 1984

Lindbergh Systems

P.O. Box 3604
Ann Arbor, MI 48106
(313)971-9733

****

It is recommended that the new OMNITERM owner read the OMNITERM User's Manual before reading this addendum. This addendum assumes familiarity with OMNITERM and the terms and concepts used to describe it. These are described in detail in the user's manual.

(TRSDOS and TRS-80 are registered trademarks of Tandy Corporation.)

Introduction

OMNITERM for the TRS-80 Model II/12/16 is an adaptation of the original OMNITERM program for the TRS-80 Model I/Model III. These adaptations, written by Bill Knaack of Sioux City, Iowa, include all significant features of the original OMNITERM, plus several enhancements. This version is intended for the Model II or 12 and runs on the TRS-80 Model 16 only in its "Model II emulation" mode. In this mode it is in all respects identical to the Model II. For the sake of brevity we will refer to the Model II/12/16 version simply as "Model II OMNITERM." The original Model I/Model III OMNITERM will be called "original OMNITERM."

This addendum is intended as a quick guide to the differences between original OMNITERM, as described in the OMNITERM manual, and this new version. As such, it assumes that the reader is already familar with the original OMNITERM program. If you are totally new to OMNITERM, we strongly suggest you read the OMNITERM manual.

To enable you to easily go back and forth between this addendum and the manual, we will keep the same subject headings and sequence as much as possible.

System Requirements

Model II OMNITERM runs on a TRS-80 Model II, Model 12, or Model 16 computer with a minimum of 64K of RAM and at least one disk drive. To use OMNITERM over telephone lines, you will also need a telephone modem compatible with the computer system you want to communicate with.

Getting Started

OMNITERM is delivered on a 8" floppy disk. Owners of multi-drive systems may use their OMNITERM as they would any other data disk. Owners of one-drive systems should do the following: 1) boot their TRSDOS 2.0 or TRSDOS II. 2) When the "TRSDOS READY" prompt appears, enter XFERSYS and then simply follow the instructions the system gives you. 3) Thereafter, you would boot the OMNITERM disk to load OMNITERM and start it running. For more details, please consult your TRSDOS manual concerning XFERSYS.

Using OMNITERM

Model II OMNITERM is more similar to original OMNITERM running on the TRS-80 Model III than on the Model I. The Model II, like the Model III, has no RS-232 board switch settings, and it has a complete UPPER/lower case

keyboard and display. Model II OMNITERM thus acts like the Model III in these respects. Also, as in Model III TRSDOS, the <BREAK> key can be used at any time in Model II OMNITERM to stop what you are doing. This is in addition to the occasions mentioned in the OMNITERM manual.

OMNITERM Normal Mode

The only major difference here is that the auto-signon string is sent with the <F2> key, not control-0 as described in the OMNITERM manual.

OMNITERM Command Mode

To enter the Model II OMNITERM command mode, use the <F1> key, not the @ key as in original OMNITERM. You need only press the <F1> key once to enter the Model II OMNITERM command mode. When you want to leave command mode, simply press the <F1> key again, and you will return to OMNITERM's normal mode. This is in contrast to the <BREAK> key used in original OMNITERM for this purpose.

Model II OMNITERM's command mode screen is slightly different from that of original OMNITERM's. In the lower middle portion you will still see the CTS and CD RS-232 control lines, but not the DSR or RI lines. Instead, you will see a display showing which of the Model II's two ports OMNITERM is using, A or B. This is because the Model II has two separate RS-232 ports, unlike the Model I or Model III TRS-80 computers. Remember that OMNITERM must be set to use the same port (A or B) to which you have connected your modem or other device. If not, OMNITERM will be unable to work with your device.

The Model II display is larger than that of the Model III. As a result, there is no need for the bottom of the command mode screen to be overwritten by commands. Instead, these commands will use the area at the bottom of your screen. Normally, you will see the message:

```
Enter your command ==>
```

at the bottom of your screen. This tells you that OMNITERM is in the command mode and ready to accept your command.

Summary of Command Mode Functions

The only differnce here is that Model II OMNITERM does not use the @ key to enter the command mode. Therefore, the @ SEND "AT" SYMBOL & QUIT function is not needed. In its place is a new function, K EXECUTE DOS KOMMAND that allows you to do any TRSDOS command such as DIR, FREE, etc. from OMNITERM. This will be described in detail later.

I INPUT TO BUFFER

When you press the I key in Model II OMNITERM it will ask you to:

```
Enter save filespec ==>
```

A file will then be opened with this name to receive the contents of the buffer. This differs from and is more powerful than the way original OMNITERM handled input to buffer. When the Model II buffer has filled, the system will send an XOFF (CTRL-S, decimal 19) to the other computer. This tells it to stop transmitting. In many systems, transmission will stop within two or three characters. These characters will not be lost; however, if more than 15 characters are sent by the remote computer after XOFF, then the excess data will be lost.

Some remote systems will not recognize the XOFF command. When this happens, you will have to handle input to buffer the same way as described in the original manual.

After the XOFF is sent the contents of the buffer are written to disk. The message

```
Writing buffer to disk...
```

appears during this process. OMNITERM then sends an XON (CTRL-Q, decimal 17) to tell the other computer to restart transmission. This cycle will repeat until the entire file has been received.

When the other side has sent the entire file, it will stop sending and the data will stop flashing across the screen. Press the <F1> key to go back to the command mode. Press I and the Input to Buffer status will go to OFF. At this point, any data remaining in the buffer will be written to disk, and the file will be closed.

Command Mode Functions -- Left Column

R SCREEN REFORMATTING

Since the Model II screen is normally 80 columns wide, this function is usually not needed in Model II OMNITERM. While it is still available to you if you wish to use it, by default it is turned OFF.

Command Mode Functions -- Right Column

System Commands

E EXAMINE DISK I/O BUFFER

This is the same function as in original OMNITERM, but it works differently. In the old format you would press the space bar to see text, press the <BREAK> key to quit, and the <ENTER> key to restart from the beginning. In Model II OMNITERM, the system asks you to

```
Press <REPEAT> and <SPACE> to see text,
  <F1> to quit, or <ENTER> to restart
```

R RTS OUTPUT LINE and D DTR OUTPUT LINE

Both the RTS (Request-To-Send) and DTR (Data-Terminal-Ready) RS-232 output lines are by default turned ON in the Model II. Since ON signifies positive voltage, both RTS and DTR are at positive voltage.

M MODEM

This command is not used in the Model II. It was originally used to allow a Microconnection modem to work on the Model I. Since there is no version of the Microconnection for the Model II, there is no need for this command. Instead, the RS-232 interface is offered as standard.

T CHANGE/EXAMINE TABLES

Control Key Table

This table is missing from Model II OMNITERM. It was only needed in the first place because Models I and III did not have a control key. It was necessary for these machines to use the down arrow key, and the Control Key Table was used to translate related entries into OMNITERM commands. The Model II does have a control key, and this table is no longer needed.

Editing and Modifying Tables

The Model II OMNITERM uses different keys to go to the next or previous page. It will tell you to:

```
Hit <UP-ARROW> for previous page, <DOWN-ARROW>
      for next, <ENTER> to alter value
```

U CHANGE UART SETTINGS

Under TRSDOS 2.0 and TRSDOS II, operation is quite similar to the original OMNITERM. However, Models I and III only supported one RS-232 port. The Model II has two, A and B. A is the default. To use B, press the U key for Change UART Settings. Make changes as described in the manual or press the <ENTER> key until you reach the line after "parity." The old settings are kept when you just press <ENTER>.

Now OMNITERM shows:

```
Port (A,B) =
```

Respond to this with either A or B. Then press <ENTER>, and OMNITERM will reconfigure the UART to the new settings.

The possible baud rates in Model II OMNITERM are 110, 150, 300, 600, 1200, 2400, 4800, and 9600. This differs from what you read in the manual. Although you may configure OMNITERM to run above 4800 baud, it is not recommended. See Appendix F in this addendum for more details on baud rates.

The default UART settings in Model II OMNITERM are 8 data bits, 1 stop bit, and no parity. With these settings, OMNITERM will still receive data sent to it with 7 data bits and even parity, but will "throw away" the parity bit in the From Comm Line table. When sending bytes, most systems will get parity errors when they see that data coming from OMNITERM in this format, but this should not cause any problems because the data itself will still be correct. Some systems will however have problems with this, and for these we suggest

you simply change the UART settings back to 7 data bits, 1 stop, and even parity.

This change from the default settings described in the OMNITERM manual has been made possible because OMNITERM's From Comm Line table now (by default) translates all bytes with values greater than 127 to their below-127 equivalents. It will make life easier for users of Radio Shack Modem IIs, and means that OMNITERM will be able to be used with more systems without need for changing the UART settings. Earlier versions of OMNITERM used default settings of 7 data bits, 1 stop bit, and even parity because this was (and is) the most common UART setting. However the changes to OMNITERM's From Comm Line table make it possible to use OMNITERM with most of these systems, as well as allowing OMNITERM to be used unchanged with systems that formerly required special UART settings.

@ SEND "AT" SYMBOL & QUIT

This command is no longer needed. In Models I and III, you would press the @ key twice to go into the Command Mode. In Model II you need only press <F1>. This frees up the "@" symbol to be sent as a normal data character.

K EXECUTE DOS KOMMAND

This is a new command in Model II OMNITERM. After entering K, you will see:

```
Enter a DOS command ==>
```

You may enter any standard TRSDOS command such as DIR, FREE, etc. The command will be executed. Then, press any key to return to the OMNITERM command mode menu. We do not recommend that you enter the name of a program, such as BASIC, because such a program will overwrite and destroy the OMNITERM program in memory. Such commands should be used by experienced programmers only.

B SCROLL BACK DISPLAY

This is the same function as in original OMNITERM, except that different keys are used. In the old format you would press the space bar to see text, press <BREAK> to quit, and the <ENTER> key to restart from the beginning. In the Model II OMNITERM, the system asks you to

```
Press <REPEAT> and <SPACE> to see text,
  <F1> to quit, or <ENTER> to restart
```

S SAVE BUFFER TO DISK

When I INPUT TO BUFFER has been turned on from the command mode and a file name specified, pressing S SAVE BUFFER TO DISK acts just like turning I INPUT TO BUFFER to OFF. It will save the buffer to the disk file, and print the message. However if a control code from the remote computer system has turned on input to buffer then pressing S SAVE BUFFER TO DISK will cause OMNITERM to ask you for a filename to save the buffer to, and only then will it save it.

Lowercase Keyboard

The LOWER/OMT file has been removed. The Model I did not have upper/lower case capability; only upper case. Model II has full upper/lower case capability, so the translation table is no longer required.

OMNITERM Utilities

The files BINHEX/CMD and HEXBIN/CMD described in the OMNITERM manual are supplied on your disk as the files BINHEX and HEXBIN. They work just as described in the OMNITERM manual.

The BINERR/CMD and ERRBIN/CMD files are not included in Model II OMNITERM.

There are two additional files supplied on your Model II OMNITERM disk that are not described in the OMNITERM manual.

The first, OMNICONV, is a file conversion utility that allows you to convert any file to another file with any other possible TRSDOS file format. When OMNITERM reads and writes files from and to disk, it treats them as files with Fixed length records with a file length of 1, just like the files produced by BASIC when you save a BASIC program with the ",A" option, or by SCRIPSIT. However to work with files of other types (different record lengths, variable length files, program files, etc.) you must convert them with OMNICONV. Simply type "OMNICONV" at the "TRSDOS READY" level and it will asK you what file you want to read, what file you want to write, and the file attributes you want the file to have. Note: If you are not having any trouble with file types, we suggest that you do not use OMNICONV at all. It will not be needed in most situations.

File Transfer Protocol

The other special file on your disk is called OMNIXFER, and it is a very powerful file transfer program. It is run by typing "OMNIXFER" at the "TRSDOS READY" level. You must have already set up the RS-232 port to 8 data bits with either the SETCOM command or with OMNITERM itself before running this program. From here you can send or receive files of unlimited size using Chistensen protocol. This is the same feature desribed in the OMNITERM manual under F FILE TRANSFER, execpt that it works from

"TRSDOS READY" level as a separate program, rather than being a part of OMNITERM itself.

Supplied Setting Files

There are no functional changes. Only one of each setting file is now required, since it is no longer necessary to differentiate between Models I and III. Other than that, the operation is the same as that in the manual.

There is an additional settings file supplied with OMNITERM for the Model II, BREAK/OMT. Normally pressing control-C or BREAK on the keyboard sends an ASCII 3, a control-C character that makes most remote computer systems stop what they are doing and pay attention to you. However some computer systems need a different signal, the true break, to accomplish this. Loading BREAK/OMT causes OMNITERM to send the true break when you press BREAK or control-C from the keyboard. This change is made by setting 03 (control-C) in the Special Command Table to 01 (the special command number for true break).

Appendix F - OMNITERM Speed Limits

OMNITERM can only handle communications up to a certain rate without falling behind and missing characters. Although Model II OMNITERM allows you to configure your UART for any baud rate up to 9600, this does not mean that OMNITERM can take data transmitted continuously at that rate. This means only that the interface hardware can take in the individual bits in each byte at that rate. OMNITERM needs a certain amount of time at the end of each line to scroll up the screen, and a lesser amount of time between each character. Only at high baud rates does this become a problem.

Model II OMNITERM uses the Model II's interrupt hardware, which effectively allows OMNITERM to run at its maximum average speed. Model II OMNITERM therefore can run at any speed up to and including 2400 baud, without any need for nulls or other pauses. In certain circumstances it can go as fast as 9600 baud, but this is not recommended for normal use.

Up to 2400 baud OMNITERM can do everything as described in this manual. At 4800 baud it only has enough time to do the basic functions of translating bytes and sending them to the correct devices. If you try to run the printer or do file I/O at a full 4800 baud with no pauses between characters, OMNITERM will not be able to keep up.

Beyond 4800 baud OMNITERM cannot keep up at all with full speed. Although you can configure your UART for 9600 baud, you should not run OMNITERM at these rates unless there is some wait time between characters. For instance, when you type on the keyboard, there is a long wait between characters and OMNITERM could keep up with this, or even something a great deal faster, at up to 9600 baud. Just do not try to run at this speed with no wait time between characters.

Appendix H -- Connecting an Amplifier for the Bell

This appendix no longer applies. There is no cassette port on the Model II, and no amplifier hookup is possible. Instead, the word "BELL" appears in reverse video in the upper left hand corner of the screen.

Notes on converting Model II/12 OMNITERM to TRSDOS II

Tuesday, November 15 1983

Model II/12 owners may use OMNITERM with either the original TRSDOS 2.0 or the hard disk operating system TRSDOS II. Since the OMNITERM program is delivered on a TRSDOS 2.0 compatible data disk, to use the program with TRSDOS II the file must be converted using the TRSDOS II utility FCOPY.

To convert the OMNITERM files to TRSDOS II, place your OMNITERM disk in a floppy disk drive and type:

FCOPY source TO destination {ALL}

where source is the disk drive you have the OMNITERM disk in, and destination is the disk drive you want the files copied onto, either another floppy disk drive or a hard disk.

For example on a system with two floppy disk drives, drive 0 and drive 1, you would put your TRSDOS II disk in drive 0 and boot it up, put your OMNITERM disk in drive 1, and type:

FCOPY 0 TO 1 {ALL}

For more details, refer the the FCOPY command in your TRSDOS II manual. If you have trouble, contact Radio Shack or Lindbergh Systems.

Omniterm

The smart terminal communications program for TRS-80 microcomputers.

User's Manual

Release 4

Lindbergh Systems

P.O. Box 3604
Ann Arbor, MI 48106
(313) 971-9733
CompuServe: 70310,267

READ THIS SOFTWARE LICENSE AND LIMITED WARRANTY BEFORE UNSEALING THE OMNITERM DISK. IF YOU DON'T AGREE WITH THE TERMS AND CONDITIONS OF THE *SOFTWARE LICENSE, RETURN THE PACKAGE TO THE SELLER WITH THE SEAL INTACT* AND YOUR MONEY WILL BE REFENDED.

SOFTWARE LICENSE AGREEMENT AND LIMITED WARRANTY

Lindbergh Systems, a Massachusetts corporation, (hereinafter "we") grants to the purchaser (hereinafter "you") a non-exclusive license to use or transfer the serialized, copyrighted OMNITERM program (hereinafter "the program") and to use the copyrighted USER'S MANUAL under the following terms and conditions.

License: You acknowledge and agree to the terms and conditions of this software license when you open the sealed disk.

We reserve all rights to the copyrighted OMNITERM program and USER'S MANUAL. You may not reproduce any part of the USER'S MANUAL.

You may not use, copy, modify, or transfer the program, except:

a. you may load the program into a single machine for use only on the single machine;

b. you may make copies of the program for use on the single machine;

c. you may transfer the program, copies, and the USER'S MANUAL to another party (herein also "you" or "the recipient") provided the recipient agrees to the terms and conditions of this license;

d. if you transfer the program, you must transfer the USER'S MANUAL, you must destroy all copies not transferred, and you many not subsequently use the program on any machine;

e. if you transfer the program, the recipient may not use, copy, or transfer the program except as provided in this license under a, b, c, d, and e.

Limited Warranty: We warrant that the disk on which the program is delivered accurately reproduces the copyrighted OMNITERM program and will be free from defects in material and workmanship under normal use for a period of 90 days after the date of first retail purchase of the program.

Replacement is the sole remedy for defective media. We shall not be responsible for any information stored on any media which are returned.

We express no other warranties. We disclaim liability for incidental or consequential damages arising out of the use or inability to use the program and all implied warranties including all warranties of merchantability and fitness for a particular use.

Some states do not allow the exclusion of incidental or consequential damages nor limitation on how long an expressed warranty lasts, so the above exclusion and limitations may not apply to you.

This warranty gives you specific legal rights and you may have other rights which vary from state to state.

OMNITERM was orignally written by David J. Lindbergh. Version 1.60 and later were modified by Donna LaMadeleine. This manual was written by David J. Lindbergh.

Table of Contents

Introduction

OMNITERM is an extremely flexible and powerful terminal program for the TRS-80 microcomputer. It will allow you to use your TRS-80 to communicate with almost any type of computer hardware with speed and efficiency. OMNITERM allows your TRS-80 to automatically do all the translation necessary when connected to other computers, as well as giving you a wealth of local functions that ease your use of the system. With OMNITERM you get a true Break key, cassette port "beep" and graphic bell on control-G, a repeat key, buffered printer, the ability to see text that has scrolled off the top of the screen, one key auto signon, lowercase support, file transfer including prompted output and XON/XOFF code recognition, direct X/Y cursor addressing and screen control functions, error count displays, XMODEM file transfer, full status display of all functions and much more.

We recommend that you read through this manual once quickly, then sit down with the computer and use each of OMNITERM's functions as they are described. This way, you will have an idea of what is going on as you read through the manual a second time.

If you encounter any unfamiliar terms or concepts, it is a good idea to refer to the Glossary in the back of this manual, rather than plowing through a description of something using words you don't understand. This manual assumes a certain level of technical sophistication, so if you are new to computers or to data communications terminology you may find yourself confused. The Glossary is there to help you along.

If you have trouble using OMNITERM please check the troubleshooting guide in Appendix M before you call us. All the common problems that people have are covered there.

Any keys on the TRS-80 keyboard other than single letter keys will be designated in this text by enclosing them in greater/less brackets. For example, <BREAK>. Other keys will simply be stated in capital letters with no other marking, such as the A key or the Q key. It is often important to distinguish the letter O from the number zero. If you are at any time confused, note that the letter is printed as O and zero as 0.

System Requirements

OMNITERM runs on a TRS-80 Model I or III computer with a minimum of 32K of RAM and at least one disk drive. It also requires either Radio Shack's RS-232C interface, or a Microconnection "Port 208" type modem as the hardware over which it communicates. OMNITERM will also work with any hardware that is software compatible with the above, such as the Lynx modem which is compatible with the Radio Shack RS-232 board.

OMNITERM is designed to work with any TRSDOS compatible operating system, from TRSDOS 2.0 onward. Since OMNITERM uses only standard DOS calls, it should work with all future operating systems as well. To date, OMNITERM has been tested on TRSDOS 2.0, 2.1, 2.2, 2.3, 3.0 for the Model I, NEWDOS 2.1, NEWDOS-80 Versions 1 and 2, VTOS 3.0, 4.0, LDOS 5.0 and 5.1, DBLDOS, DOSPLUS, ULTRADOS, and TRSDOS 1.1, 1.2, and 1.3 on the TRS-80 Model III. OMNITERM will work correctly with all of these DOSes, although with some of them certain special settings are needed, as described below.

In this manual, you will sometimes see a reference to the <BREAK> key used to leave one mode and go into another. Under some DOSes, notably LDOS or Model III TRSDOS, you may use <BREAK> to stop what you are doing at ANY TIME at all, not just when it is mentioned in the text. If you do not have one of these systems, the <BREAK> key will perform as described.

Getting Started

Model I

The OMNITERM disk as delivered will run on the Model I TRS-80. To run OMNITERM, boot in the disk supplied. When the "DOS READY" prompt appears, type "OMNITERM" and hit the <ENTER> key. This will load OMNITERM and start it running.

Model III

OMNITERM and its utilities are compatible with the TRS-80 Model III computer, as well as the Model I. Since the disk is delivered in Model I single-density form, the disk must first be converted to double density using Model III TRSDOS's CONVERT utility.

You will need at least two disk drives to run CONVERT. If you have only one, you will have to borrow a Model III computer with two drives to do this conversion. Your local Radio Shack should be able to help you here. Once OMNITERM has been converted, you will need only one drive to use it.

When you first get your OMNITERM disk, you should make up a backup of your original Model III TRSDOS disk. This will become your master OMNITERM disk. Boot up your new backup TRSDOS disk and run the CONVERT program by typing CONVERT in response to TRSDOS READY. The program will ask you which drive will be the source drive. Place the original OMNITERM disk in drive 1, and answer the question with 1. CONVERT will then ask you for the destination drive. Type in 0, since your TRSDOS disk is the destination drive. The CONVERT program will then begin copying the files on the original OMNITERM disk to your TRSDOS disk. As it does this it will list the files on the screen one by one. When the CONVERT program finds a file on the OMNITERM disk that is already on your TRSDOS disk, it will stop and ask you if it should convert it. It should stop on files such as FORMAT/CMD, BACKUP/CMD, and BASIC/CMD. You should answer NO to this question, because these are Model I TRSDOS programs that will not work with your Model III. Your Model III TRSDOS already has programs which do the same things as these programs, so you do not need them.

When all the files have been converted to your Model III TRSDOS disk, CONVERT will stop and return you to TRSDOS READY. You should mark your TRSDOS disk "OMNITERM Master Disk - Copyright 1980-83 Lindbergh Systems, Inc.", and then back it up. This is your master disk, and you should keep it in a safe place, and only use it to make other disks from. You have now converted the OMNITERM package to double density, and are ready to use it. To run OMNITERM, boot in the disk. When the "TRSDOS READY" prompt appears, type "OMNITERM" and hit the <ENTER> key. This will load OMNITERM and start it running.

Once OMNITERM is running on the Model III, there should only be two differences in operation against the Model I. First, since the Model III RS232 board has no sense switches, OMNITERM cannot read them for default settings. Instead, it will use a standard setting of 300 baud, 8 data bits, 1 stop bit and no parity whenever the sense switches would be read on the Model I. This setting is the one most commonly used for communicating with timesharing computer systems.

The second difference is that the Model III never needs a lowercase driver, since this is the normal mode of the Model III. Ignore all references to the Model I lowercase driver if you have a Model III.

Some special modems for the Model III, such as the Lynx, do not require use of the Model III RS-232 board because they have their own software-compatible interfaces that work through the I/O Bus connector on the bottom of the Model III. OMNITERM automatically turns on the Model III I/O bus, so these special modems will work with OMNITERM without the user needing to manually control the I/O bus.

More will be said about these items as this manual goes along, and these items will be clarified if they are not clear at this point.

NEWDOS-80

OMNITERM is compatible with NEWDOS-80, just as it is with all TRSDOS-like TRS-80 operating systems. However, for OMNITERM to operate correctly with NEWDOS-80, the <BREAK> key must be enabled within NEWDOS-80 by setting the NEWDOS-80 system parameter AG equal to Y. This is true for both Version 1 and 2 of NEWDOS-80. As described on page 2-22 of the NEWDOS-80 Version 1 manual:

> "If ag=Y, BREAK is considered a normal input key with code = 01. If ag=N, the BREAK key is not considered a normal input key and its occurrence is changed to the key code 00. The state of the BREAK key is set according to option AG at reset and then again everytime the system returns to normal DOS READY."

DOSPLUS

DOSPLUS users should note that the default DOSPLUS printer driver will not work correctly with OMNITERM unless a FORMS (P=66,L=66) command is executed before using OMNITERM. This is because the DOSPLUS printer driver does not correctly indicate when the printer is ready. The exact details of what is required of user or DOS-added printer drivers is explained in the appendix.

LDOS

OMNITERM works normally with LDOS on the Model I, but to use LDOS version 5.1.3 and later on the Model III you must install the following patch to the OMNITERM program:

```
. OMNILDOS/FIX
.
. Patches to OMNITERM/CMD, version 1.60
. For use with LDOS 5.1.3
.
. PATCH OMNITERM/CMD USING OMNILDOS/FIX
.
. Change RS-232 receive interrupt vectors
X'6040'= 7F 44
X'63A4'= 7F 44
X'72A5'= 7F 44
X'7FA7'= 7F 44
. NOP out check for printer driver
X'604D'= 00 00
. End of patches
```

Be sure you only apply this patch if you are using OMNITERM 1.60 with LDOS on a Model III TRS-80. To apply it, enter the patch and save it in an ASCII text file called OMNILDOS/FIX. This can be done with a word processor such as SCRIPSIT, or with another text editor such as TEXTED/BAS included with OMNITERM. The only lines you must type in are the five lines that start with the letter X. The other lines are only comments and are optional. Then, at LDOS command level ("LDOS READY") type:

```
PATCH OMNITERM/CMD USING OMNILDOS/FIX
```

For more details, refer to the PATCH command in your LDOS manual. If you have trouble installing the patch, call Lindbergh Systems.

LNW-80 Computer

The LNW-80 computer is a TRS-80 Model I "lookalike" computer that is supposed to run all Model I software. To use OMNITERM with the LNW-80 you must install this patch to OMNITERM:

```
PATCH OMNITERM/CMD (ADD=6364, FIND=DBD1, CHG=1808)
```

If you have trouble installing this patch, please call Lindbergh Systems.

```
**********************************************
*                                            *
*              !!! IMPORTANT !!!             *
*                                            *
*    BEFORE YOU USE OMNITERM, BE SURE YOU    *
*       ----------> BACKUP <----------       *
*    YOUR OMNITERM DISK FIRST!!  YOU WILL    *
*    BE VERY SORRY IF YOU MAKE  A MISTAKE    *
*    AND DESTROY YOUR ONLY COPY!  SO MAKE    *
*    AT LEAST TWO BACKUPS OF OMNITERM AND    *
*    PUT THEM IN A SAFE PLACE!               *
*                                            *
**********************************************
```

Using OMNITERM

To run OMNITERM, put your copy of OMNITERM in drive 0 and reboot the system. At the DOS READY prompt type OMNITERM, then press the <ENTER> key. OMNITERM will clear the screen and print its introductory message and copyright notice at the top of the screen. You are now in OMNITERM's normal mode ready to communicate. If you want to send characters, just type them and they will be sent out. If characters are received, they will be shown on your screen. You must be connected to another computer to see the characters you type unless you change the DUPLEX setting to HALF.

Default Settings

When run on a Model I, OMNITERM reads the RS-232 board's switch settings and sets the UART to match these settings. It also checks to see if you have Radio Shack type lowercase hardware and, if so, uses a lowercase driver.

The Model III communications board does not use sense switches, so OMNITERM is set up to use 300 baud, 8 data bits, 1 stop bit, and no parity as default settings. Of course, these can be easily changed using OMNITERM's Command Mode. But these settings will work with most remote systems.

Special Keys

On the Model I:
The <SHIFT> key is used to send characters in lower rather than upper case.

On the Models I and III:
The <DOWN ARROW> can be used as a CONTROL key to send control characters.

The <UP ARROW> can be used to send an ESCAPE (also known as ALT mode.)

The <CLEAR> key works as a repeat key. To send a series of the same character, press the key you want to send with <CLEAR>.

The <BREAK> key will send a true break.

Since OMNITERM uses the standard keyboard device, special keyboard functions available with differect operating systems WILL work with OMNITERM. For example, some operating systems allow you to toggle from an UPPER/lower case keyboard to a lower/UPPER case keyboard by pressing <SHIFT>-0. If you have this feature, it will work fine.

While OMNITERM boots up ready to communicate with many common systems, the power of the program is found in its ability to configure your TRS-80 to exactly match whatever settings the remote systems requires. This is done using the options available from the Command Mode Menu. Each option in the Command Mode Menu is described in detail in the rest of this manual. To get you started, the following examples will show you how to use these options to accomplish some of the most basic tasks.

An Example of Contacting an Information Service or Mainframe Computer.

Because computer communications are so diverse, it is impossible to give a real example that will work for a large audience. Phone numbers, passwords, logons will be different for each individual. Yet the easiest way to learn is by doing. So we have invented a fictitious information service to illustrate how you might set up to communicate with a remote system.

Whenever you communicate with another computer, there are a number of steps you must follow to establish communications.

Typically, you must:

1) Dial the telephone to access a network.
2) Connect to a system through the network.
3) Identify yourself to the system.
4) Verify your identification with a password.

Dialing depends on your modem. Our example assumes that you are using a Hayes Smartmodem(tm) for communications so that you can set up OMNITERM to automatically dial for you. If you are using an "acoustic coupler" modem, you must dial the telephone yourself, wait for the "carrier" tone, and place the telephone handset in the acoustic cradle. See your modem instructions for the specifics about dialing with your modem.

Networks, such as Tymnet(tm) and Telenet(tm), provide the communications so that you can use a local call. Our example below uses a fictitious network named Commnet with a local telephone number 555-1234. You must know what telephone number to dial for the computer you wish to contact.

Systems are typically identified by a series of letters and numbers. The Source(tm) and Dow Jones News Service(tm) are examples of services you may be contacting. Our example uses a fictional data service named InfoServe. You must know how to get to the system you wish to contact through the network you will be using. If you are contacting a local service, such as a Computerized Bulletin Board Service, there will be no network involved and you simply skip this step.

User identifications and passwords depend on the system that you are using. Each one has its own rules. Our example uses TXH1138 (I.D.) and LETMEIN (password). "Public" services do not require a password, but usually request your name for identification.

Dialing and Logging On

This model is a step by step example of how to manually log on to our fictitious information service. Read through it once without your computer to get an idea of the process. Then go through the example and substitute the real phone number, modem commands, prompts, system name, your I.D. and password to log on to the system you want to contact.

1) Load the OMNITERM program.

2) You will be working in normal mode.

3) Type **ATDP 555-1234** and press <ENTER> to dial the phone.
This is for the Hayes Smartmodem. Your modem may require some other method of dialing. Depending on the modem and the switch settings, you may not see the typed characters echoed on the screen.

4) Wait until "COMMNET" is on the screen.
Your actual response probably will be TELENET, TYMNET, or the "sign on" message of a local system.

5) Press <ENTER> twice.
This is almost universally required. Sometimes only one <ENTER> is needed. In some unusual situations, you may be required to use a special code to identify the baud rate or parity you are using.

6) Wait for the prompt "SYSTEM I.D.?"
From this step on, the actual messages and codes will depend entirely on the network and system.

7) Enter **INFOSERVE** to connect to the system.
This might be CIS if you were calling CompuServe(tm).

8) Wait for the prompt "User I.D. >"

9) Enter **THX1138.**
Your User I.D. is usually given to you by the Service or System when you obtain an account.

10) Wait for the prompt "Password >"

11) Enter **LETMEIN** to verify your identity.
This password you usually choose yourself when you first get the account on the system.

12) When you see the prompt ">" it indicates that the system is ready.

As soon as you have a system to contact (including the telephone number, codes, and instructions for contacting the system) you can use your OMNITERM program. Chances are that you will not have to change any of the OMNITERM settings.

For this "simple case" you will never need to use any of the OMNITERM menus except to end the program with the Q for QUIT option from the System Command menu.

It is likely that you will soon find these "signon" steps tedious. If you have an autodial modem, you don't have to go through them every time. You can let the OMNITERM auto signon functions do them for you. The next section will show you how to "automate" these steps. Complete details can be found under the system command, AUTO SIGNON & AUTO DIAL.

Setting Up an Automatic Signon

The steps below show how to create the automatic signon sequence for our fictitious system in the last section. You will have to change the actual codes and steps to match the real sequence for your system.

REMEMBER THAT THE KEYS TO SUCCESSFULLY AUTOMATING YOUR SIGN ON ARE ACCURACY AND TIMING. YOU MUST ENTER THE SIGNON EXACTLY THE WAY YOU TYPE IT MANUALLY. YOU SHOULD INCLUDE LOTS OF PAUSES.

Follow these steps to create the automatic signon:

1) Press the @ key twice to go to the Command Mode Menu.

2) Select X, System Commands.

3) Select the A AUTO SIGNON & AUTO DIAL option and press A again for Auto-Signon string change. A prompt will tell you how you can include pauses and control characters in your string.

4) Type **ATDT** (the Hayes dial command) and then **555-1234**.

5) The phone number must be followed by a carriage return because the Hayes modem requires this to end a command. But if you press <ENTER>, the program will interpret it as end of input. So, the way to include a carriage return in your logon sequence is to press the <DOWN-ARROW>, 13 (the decimal code for carriage return), and another <DOWN-ARROW>.

6) Now you are ready to begin the logon sequence. Here is where the pauses are most important. Press the <DOWN-ARROW>. Then type several letter Ps. Press the <DOWN-ARROW> key again. This will make OMNITERM wait several seconds before sending the signon. This assures that the modem has successfully dialed and connected to the remote computer before the signon is sent.

7) Press <DOWN-ARROW> 13 <DOWN-ARROW>.
Press <DOWN-ARROW> P <DOWN-ARROW>.
Press <DOWN-ARROW> 13 <DOWN-ARROW>.
This will send the remote system the two carriage returns.

8) Type **INFOSERVE** for the response to the network query for the name of the service you want to reach. Then press <DOWN-ARROW> 13 <DOWN-ARROW> to send a carriage return. You will probably need some more pauses after the CR.

9) Type **THX1138** and press <DOWN-ARROW> 13 <DOWN-ARROW> for the I.D.

10) Type **LETMEIN** and press <DOWN-ARROW> 13 <DOWN-ARROW> for the password.

11) Press <ENTER> to end the logon entry.

The logon should look like this:

ATDT555-1234[13][PPPPPPPPPP][13][P][13][PPPPP]INFOSERVE[13]
[PPP]THX1138[13][PPP]LETMEIN

You are now ready to automatically dial and sign on to InfoServe. If this were a real signon, you would press BREAK three times to return you to normal mode and press <DOWN-ARROW>-0 (zero) to automatically dial and sign on.

Try setting up a real auto signon with your own sequence of commands. Write down each step you take to sign on manually and then type those steps into a string. Remember to use lots of pauses especially after the telephone number and the two <ENTER> keys required by the remote system. More details on setting up an automatic signon are given under the description of the system commands.

Saving your settings

Now that you have set up OMNITERM to automatically dial a remote system, you will want to save the settings for future use. OMNITERM does not just save telephone numbers. You create a settings file tailored for that particular system. It will include all the UART settings, any changes in the translation tables, and all other OMNITERM settings including the auto signon string. It can even begin automatically dialing as soon as the program is started. Here is what you do:

1) From the System Commands menu select S Save OMNITERM Settings.

2) The program will ask whether the settings file should do Auto-Start. If you answer "Y", then the settings file will automatically dial and signon whenever you load it.

3) When the prompt appears asking for a filename, enter "INFOSERV". (Of course, you use a name that describes the service you are actually communicating with.)

You now have created a settings file called INFOSERV.OMT. (Since you did not add an extension, OMNITERM adds .OMT automatically.) Any time you want to use these settings, you can either load them from the System Commands menu by pressing L, or if you are just starting up OMNITERM from DOS, enter OMNITERM (space) INFOSERVE. In either case, the settings file will be loaded and will begin to automatically dial and sign on to the system. For complete details on using OMNITERM settings files, see the discussion of S SAVE OMNITERM SETTINGS, below.

You will probably want to create settings files for each system you contact. This is the easiest and fastest way to assure that you have the correct settings. You may, for example, want to set up one file for CompuServe at 300 baud and another file for the Source using a 1200 baud modem. You can set up as many settings files as your disk will hold.

Connecting to Another Microcomputer

OMNITERM can be set up to communicate with another microcomputer very easily. You don't need OMNITERM on the other end, but you do need some kind of communications software on the other end, along with an RS-232 port.

Micro-to-micro communications require one end to act like a "host" computer, and the other like a terminal. OMNITERM can do either. To use OMNITERM as the terminal, follow the same procedures just described, Contacting an Information Service or Mainframe Computer.

Since some personal computers can only be set up to act like terminals, here's how you set up OMNITERM to make your TRS-80 act like the host:

1) In the OMNITERM Command Mode menu, toggle E ECHO to ON. This will make OMNITERM echo back to the other computer the characters that the other computer sends.

2) In the same menu, toggle G CR/LF GROUPING to ON. This will guarantee that carriage returns and line feeds are sent properly to the remote computer.

3) Put OMNITERM in HALF Duplex so that what you send out to the remote computer will be echoed by OMNITERM to your screen.

4) Set both computers to the same baud rate, data bits, stop bits, and parity. Use OMNITERM's defaults unless the other system requires something else, or you want to go faster than 300 baud. You can change these in OMNITERM's U Change UART Settings option.

5) If you are using modems, set one to "originate" and the other to "answer". It doesn't matter which one is which, as long as they are different.

6) If the computers are near one another and you have them directly connected without modems or telephones, you need a special device called a "null modem adapter". This is available in most computer stores.

Once you are hooked up as described, both of you can type on your keyboards and both ends will see it. This is called "chatting," and you should do it to make sure that everything is working OK.

You can send files back and forth with the other microcomputer using terminal mode, or Christensen protocol if the other computer also has Christensen protocol.

Transferring files

Most of the data you want to send and receive will be text. The most common way to do this is as a terminal, character by character. This is done in normal mode as though you were typing the file from your keyboard.

You might also wish to send or receive very large files at high speeds using an error correction protocol. Using the popular Christensen (also known as XMODEM) protocol, you can transfer text or non-text files such as a program. This is done with the F FILE TRANSFER command which is described in detail under System Commands.

Sending files -- Normal Mode

Most remote computer systems have a function called an "editor" where you can type a file into the remote computer from your keyboard. If you are communicating with another microcomputer you can probably do the same thing by having the other microcomputer save everything it receives to disk.

To send a file you simply prepare the remote system to save what you type in, and then have OMNITERM "type" it for you— only much more quickly than you could.

1) Get the remote system ready to accept a file, as if you were going to type it in one line at a time.

2) Press F, for the FILL BUFFER FROM DISK function from the Command Mode Menu.

3) Enter the name of the file you want to send. OMNITERM will load the file into its buffer.

4) Press O, for the OUTPUT FROM BUFFER function. OMNITERM asks you three questions about how you want it to slow down when sending. These questions are explained later so you can ignore them for now and press <ENTER> after each one.

5) After you go through each question, you can press <BREAK> to return to normal mode. OMNITERM will send the contents of the buffer to the remote system. You can then tell the remote computer system that "you are done typing," if it hasn't figured that out for itself.

Saving files -- Normal Mode

The idea is to get the remote system to show the file to you on the screen as if you were reading it, and have OMNITERM save it in its buffer as the remote system lists it. You turn I INPUT TO BUFFER on, get the remote system to list the file, then turn INPUT TO BUFFER off. The S SAVE BUFFER TO DISK command is then used to save the contents of the buffer to your disk.

1) Get the remote system ready to send you the file. Usually you do this by giving the remote system a command line that ends with the <ENTER> key. So the best tactic is to type out the whole command line, but don't press <ENTER> yet.

2) Now press the @ key twice to return to the Command Mode menu.

3) Press I, INPUT TO BUFFER in the Command Mode Menu to turn the function on.

4) Press the <BREAK> key to return to normal mode. From this point on, all data that enters your computer will also go to the buffer, until the buffer is full.

5) Make the remote system send you the file. If you set it up as described in step 1, just press <ENTER>. As the file comes in, OMNITERM will be storing it in the buffer.

6) When the whole file has been received, press the @ key twice to return to the Command Mode menu and turn INPUT TO BUFFER off by pressing I.

7) You can now save the contents of the buffer using the S SAVE BUFFER TO DISK command or use System Commands to examine the buffer or re-open it to capture more data. (The buffer is automatically cleared when you press I INPUT TO BUFFER once it has been turned off.)

These steps are all described in detail under each command. Our purpose here is to give an overview of how you use various commands to accomplish your task.

With this overview, you have been exposed to the most commonly used features of the OMNITERM program. The rest of this manual describes in detail each of the commands and powerful features of the program. If you are having problems with a specific feature, you should read the details under the command or refer to the troubleshooting guide, Appendix M.

OMNITERM Command Mode

To get at the real power of OMNITERM, you will want to go into OMNITERM command mode. To do this, just press the @ key twice. If you press @ once, nothing will happen. It is only when you press @ a second time, with no other keys between them, that you enter command mode.

When you want to leave command mode, just press the <BREAK> key and you will return to normal mode, with the screen just as you left it before.

When you enter command mode, the information on the screen will be replaced with a large menu. What had been on the screen is not lost. When you return to normal mode later, it will reappear so you can see where you were.

The command mode screen you see contains a status display of the major functions of OMNITERM and a summary of the commands available to you in the form known as a menu.

On the bottom left of the screen you see the current UART settings in use. This includes BAUD RATE, number of DATA BITS, number of STOP BITS and PARITY. This is a quick reference telling you how your hardware is configured.

The bottom middle of the screen shows the status of the most important RS-232 input lines. Beside the wires for transmitting and receiving data, there are extra control lines on the RS-232 interface. These lines fall into two types, input lines which report information from the modem or other device connected to your computer, and output lines that affect the device and are set by the TRS-80. At the bottom middle of the command mode screen is a display of the state of the four important input lines, CTS or Clear-To-Send, DSR or Data-Set-Ready, CD or Carrier Detect, and RI or Ring Indicator. Each time you enter the command mode and the command mode screen is printed these lines are read by OMNITERM and printed on the screen as ON or OFF. OMNITERM does not require these lines to be set any particular way (except optionally in the case of CTS, described later) but this can be useful information if your modem or other RS-232 device sets these lines to indicate various conditions. If your RS-232 hardware does not use these lines, then simply ignore them.

The bottom right shows the number of errors of various types that have occured. This will help you in determining the quality of your connection to the other equipment and will give you a hint when something is going wrong. These counts are set to zero when you start up OMNITERM. Below this is a summary of your disk I/O buffer. It shows how many characters are currently stored in it, and how many it can hold in total.

These bottom portions of the screen may be overwritten by some commands that need space to print extra messages. For instance, when saving a file to disk, you will type the file name in this area.

The top portion of the screen has two columns, listing the commands available to you in command mode. The left column shows a single letter, a single-line summary of that command, the word "is:" and a status flag showing you the

state of that OMNITERM function. For instance, the top line on the left shows:

P PRINTER is: OFF

This tells you that the printer is currently off. This will be discussed further in a moment. The commands on the left are invoked by typing the appropriate letter. It is not necessary to type <ENTER>. For example, if you press the P key, you will invoke the PRINTER command.

All the commands on the left column affect the status flags in one way or another. The commands on the right column do not directly affect any visible status flags, but otherwise work the same as the left column functions. When you are in command mode, all you have to do to use any function or invoke any command (a "command" and a "function" in OMNITERM are the same thing) is press the key associated with that function. In some cases, this is all that needs to be done, because there is only one possible way to do that command and it is done immediately. In other cases, OMNITERM will ask you for more information, such as a value or filename, before executing the command.

Summary of Command Mode Functions

Key	Command Name	Function
P	PRINTER	Turn printer on and off
R	SCREEN REFORMATTING	Stop words splitting over edge
C	CR SUPPRESSION	Stretch lines to full screen size
L	LF SUPPRESSION	Prevent double spacing of lines
D	DUPLEX	Send typed characters to screen
E	ECHO	Retransmit all received data
G	CR/LF GROUPING	Send LF after every CR
I	INPUT TO BUFFER	Save received data in buffer
O	OUTPUT FROM BUFFER	Send buffered data to interface
X	SYSTEM COMMANDS	Menu of less often used commands
T	CHANGE/EXAMINE TABLES	Review and modify OMNITERM tables
U	CHANGE UART SETTINGS	Reconfigure interface hardware
A	SEND CONTROL-A & QUIT	Trasmit ASCII value of 1 & leave
@	SEND "AT" SYMBOL & QUIT	Transmit @ sign, go to normal mode
B	SCROLL BACK DISPLAY	See text that scrolled off screen
Z	ZERO REAL TIME CLOCK	Set system timer to 00:00:00
F	FILL BUFFER FROM DISK	Load a file into the buffer
S	SAVE BUFFER TO DISK	Write buffer contents to a file

Command Mode Functions - Left Column

First we will cover the command mode functions down the left hand column of the command mode screen. These all have status flags associated with them that show the current state of each function. You can use these to verify your function settings and any time you want to check on the status of any OMNITERM function you can flip into command mode and check these status flags. From top to bottom, the left hand column functions are:

P PRINTER

OMNITERM allows you to run your line printer at the same time you are communicating over the serial interface, so that you can have a written record of your session. To turn on the printer, just press the P key from command mode and the printer will go ON if it was off. If you want to turn it off again, just press the P key again. You will notice that as you turn it on and off, the status flag for the printer goes from ON to OFF. This is to let you know at all times if your printer is on or not. The default setting for the printer is OFF.

OMNITERM allows the printer to fall behind the screen if it is not fast enough to keep up. OMNITERM can buffer up to 2048 characters of data to the printer before it runs out of room, thus you can use a printer that is slower than your communications. If the printer falls so far behind that OMNITERM's buffer space is filled, you will see the message:

```
*** ERROR --> PRINTER BUFFER FULL
*** PRINTER TURNED OFF
```

The printer will continue to print what was in the print buffer until it is empty, but no new data will go into the print buffer. This will not affect the normal flow of data to the screen or any other function of OMNITERM. If you go to the command mode, you will notice that the printer has been turned OFF. If you wish, you can turn it on again at this point.

R SCREEN REFORMATTING

One of the problems that must be dealt with when using many different computer systems with your TRS-80 is the fact that most of them are not set up for the TRS-80 64 column wide screen. Many are set up for the 80-column standard, others for 40-column Apple or Atari computers and still others for the 32-column VIDEOTEX type units. OMNITERM meets this problem by allowing you to re-format your screen locally in your TRS-80, so that what you see is a nice 64 column style screen, with all lines filled out and no words split halfway over a line.

The SCREEN REFORMATTING function prevents words from being split halfway over a line by checking for spaces and when the cursor gets too close to the end of a line (where the next word might be split over) it displays the space

as a carriage return and begins on the next line. OMNITERM will insert this carriage return only on the screen, the printer and disk output will remain just as the data was originally received.

When you press R and invoke the SCREEN REFORMATTING function, you can re-define which screen column OMNITERM will consider to be too close to the edge of the screen. As you press the key, OMNITERM will clear the bottom of the screen and ask you to:

```
Enter new Screen Reformat value -->
```

You simply type in the new value and press <ENTER>. The value may be from 0 to 255, or you may enter OFF. If you enter 0 or OFF, the status display for this function will show "OFF" and screen reformatting will be shut off. If this function is OFF, words will be split whenever they begin too close to the right edge of the screen. You will normally turn this function OFF only when communicating with other TRS-80's or other machines meant for 64-column wide use, such as some bulletin board systems. If you enter a value greater than 255, it will be ignored and the value will not change.

You will notice that the default value for this function is column 54. This would mean that a space (between words) received AFTER COLUMN 54 will be sent to the screen as a carriage return. We have found that 54 is about right for most uses. If you want to allow longer lines, but risk having lines split halfway over the edge (at column 64) you could set a higher reformat value, such as 58 or 60. If you want to be absolutely sure no words get split, set it at 50 or so. Here only words 14 characters or longer will be split, since 50 is 14 less than 64.

C CR SUPPRESSION

This is the other half of OMNITERM's ability to re-format your screen when working with various systems. Imagine that you are using a system that is sending 80-column lines. The R reformatting is preventing the lines from being split by changing some spaces to carriage returns in each line. That same line then continues on the next line of your screen. After two or three more words on your next line, the other system has finished its line, so it sends a carriage return. If OMNITERM processes this carriage return normally, you will see a full line of text followed by one with only two or three words on it. The next line will be full and the one after that will again have only a few words. This looks very sloppy. Here is where the CR SUPPRESSION command comes in.

When CR SUPPRESSION is ON (the default is OFF, explained later) every time a Carriage Return (CR for short) is received, it is translated to a space before going to the screen. This means when that two or three word line is done, the carriage return is shown as a space and the next line sent is shown as a continuation of the previous line. The only exception to this is when two or more CRs are received in a row. Then only the first one is converted to a space, so that a series will properly leave some blank lines on your screen.

Thus, when the cursor nears the end of the line, the R screen reformatting changes one of the spaces between words into a carriage return and your next line starts. This may all seem complicated, but it works rather well. For instance, say some 80-column lines are received with CR SUPPRESSION off:

```
Sparky Snap Crackers are the snappiest snap crackers
you can buy
without the dangerous addition of gunpowder in the
dough.  That's because
they taste soooo good!  You should get some.  Your
children will love
you for it.  No artificial flavorings added.  USDA
approved.  FDA approved.
```

But, with CR SUPPRESSION on:

```
Sparky Snap Crackers are the snappiest snap crackers
you can buy without the dangerous addition of gunpowder
in the dough.  That's because they taste soooo good!
you should get some.  Your children will love you for
it.  No artificial flavorings added.  USDA approved.
FDA approved.
```

That is how those same four lines would look with CR SUPPRESSION turned ON. The large lines have been re-formatted for your screen. This works equally well for expanding short 40 or 32 column lines to your screen size. The carriage returns at the end of each line become spaces and the line continues until it reaches the R value's setting, where the next space does a CR.

As you can see, the CR suppression is not much good without the R reformatting feature. Also, the CR suppression is really only useful when you are looking at large blocks of text. If you are seeing a menu on CompuServe, for instance, which is meant for 32 columns and looks like this:

```
1 Startrek
2 Space War
3 Checkers
4 Chess
5 Hangman
6 Poker
7 Civil War

Choose a number.
!
```

It does you very little good to see that as:

```
1 Startrek 2 Space War 3 Checkers 4 Chess 5 Hangman
6 Poker 7 Civil War Choose a number. !
```

Which is exactly how it would look with CR SUPPRESSION turned ON. For this reason, the default value for this is OFF.

L LF SUPPRESSION

Most computer systems are designed to use the ASCII code which was designed to run Teletype-style mechanisms. Teletypes need two separate codes at the end of each line- first, a CR to move the carriage back to the beginning of the line and then a line feed (LF for short) to move the paper up. The TRS-80 needs only one code to do both these functions and the way the screen driver was written, either of these codes will do both functions. Most computer systems send both a CR and a LF at the end of a line. If OMNITERM were to respond to both, you would get a blank line between each line on your screen. Thus the LF SUPPRESSION function allows OMNITERM to ignore line feeds and only accept the CR'S, so that you will not get a blank line. If you receive more than two LFs in a row, all but the first WILL be accepted, so your screen will scroll when blank lines are sent. If LF SUPPRESSION is turned OFF, all LFs received will be shown normally.

This function must be used with almost every commercial timesharing computer, since even non-Teletype video terminals now act like Teletypes. The default for this function is ON and the only time you are likely to want to change it is when communicating with other microcomputers that use only the CR.

D DUPLEX

The way most timesharing computer systems work is called "full duplex". This means that when you, the user, send a key to the remote computer, you do not see it on your screen immediately. Instead, the character is sent to the far end of the line, processed by the remote computer and only then sent back to you to appear on your screen. This process of sending back characters just received is called echoing back the characters. The main advantage in this is that if there was a transmission error somewhere along the way, you will know about it, since the wrong character will appear on your screen.

The other possibility is called "half duplex". This means when you press a key it is immediately shown on your screen, as well as being sent down the line. The remote computer does not echo it back at all and you must simply assume that it was received correctly. This is common with IBM mainframes. If you are communicating with another microcomputer, it will probably not echo back to you either so you will want to set OMNITERM to HALF (see Connecting to another Microcomputer.)

The default mode for OMNITERM's DUPLEX function is FULL. This means simply that OMNITERM will not send your keystrokes to the screen, but will only show what is received over the interface.

NOTE: Half and Full duplex have a different traditional meaning than that used in this context. There is a long story about how this came about, but if you ever hear "Half duplex" to mean "sending one direction at a time" and "Full duplex" to mean "sending both directions at one time", do not be surprised, since this is the original meaning.

E ECHO

The ECHO function is used when your TRS-80 is the computer that must supply the echo to another computer or terminal. When the ECHO function on ON, every character OMNITERM receives is immediately sent back out again. This will allow your TRS-80 to communicate with computers or terminals that can only connect to machines that echo back their data to them. Since most users of OMNITERM will be connecting to a timesharing system that echos to them, ECHO is not needed and the default is OFF. If you are setting up to communicate with another microcomputer, you will probably want to turn the ECHO function ON.

G CR/LF GROUPING

Turn the CR/LF GROUPING feature ON when you want to use OMNITERM to communicate with another microcomputer or terminal. When the CR/LF GROUPING function is ON, a line feed is automatically sent out along with each carriage return when you press the <ENTER> key. This is normally used together with the ECHO function, so that your TRS-80 acts like the "host" computer for a device like a Teletype, video terminal, or another microcomputer. The default state for this function is OFF, since most timesharing systems are set up to act like the host and therefore do not need special echo and line feed handling.

I INPUT TO BUFFER

The way OMNITERM does file transfers is through a buffer in memory that stores the data to be transfered. This buffer extends from the end of the area OMNITERM uses to the end of your memory as determined by the HIMEM pointer. This means that OMNITERM will not try to write over any high-memory drivers that you may be using, as long as they properly modify the HIMEM pointer value to point below themselves. The size and contents of this buffer is summarized on the bottom right line of the screen in command mode. It might show:

```
Buffer:  7194 of 27644 used
```

This means that the total size of the buffer is 27644 characters (or bytes) in length and that currently 7194 bytes are in use. When the I INPUT TO BUFFER function is ON, every character that is shown on the screen, that is, every byte transmitted or received in half duplex, or received in full, is placed into this buffer. When the INPUT function is OFF, the buffer is unchanged and the data in it simply sits there unaffected. Note that the exact amount of buffer space you will have depends on your operating system and version, and how much memory is in your computer.

As you press the I key to activate the INPUT command, different things happen depending on the current state of the function. If it is currently ON, it will simply turn OFF. But if it is OFF, when you press I, the buffer will first be

cleared and the display will go to:

```
Buffer:      0 of 13231 used
```

Only then will the INPUT function go on, so that all data will now go into a new, empty buffer. You must return to normal mode by pressing the <BREAK> key to actually receive data into the buffer. The command mode is used only to turn the buffer ON or OFF.

The remote computer can also open and close your buffer automatically by sending you control codes. By default receving a control-R will turn INPUT TO BUFFER to ON, and control-T will turn it OFF again. The remote computer can use this method to automatically open your buffer, send a file into it, then close it again.

If the buffer becomes full, the normal screen will show:

```
*** ERROR --> I/O BUFFER FULL
*** INPUT TO BUFFER STOPPED
```

If you now go to the command mode, you will see that the entire buffer is filled and that INPUT TO BUFFER is shown as "OFF". This means that the file you are trying to transfer is too big to do in one chunk. You must first split it into smaller pieces that will fit in your buffer and then send or receive them one at a time.

INPUT TO BUFFER is used mainly when receiving a file from a remote system. The default value for it is OFF. Once you have the desired data in the buffer, you can save it in a disk file with the S SAVE BUFFER TO DISK function, described later.

Note that you cannot INPUT and OUTPUT from your buffer at the same time. If you try to turn on INPUT with OUTPUT is already ON, you will not be able to. The same goes for turning OUTPUT on when INPUT is already ON.

O OUTPUT FROM BUFFER

To send a disk file to another machine, you must get the file into your buffer (with the F FILL BUFFER FROM DISK function, described later) and then use the OUTPUT FROM BUFFER function. This function sends the contents of your buffer out to the remote computer, just as if you were typing it from the keyboard. The computer on the far end of the line must then receive it and somehow store it. Not all computer systems can keep up with a continuous output of data. They will fall behind and lose some unless there is a way of either slowing the output or stopping it until the other computer is ready for more. OMNITERM has several different ways of doing this, which may be used individually, in combination, or not at all.

The most common way to slow down output from OMNITERM is called XON/XOFF handshaking. The remote system will send an XOFF code (control-S) when it wants OMNITERM to stop sending. When it is ready to resume, the remote system will send an XON (control Q). OMNITERM is set up to respond to these commands and if the particular system you are communicating with

sends them, you may not need any other ways of slowing OMNITERM down.

When you press O to turn the OUTPUT FROM BUFFER function ON, the bottom of the command mode screen will clear and you will see:

```
Enter your prompt string.  If none, just hit <ENTER>.
Use [ and ] to enclose control characters -->
```

OMNITERM has the ability to do prompted output. This means that OMNITERM can send one line of text from the buffer and then wait for a prompt string to be sent by the remote computer indicating that it is ready to go on. This is done often with IBM computers and bulletin board systems. Usually, but not always, the prompt will be a single character such as a "?" or a ">" symbol. When the remote system sends this prompt string, it means that it is ready to receive another line of text and OMNITERM then sends it.

You may enter a promt string of up to 16 characters. If you do not want to use prompted output, but instead want OMNITERM to transmit data continuously, just press <ENTER> and no prompt string will be used.

If you want a carriage return or other control character to be part of the prompt string, you must enclose the code of the character you want in "graphic quotes". Press the <DOWN-ARROW> key to open a graphic quote, type in the code of the character you want and then <DOWN-ARROW> again to close the quote. For example carriage return would be:

```
[13]  (in decimal)          [0DH]  (in hex)
```

Where the [and] chraracters are the graphic quotes generated with the <DOWN-ARROW> key.

Once you have entered the prompt (if any) OMNITERM shows:

```
Enter time before each line (0 to 255) ----->
```

This allows you to have OMNITERM wait a preset amount of time before sending each line. To get this time delay, enter a number where 0 (or <ENTER>) means no time at all and 255 is the longest possible. This can be used to make OMNITERM wait a certain amount of time after receiving the prompt, or simply to generate a pause between lines even where the prompt wait feature is not used. This is often used with systems like IBM mainframes that send the prompt a little bit before the are really ready to receive a line.

Finally, OMNITERM shows:

```
Enter time between characters (0 to 255) --->
```

This lets you to set a fixed time delay that OMNITERM will wait between sending successive characters. Again, 0 or <ENTER> means no delay and 255 is the longest possible. Many timesharing computer systems are set up for someone typing about the speed of a typist and not a computer, so OMNITERM allows you to simply slow things down this way. You may find that values below 50 or so for the delay will seem to make no difference at all at 300 baud. These smaller delay values will only be useful at higher baud rates.

This is the least desirable method for sending data to a remote computer, since it is so slow. You should only use this when nothing else will work.

One final way of slowing down OMNITERM's output is with the W WAIT FOR CTS function. It is not one of the options included under this function but is described later in this manual.

Once you have gone through all three questions, you will see that the status flag for OUTPUT FROM BUFFER is ON and when you press <BREAK> to go back to normal mode OMNITERM will begin sending from the buffer. When OMNITERM finishes sending the contents of its buffer, it will show on the normal mode screen:

*** OUTPUT COMPLETE

This is simply to let you know that OMNITERM is done sending the buffer. If you want to stop OMNITERM when it is sending data out, you must go into command mode and press O again. This will turn the OUTPUT function OFF. Your buffer is unaffected and you may resend the buffer at any time.

Command Mode Functions - Right Column

The OMNITERM command mode functions in the right hand column do not have status flags, because they do not control ongoing functions that can be ON, OFF, or have a value to work with continuously. Instead, they are one-time functions that do their job and then finish. You will activate them when you want them and then be done with them. Again from top to bottom, the right hand column functions are:

X SYSTEM COMMANDS

When you press the X key from command mode, the screen will clear and a new menu of system functions will appear. You have entered what is called system mode. To leave system mode, press the <BREAK> key and you will return to the command mode.

Some of these system commands have very drastic effects and so they are harder to get at, as you would not want to use them by accident. Others are simply commands that are used only rarely, and need not take up space on the regular command mode menu. The system commands work much like the command mode commands, in that they work when you press the key in front of the description, with no need to press <ENTER>. Also, some have status flags much as in the command mode. The system commands are:

Key	Command Name	Function
Q	QUIT OMNITERM & GO TO DOS	Leave OMNITERM
C	COLD OMNITERM RE-START	Reset all defaults
S	SAVE OMNITERM SETTINGS	Write all settings to disk
L	LOAD OMNITERM SETTINGS	Read settings file from disk
E	EXAMINE DISK I/O BUFFER	Look at text in disk buffer
I	RE-OPEN BUFFER FOR INPUT	Add received data to buffer
A	AUTO-SIGNON & AUTO-DIAL	Setup & use Auto-Signon function
F	FILE TRANSFER	Transfer files with XMODEM protocol
R	RTS OUTPUT LINE	Toggle RTS RS-232 output line
D	DTR OUTPUT LINE	Toggle DTR RS-232 output line
W	WAIT FOR CTS	Pause output when CTS is ON
M	MODEM	Change modem interface type

Q QUIT OMNITERM & GO TO DOS

This is your way to leave OMNITERM and return to "DOS READY" level. Simply press the Q key and you will be finished with OMNITERM.

C COLD OMNITERM RE-START

If you press the C key, you will do what is known as a cold start of OMNITERM, which is the same as if you had left OMNITERM and gone to "DOS READY" and then re-run OMNITERM from the start. All flags, settings, etc., are put back to their default values regardless of any changes you may have made. This is used when you have mistakenly changed things, or simply want to start over. Just press the C key and OMNITERM will re-start for you from scratch.

S SAVE OMNITERM SETTINGS

As you have already seen in the "Using OMNITERM" examples you can set up the program ONCE for your application and then SAVE those settings onto a disk file. Almost everything about OMNITERM that can be changed is saved on disk when you use the S system command. The appendix gives a full list of what is saved in the file, but it is simply everything you can change within OMNITERM, except such things as the error counts and the contents of the buffer. When you have configured OMNITERM the way you want it, go into system mode and press S. OMNITERM will ask:

```
Should the settings file do Auto-Start?
```

The A AUTO SIGNON & AUTO DIAL function, described later, lets you setup OMNITERM to automatically dial and signon to your computer system. If you answer YES to this question, the settings file will be saved with a special auto-start setting. Then when you later re-load the settings file from DOS by typing "OMNITERM (space) settings-name", OMNITERM will automatically dial the number and sign you on to the remote system. (You can stop it from autodialing by pressing <BREAK> while the file is being loaded.) If you answer NO to this question, OMNITERM will not dial your number when you load the settings file until you tell it to do so manually.

OMNITERM will then say:

```
Enter save filespec -->
```

You must type in the filename you want to save the settings under. It may be any standard TRSDOS filename, including extentions, passwords and drive numbers. When you are done, press the <ENTER> key to indicate the end of the filename.

If you specify no extention, OMNITERM will add on "/OMT" as the default extention. This can be useful, since when loading in settings, OMNITERM will also assume "/OMT" is the extention if you do not give one. If for any reason a disk error occurs while saving your settings (disk full, illegal filename, etc.) OMNITERM will print the DOS error message and wait for you to press any key. It will then clear the screen and you will remain in system mode to try again. After a successful save, you will be in command mode.

You can have many different setting files for OMNITERM, each for use with a different computer. You may, for instance, have one called CPM/OMT, for use with a CP/M computer, or one called PRIME/OMT, for use with a Prime minicomputer. There are two ways of loading in these settings once you have saved them. The first is the L LOAD OMNITERM SETTINGS command described below. The other is when you enter OMNITERM from "DOS READY" level. If you add the name of the file on the same line as OMNITERM, that file will automatically be loaded in by OMNITERM, so that you will never see the default OMNITERM. Instead, as soon as OMNITERM starts, it's configured with your settings. So from "DOS READY" you could say:

```
OMNITERM IBM
```

to use OMNITERM with the settings for an IBM computer stored in a file called IBM/OMT, or you could say:

```
OMNITERM APPLE
```

to use OMNITERM with the settings for an Apple computer stored in a file called APPLE/OMT.

L LOAD OMNITERM SETTINGS

Once you are already in OMNITERM, you may wish to load in a different set of settings from disk than the one you started with. Press L in system mode to do this. OMNITERM will show:

```
Enter load filespec -->
```

You must type the name of the setting file you wish to load. As before, it can be a standard TRSDOS file specification, with extention, password and drive number. If you omit the extention, OMNITERM will assume "/OMT" for it. When the settings are loaded, you will return to command mode. If for any reason a disk error occurs while loading the file (file not found, etc.), the error message from DOS will be printed and OMNITERM will wait for you to press any key before going on. After you press the key, the screen will clear and you will still be in system mode so you can try again.

When you use this function OMNITERM will NOT automatically dial and signon even if you said YES to the Auto-Start question. This happens only when you load the settings file from the "DOS READY" prompt.

Settings (/OMT) files made for one version of OMNITERM cannot be used with other versions of OMNITERM. OMNITERM checks that any settings file that you try to load was made with OMNITERM version 1.60, and not any other version. If you try to load a settings file that won't work properly OMNITERM will give you the message:

```
Improper settings file.  Default settings will be used.
Hit any key...
```

You may then press any key and OMNITERM will return to normal mode with full default settings in place, just as if you had used the C COLD OMNITERM RE-START special command.

E EXAMINE DISK I/O BUFFER

This lets you examine text you have received with OMNITERM before saving it to disk to make sure it is correct. Just press the E key from system mode, and the screen will clear. At the top of the display you will see the message:

```
Hit SPACE to see text, <BREAK> to quit, or <ENTER> to restart
```

This means about what it says. To examine the text, hold down the space bar. As long as you hold it, the text will print on the screen. When you let go, it will stop until you press it again. If at any time you want to start with the text in the buffer from the beginning again, press the <ENTER> key and you will start over. If you see a large horizontal graphic bar go across the screen, you have reached the end of the text in the buffer. The output will stop until you let go of the space bar and press it again. The text will then start over from the beginning. To go back to system mode, just press <BREAK> at any time.

I RE-OPEN BUFFER FOR INPUT

This allows you to re-open the input buffer without clearing out the current contents.

This would be used when some data has already been received and the buffer closed, but you later wish to put more data into the buffer before saving it to disk. The I INPUT TO BUFFER function cannot be used, since it clears the buffer contents each time. Instead, use I RE-OPEN BUFFER FOR INPUT. As you press I, you will return to command mode and you will note that I INPUT TO BUFFER is now ON. The buffer size and contents are unchanged.

A AUTO SIGNON & AUTO DIAL

When you press A for this command, you will get a new sub-menu that will show you the current auto-signon string and current stored phone number. The stored phone number (and all references to dialing the phone) refers only to the Microconnection and Lynx modems. To dial "smart" modems such as the Hayes and Radio Shack Modem II, you can send the dialing commands from normal mode or the auto-signon string. This will be described further when the auto-signon string and auto-start are discussed.

This sub-menu shows the current auto-signon string and stored phone number, and includes commands to change the auto-signon string (A for Auto-signon), change the stored phone number (P for Phone), dial the stored phone number (D for Dial), and toggle the phone connection on and off (T for Toggle). From this sub-menu you may enter any of the above commands, or press <BREAK> to return to the system command menu.

A Auto-Signon string change

OMNITERM has the ability to allow you to store a series of characters to be used as an auto signon string. This is your normal sign-on sequence that you use to log on to a computer. You can store it here once, save it in the settings file and every time you need it, press control-0 (zero, not the letter O) to send it. The auto signon string can also be sent out automatically on request from a remote computer system. By default, each time OMNITERM receives an ASCII 5, ENQ, it will send out the Auto signon string.

There is no default auto-signon, so to change it you must press the A key from this Auto-dial sub-menu. OMNITERM will show:

```
Enter your new Auto-Signon message below. Use [ and ] (down-
arrow key) to enclose pauses and control characters. For
example: [PP] for two pauses; [13] or [0DH] for carriage return.
```

You may now type in your auto-signon message. It may be up to 255 characters in length and you can include control characters such as the <ENTER> key and pauses if necessary, using the graphic quotes.

With the graphic quotes, you use the <DOWN-ARROW> key to generate an "opening graphic quote", then type the decimal or hex code for the control character, or one or more of the letter P for pauses (about 2 seconds for each P), and then the <DOWN-ARROW> key to make a "closing graphic quote". You can then either press <ENTER> to end the string, continue with more characters in the usual way, or enter another control character by using the graphic quotes again. Although we cannot print the graphic quotes on paper, we will use the left-bracket ([) character to show an "opening graphic quote" and the right-bracket (]) character to show the "closing graphic quote" in these examples. Remember, both graphic quotes are made with the <DOWN-ARROW> key.

For instance, the code for carriage return is 13 decimal, or 0D hex. To enter a carriage return for your prompt you would type <DOWN-ARROW>, 13 (or 0DH, with the H on the end for Hex), and then <DOWN-ARROW> again. This would look like:

```
[13]  (in decimal)        [0DH] (in hex)
```

A possible auto-signon string might look like this:

```
[13][P][13][P]DECWR[13][P]C 301 24[13][PPP]ID TCA000 PASSWORD[13]
```

Which would send two carriage returns (code 13 decimal) followed by pause, the

string "DECWR", a carriage return and pause, the string "C 301 24", a carriage return and three pauses, the string "ID TCA000 PASSWORD" and a carriage return. This might be a typical auto-signon string for the Source through TELENET.

Note that you cannot mix control characters and pauses in the same set of graphic quotes, nor can you have more than one control character in the same set of quotes. You can however have more than one pause in a set of graphic quotes.

When OMNITERM sends the auto-signon string, it will not display it on the screen. OMNITERM automatically stops displaying incoming data as it sends out the auto-signon string. This is so that bystanders will not see your password. This feature gives you double protection, since nearby people cannot see you type the keys of your password, and cannot see it on the screen either.

Some computer systems with which you may use OMNITERM will echo back the auto-signon string some time after OMNITERM has sent it out. This could cause only the last few characters of the auto-signon string to be shown on your screen. Do not be alarmed if this happens. OMNITERM has still sent out the entire string, and it should work normally.

The text of the auto-signon string message is put through the FROM Disk Buffer table (described later) before being sent out. This table was chosen because the auto-signon was not felt to be important enough to merit a table of its own, although it did need to go through some table. The FROM Disk Buffer table was chosen because the auto-signon string's function is closest to that of outputting text from the disk I/O buffer, and thus it should use the same table. This information could be important if you wish to send characters that cannot be put in the auto-signon string, since you could translate other characters to the desired ones, as described below.

The most common problem people have with the auto-signon string is that you have entered an auto-signon string you believe should work, and it does not. Usually what happens is the remote computer system accepts the first part of the auto-signon string, and refuses to accept the rest. This problem is caused by an auto-signon string that is either incorrect, or one with insufficient pauses. It should be emphasized that all the auto-signon function does is send out data in the exact same way that you do when you press keys. IF IT WORKS WHEN YOU PRESS KEYS, IT WILL WORK WITH THE AUTO-SIGNON STRING. This is necessarily true, since the only difference between keystrokes and the auto-signon function is timing, and you can control this with pauses.

Before making up an auto-signon string, you should therefore be very familiar with how you would sign-on the system manually. It is important not to forget about carriage returns, which must be specified as [13] in the auto-signon string. One good technique for assuring that no keystrokes are forgotten when making up the auto-signon string is to first make up a list of keys by signing on to the system by hand, and as each keystroke is made, writing it down on paper.

Once you have the correct sequence in the auto-signon string, most of the time the auto-signon string will work as-is. In most cases the need for pauses is minimal, and they can usually safely be left out altogether. If this is done, it is important to wait several seconds after sending the auto-signon string for a

response, because many remote computer systems, especially the large networks such as Tymnet and Telenet, take a certain amount of time to respond to each sequence. If you become impatient with this and type needless extra information, you can cause extra problems.

If you follow the above and, after double-checking the accuracy of your auto-signon string, you still cannot get the remote system to accept your commands, then is the time to experiment with adding pauses.

The importance of experimentation cannot be over-emphasized. The auto-signon CAN be made to work, but you must be willing to experiment until you find your own mistake or until you find the proper combination of pauses for the system you are working with.

The code that control-0 (zero) generates by default is an ASCII 5, or ENQ. This byte is often used by other computers systems as a "who-are-you", or "answerback", character to request the caller to identify himself. OMNITERM will respond to this by acting as if you had pressed control-0 (zero), automatically sending your auto-signon string. This should make it easier to sign on to systems supporting the ENQ code.

P Phone number change

This function is ONLY for use with the Microconnection or Lynx modems. For a "smart" type modem such as the Hayes, you must include the dialing command and phone number in the auto-signon string.

When entering the phone number, you may tell OMNITERM to pause while dialing the phone by using the graphic quote feature. The only valid characters that OMNITERM will use when dialing is the digits 0 through 9 and pauses. Any other characters (letters, punctuation) will be ignored by OMNITERM when dialing. This means you are free to add spaces and punctuation to the phone number, as well as a short note describing who the number calls. Since the stored phone number may be up to 32 characters in length there should be room for a complete international phone number with pauses, or a local phone number with a short description. For instance, the phone number for your local TELENET node might be:

```
TELENET  9 [P] 754-9451
```

Where "TELENET" indicates who you are calling, the 9 is the initial 9 to get out of a centrex telephone system, a pause to get an outside line, and then the complete phone number. For most direct dial systems pauses will not be needed.

D Dial stored phone number with Microconnection or Lynx modem

To dial the stored phone number, just choose D from the menu. The number will be dialed and you will be returned to OMNITERM's normal mode to communicate with the called system. When you dial a number, make sure the Lynx is set to "Talk" (if you have a Lynx) and make sure that the Microconnection is set to "Data" (again, if you have one). When you dial a number OMNITERM will automatically hang up the phone for a couple of seconds, then pick it up and wait a few more seconds for a dial tone, then dial the number.

T Toggle phone line with Microconnection or Lynx modem

There is also a command to toggle the phone line. Press T to change from on-hook to off-hook (pick up the phone), or from off-hook to on-hook (hang up the phone). This insures that you don't have to worry about setting up the phone before dialing, and is an easy way to hang up when you are done.

With these powerful auto-signon and auto-dial features you can define all the elements of a sequence that will automatically dial a phone number, wait for the far end to respond with a carrier, and sign on to the remote system. Remember, that with a smart modem like Hayes or Radio Shack Modem II the phone number is stored as part of the auto signon sequence and not under the stored phone number option. You tell OMNITERM to automatically dial and signon when you first enter OMNITERM by saving the settings file using "auto-start".

This auto-start feature allows you to make up a separate settings file for each system you want to call in this way, and thus actually use your file directory as a phone directory! You will find this a very powerful and useful feature. When you are making up auto-signon settings for use with auto-start be sure you put in some pauses after dialing the phone to allow the phone system to connect you, and the other end to answer before the auto-signon is sent.

F FILE TRANSFER

TO USE THE XMODEM PROTOCOL
SEE PAGE 9 OF THE ADDENDUM

Pressing F from the system command menu loads into OMNITERM a special file called OMNIXFER/CIM. Since this file loads into the disk I/O buffer area both input to buffer and output from buffer must be OFF before the file can be loaded. If you try to load this file while either input or output are ON OMNITERM will not load the file, but will give you an error message. Also be aware that any data in the disk I/O buffer will be destroyed by loading this file, so be sure to save any data in the buffer that you want to keep.

Once the file has been loaded, you will be in another program, OMNITERM's file transfer program. From here you can send or receive files of unlimited size using Chistensen protocol. This protocol is also known as MODEM, XMODEM, or CP/M protocol. The advantage of using this method of file transfer is that there are no limitations on file size, any errors in transmission are automatically detected, and the correct data re-transmitted, and files of any type, including binary and compressed files, can be transferred with no need to convert them to ASCII first. The main disadvantage of this transfer method is that it can only be used with computers that have a compatible program such as XMODEM, MODEM72, MODEM80, or others. Many microcompuers and bulletin boards support the Christensen file transfer protocol. Of course it can also be used with other microcomputers using OMNITERM.

When you enter the file transfer program you will see a short copyright notice and explanation of the program, and then be asked:

```
Do you wish to send or receive?
```

If you want to send a file, answer with S. Answer with R to receive a file. You will be asked the name of the file you wish to transfer. When you name the file, the transfer will begin. As you are doing this, the computer on the other end of the connection must also get set up to transfer the file. In addition, both systems must be using the exact same UART settings for baud rate, stop bits, and parity. Because of the way the program checks for errors, the number of data bits must be 8. Any other value will not work. If you are communicating with another microcomputer user, this is no problem as long as you have agreed who is sending, receiving, and the name of the file. If you are communicating with a bulletin board system or other host computer, you generally must send some commands to the other computer before you enter OMNITERM's file transfer program. You will have to get the information from the other computer on how to do this.

Normally the two programs, the sender and receiver, will need a few seconds to get "in sync" with each other. While this is happening the file transfer program will keep you updated on the status of things. The file transfer program breaks up the file into blocks of 128 characters, and then sends each block along with error-detection checksums. As the transfer proceeds the file transfer program will show the status of each block, and the upper right corner of the screen will show each byte as it is transferred. This flashing display informs you that the transfer is progressing normally. If there are problems with a block the program will automatically re-send it, with an appropriate message to the screen. If, after 10 tries, it cannot transfer a block it will give up and allow you to decide if you want to keep trying or quit.

Should you have trouble transferring files this way, it is probably because the other computer is not running a compatible program, or it is not using the same UART settings including 8 bits, or you have not told the other computer to send (or receive) the file correctly.

R RTS OUTPUT LINE

In addition to the RS-232 input lines described earlier, there are two RS-232 output lines set by the TRS-80 that can control RS-232 devices. This command lets you control the RTS, or Request-To-Send RS-232 output line. When in the system mode the current setting of the RTS line, ON or OFF, is shown next to the command. By default, this is OFF for RTS, which is the correct setting for most modems. To change it, just press the R key. You will see the change on the system command mode screen. Some RS-232 devices such as modems require the RTS line (or DTR line, described below) to be set a particular way in order to work or to perform a special function. This is why you might want to change the setting of the RTS line. The status of the RTS line, like all other settings, can be saved as part of a /OMT setting file.

D DTR OUTPUT LINE

This is exactly like the R RTS OUTPUT LINE just described, except that it controls the Data-Terminal-Ready, or DTR output line. By default it is ON, the correct setting for most modems. To change it, just press the D key, and the change will be shown on the screen.

M MODEM

Beside this system command will appear the current modem interface used by OMNITERM. This can be one of three values, RS232, meaning the standard Radio Shack RS-232 interface board (or other hardware which is software-compatible with the Radio Shack board), LYNX, for the Emtrol LYNX direct-connect modem, or MCONN, meaning the Port 208 version of the Microconnection. This value is set each time OMNITERM is started. If a Port 208 type Microconnection is connected to the TRS-80 and has been turned on prior to OMNITERM being started, this value will be set to MCONN, and the Port 208 Microconnection will be used. Otherwise, the regular RS-232 board is used.

To use the Lynx modem it must be set to use the same port (port 232 decimal) as the Radio Shack RS-232 board if you want it to work with OMNITERM.

Each time the M key is depressed in system command mode to execute the M MODEM command, the modem type (as shown next to the M MODEM command) will change. It will switch between RS232 and LYNX unless you have a ON Microconnection modem connected and turned ON, in which case OMNITERM will cycle through all three types. This is how you can change the modem type you wish to use without restarting OMNITERM. This would be useful if you have different types of hardware on your TRS-80 and wish to switch between them. More often, since most people only have one type of modem, you might want to use an OMNITERM setting file that was created for use with the other

type of interface. Since modem type is saved as part of an OMNITERM setting file, your modem type could be set for the wrong type of interface. You would then use the M MODEM command to change this to be correct for your hardware.

Please note that all the sample OMNITERM setting files supplied on the disk use the standard RS-232 interface, and that LYNX and Port 208 Microconnection users must use the M MODEM command before these files will work with their modem.

Other differences when using the Lynx and Port 208 Microconnection modem are that the baud rate is always fixed at 300 baud, and that only the DSR line, of the four lines shown on the command mode screen, is active. This is because only DSR is detected by the Lynx and Microconnection hardware. Otherwise operation is the same as with the regular Radio Shack RS-232 board.

W WAIT FOR CTS

When this function is turned ON (it is OFF by default) OMNITERM will not output anything from the disk I/O buffer unless the CTS (Clear-To-Send) input line on the RS-232 port is ON. Instead, OMNITERM will wait while the CTS line is OFF. Anytime the CTS line is OFF OMNITERM will pause, and when it goes ON again OMNITERM will continue. Some types of RS-232 hardware use this method to indicate when they want transmission to pause.

To leave system command mode, simply press the <BREAK> key and you will return to command mode. These are all the system commands and now we continue with the command mode right-hand column functions.

T CHANGE/EXAMINE TABLES

One of the most powerful features of OMNITERM is its ability to translate a byte to or from any device to any other byte. OMNITERM uses as its devices the display screen, the communications, or comm line, the disk file buffer and the printer. The T function in command mode allows you to examine and modify seven byte translation tables, one for each possible device and one for each direction, input FROM a device, or output TO a device. You can also examine and modify the Control Key table and the Special Command table.

Translation Tables

The purpose of the translation tables is to allow you to translate any byte to or from a particular device to any other byte. This can be used for code conversions, or it can be used for customizing OMNITERM to work with various

hardware.

Your TRS-80 uses what is known as the ASCII character set. This stands for American Standard Code for Information Interchange. Although this code is the most popular by far, it is not the only code in existence. IBM computers use a code called EBCDIC, Extended Binary Coded Decimal Interchange Code. If you are, for instance, connected to a printer built for an IBM computer, you must translate all the ASCII bytes into EBCDIC before sending them to the printer. This can be done by changing the TO PRINTER table so that each ASCII code is translated to its equivalant EBCDIC code. You will find tables of both codes in the Appendix.

Since the majority of computer systems use the ASCII code, you will not be doing many code conversions with OMNITERM's translation tables. More often they will be used for customizing OMNITERM to specific hardware and to your taste. For example, the normal way to stop something on the TRS-80 is to press the <BREAK> key. OMNITERM tries to support this, since its default use of the <BREAK> key is to send a true Break, done through the special command table, described below. But the TRS-80 internal routines generate a value of 1 when the <BREAK> key is depressed. You may find a computer system that does not like true Breaks and instead wants a control character to get its attention. This is often a control-C or a control-P. Of course, you could simply send the control character to stop it, but this can be clumsy if you are already used to using the <BREAK> key. The solution is to use the FROM KEYBOARD table and translate the value 1 to the control character. This way, whenever you press the <BREAK> key, the control character will be sent instead.

This is how all the translation tables work. Another possible use is translating or masking out unwanted control characters. If a computer system sends control codes that make your printer do unwanted things, such as underlining, wide printing, or beeping, you could change these characters to nulls (ASCII 0) in the TO Printer table. Or, if you wanted a received disk file to go into a word processor with the CR's converted to spaces, you would simply change the value for CR to the value for space in the TO Disk Buffer table.

Control Key Table

Control characters and other special characters not on the TRS-80 keyboard are generated through use of the OMNITERM control key, or down arrow key. When the control key is pressed with another key on the keyboard, the character is first sent through a special table, the Control Key table. This table, by default, translates both upper and lowercase characters to the proper control code. Also by default it translates the digits 0-9 to special characters not on the TRS-80 keyboard, such as square brackets, broken bar, etc. Appendix A lists all of these.

The character is then sent through the FROM Keyboard table, just as are all other characters typed on the keyboard without the control key held down.

If you want other special characters of your own to be sent with the control key, you can add new values to the Control Key table. For example, to send

an ASCII ENQ code by pressing control-? you would change position 63 in the Control Key table to 5, since the ASCII value of ? is 63, and the code for ENQ is 5. Other changes could be made in a similar manner.

Special Command Table

OMNITERM has many functions that are done only when a particular byte comes from a particular device. For example, when you press the <BREAK> key, the value of 1 (control-A) is generated. Rather than the 1 being sent, a true Break occurs. As another example, when a control-G character (the ASCII code for ring the BELL) is received, a picture of a large bell will flash on the screen and you will hear a "beep" sound come through the cassette port (if it is wired correctly).

Both these functions, and many others described later in this manual, are handled by the SPECIAL COMMAND TABLE. This table is a list of all possible bytes. Each time a byte comes in from any device, the table entry for that byte is checked. If the value there is a 0, nothing unusual happens. But, if any other value is found there, the special command with that number is executed. For example, the special command number for the BELL function is 2. The ASCII code for control-G is 7, so the seventh entry in the special command table is 2. Thus every time a control-G is received, special command 2 is executed. If instead a 1 were there, a true Break would be sent, since 1 is the special command number for the Break function. If a 0 were there, nothing would happen.

If that value of 2, the special command number for the Bell function, were instead (or also) placed in special command table entry number 65, every time a capital A was received the Bell function would occur, since 65 is the ASCII code for A.

You can place any value in any spot in the special command table, so you can have the same function occur on any of a number of different bytes, or you can replace one function with another, or delete a function altogether.

Note that since OMNITERM checks the Special Command table only when a byte first comes in from a device, it cannot execute different special command functions based on what byte a TO translation table translates it to. OMNITERM can only do a special command on a byte as it comes in from a device, having only passed through the FROM translation table.

This is the how all the special commands work. There is a list of thirty of them later in this manual and they comprise some of the more powerful features of OMNITERM. If you can learn to use them they can make OMNITERM much more useful to you and since like all tables, they are saved in the OMNITERM setting file, you can use them to customize OMNITERM for each application.

Examining and Modifying Tables

You can look at and change the contents of any table by using the T key from command mode and selecting the number of the table you want to examine or alter. The screen will clear and you will see a list of the tables, numbered from one to nine. All the tables work essentially the same way when you want to access them. You choose one of the tables by pressing the digit from 1 to 9 that corresponds to the number of the table you want to access. There is no need to press the <ENTER> key. The screen will clear and you will see a message telling you to "Hit <BREAK> to quit", the name of the table you are looking at, a table of 64 numbers in Hex and their entries also in Hex and a message:

```
Hit - for previous page, +/; for next, <ENTER> to alter value
```

What you are seeing is the normal table display. This is a table of 64 of the total 256 possible bytes, all at once on the screen. There are four pages to each table, one for the bytes 0 to 63, one for 64 to 127, one for 128 to 191 and one for 192 to 255. When you first enter a table, you see the first page, for bytes 0 to 63 (00 to 3F in Hex). As the summary line describes, you may see the previous page by pressing the - key and the next higher one with the + key, which also has semicolon on it, but the <SHIFT> should not be pressed.

The way the table is set up, bytes enter on the left side of the = and the value for that entry is on the right side. A good table to look at for example is the "Control Key" table, which is table number 9.

Here you will see the special characters defined by default on the bottom two rows of data on the screen. The 1 key, which has a decimal value of 49 and a Hex value of '31', is translated to a Hex '7C', which is 124 decimal, the ASCII code for broken bar. Thus when you press control-1, the byte of 49 will be generated by the TRS-80, which is then translated to 124, broken bar. If you wanted the control-1 to send a different value, you could change the byte of '7C' Hex to whatever else you wanted. Most bytes in translation tables are not translated by default, so you will see many bytes with the same Hex value on both sides of the equal sign. To make bytes that ARE altered stand out, you will see small graphic bars on each side of a translation table byte that IS altered. Since most bytes in the special command table are zero, you will see bars there around bytes that are not zero.

If you want to change a table entry, press the <ENTER> key. You will see:

```
Enter byte to alter (add H for Hex)   -->
```

Type in the byte on the left of the equal sign you wish to alter. If you type it in Hex, place an H on the end of the number before pressing <ENTER>. If you type it in decimal, simply press <ENTER> when you are done. OMNITERM will respond:

```
That byte is currently (in decimal)   --> ###
Enter your new value (add H for hex)  -->
```

The ### will be the old decimal value of the byte you have selected. Since you may have entered your selection in decimal and now know its old value, you need not use Hex at all. If you just wanted the current decimal value, you can now press <ENTER> and nothing will change. To enter the new value, just type it in Hex or decimal and press <ENTER>. If you typed it in Hex, be sure to place the H after it so OMNITERM will know it is Hex.

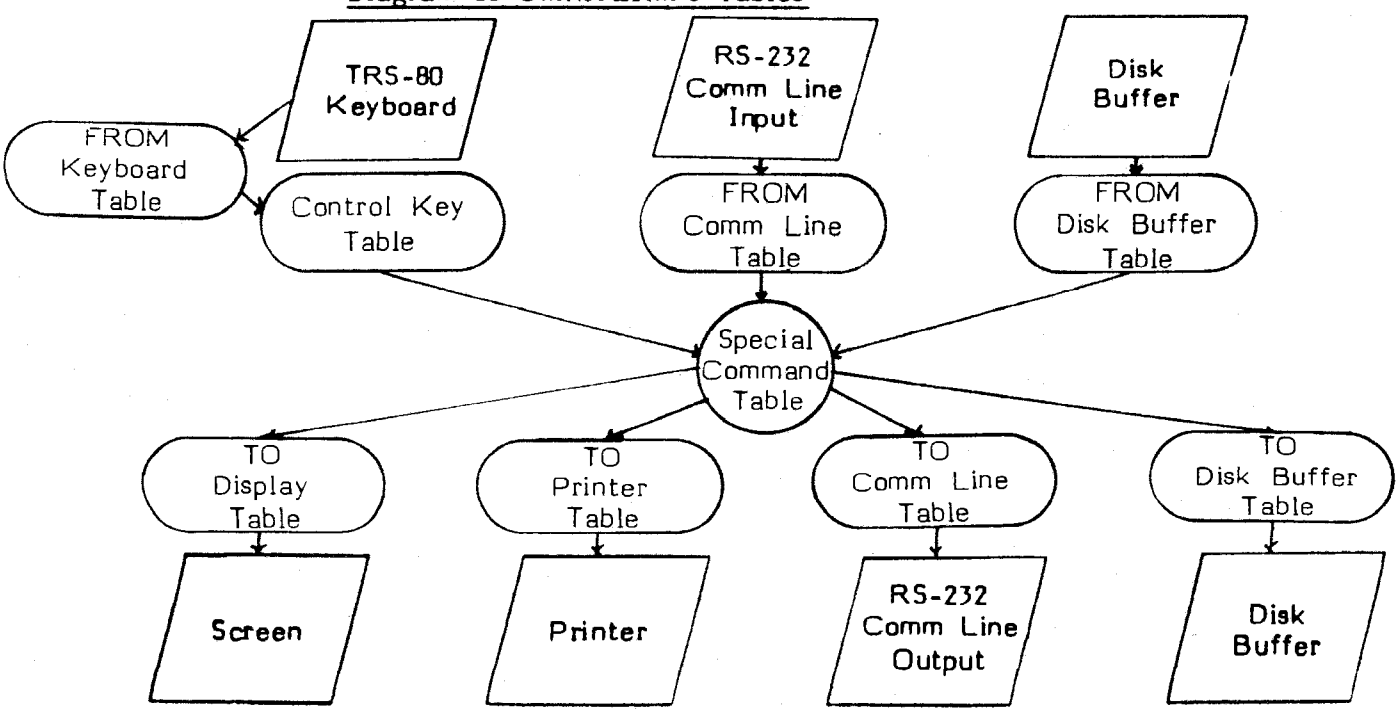
After you have changed a byte, you will see the new value in Hex in the table if you are on the correct page. You may now change more bytes, or simply press <BREAK> to return to command mode.

Examples of the use of OMNITERM's Tables

Since OMNITERM's special keyboard driver is only active when in OMNITERM's normal mode, you cannot use it to embed control characters, etc. into the auto-signon string (or for that matter into any string defined outside of OMNITERM's normal mode). Also, the semicolon character itself may not be placed into the auto-signon string, since it is used to symbolize the carriage return. The solution here is an example of the creative use of OMNITERM's tables. Since the auto-signon string is put through the FROM Disk Buffer table before being sent out, you could put some printable characters such as *, ?, >, digits, etc. that are not used in the rest of the auto-signon string in place of the desired unprintable characters, and then translate these to the desired characters in the FROM Disk Buffer table.

This is similar to the method used to allow OMNITERM to wait for a prompt made up of control characters. Since the control characters cannot be typed into the prompt string, other characters are used, and the control characters are translated to the characters actually typed in the FROM Comm Line table. This is another good example of clever use of the OMNITERM tables to get around a seemingly unsolvable problem.

Diagram of OMNITERM's Tables



U CHANGE UART SETTINGS

The UART is the device in the interface that does most of the work involved with making a series of pulses from a data byte when transmitting data and making data bytes from series of received pulses. Your Radio Shack RS-232C interface manual describes this device in detail and tells the meaning of Baud Rate, Parity, number of Data Bits (same as Word length) and number of Stop bits. Note that most timesharing systems can be used with OMNITERM at 300 baud, 8 data bits, no parity and 1 stop bit. If you are having trouble communicating at 300 baud, this is one of the first things you should check. Almost always you should use these settings at 300 baud.

OMNITERM allows you to fully reprogram the UART from the keyboard. If you have a Model I OMNITERM will by default set your UART to whatever values you have set the sense switches on the board, as described in Radio Shack's manual. If you have a Model III OMNITERM will use 300 baud, 8 data bits, no parity and 1 stop bit as its default settings, since the Model III has no sense switches. To change these settings with OMNITERM, press the U key from command mode. The bottom right of the screen will clear, and across from the message showing your current baud rate, you will see:

```
Baud Rate:
```

If you simply press <ENTER>, the baud rate will not change and you will go on to the next item. Or, you can type in any baud rate that the UART will accept and the UART will use this when you finish with the U command. The possible baud rates are 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, or 19200. If you type a number that is not one of these, no change will be made. After you type the number, press <ENTER> to go on to the next item, Data Bits.

OMNITERM will then show just below the previous line, across from the Data Bits display of the current number of data bits per word:

```
Data Bits:
```

Here you can type in the new number of data bits, either 5, 6, 7, or 8. If you do not type in any number, or if you type any other number than these, no change will be made. Press <ENTER> to go on.

Now OMNITERM will show on the next line:

```
Stop Bits:
```

You can now type in the new number of stop bits to be sent after each word. This may be either 1 or 2. If you type nothing, or anything but 1 or 2, no change will again be made. Press <ENTER> to go on to Parity.

The parity bit is an extra bit which, when used, makes the total number of bits that are 1 in each data word either an even or an odd number, even for even parity, or odd for odd parity. The purpose of this is error checking, since if a bit changes due to an error, the count will be different by one and the parity will be wrong. You will notice the Parity Error count in the usual

command mode display. This counts the number of parity errors as they occur.

Now across from the current Parity status display, OMNITERM shows:

Parity (E,O,N):

Type in E, O, or N to change the parity the UART sends and expects to receive. E stands for Even, O for Odd and N for None. Press <ENTER> when you are done. The usual command mode display will return, with your new UART settings included.

A SEND CONTROL-A & QUIT

Since the TRS-80 interprets the <BREAK> key as a value of 1, the same as a control-A and this value of 1 is usually (as a default) used to generate a true Break, some easy way of sending the control-A is needed, i.e. some way to do it without generating a true Break. This is the function of the A command. Just press it and the control-A will be sent and you will then leave command mode and go back to normal mode.

There are other ways to do essentially the same thing, such as disabling the true Break function in the special command table, but this way is simpler.

@ SEND "AT" SYMBOL & QUIT

In normal mode, when you press the @ key, it is to go into command mode. To send an @ character out, you must use the @ key from command mode. This means you must press the @ key THREE times to send it once. As soon as you press it from command mode, it will be sent and you will return to normal mode.

B SCROLL BACK DISPLAY

The scroll back function is one of OMNITERM's most useful and unique functions. OMNITERM automatically stores in memory the last 2048 bytes it has sent to the screen. You may look back at it to see information that has scrolled off the top of the screen. This can be very useful, usually for checking back at directions or numbers that went by when you weren't looking or forgot to pay attention. Press the B key from command mode and from that point it works just like the E EXAMINE DISK I/O BUFFER system command. The screen will clear and you can then use the space bar to see the text, the <ENTER> key to re-start from the beginning of the text, and <BREAK> to leave.

Z ZERO REAL TIME CLOCK

If you wish to use the TRSDOS real time clock as a timer, so you know how long you were on a computer system where you pay by the minute or must pay for a long distance phone call, you need a way of setting the clock to zero when you start. Just press the Z key from command mode and the real time clock will be set to zero. Don't forget that you must always say "CLOCK (ON)" to DOS before you run OMNITERM so the clock will display on the screen.

F FILL BUFFER FROM DISK

In order to send files out from your TRS-80, you must first bring them into your disk I/O (Input/Output) buffer. This is usually done with the FILL BUFFER FROM DISK command. Press the F key from command mode and the bottom of the screen will be replaced with:

```
Enter load filespec -->
```

Type in the name of the file you wish to transfer. It should be a standard TRSDOS style filespec, with extentions, passwords and drive numbers if you want them. Unlike the LOAD OMNITERM SETTINGS command, OMNITERM will assume no extention at all if you do not supply one. When you are done, press <ENTER> and OMNITERM will try to load the file. If a disk error occurs, the DOS message will print out and OMNITERM will let you read it while waiting for you to press any key. When you press a key, the usual command mode screen will return and you can try again.

A successful load will clear out whatever was in the buffer before and replace it with the file. You are now ready to transmit the file using the OUTPUT FROM BUFFER command, discussed earlier.

S SAVE BUFFER TO DISK

Once you have received some text and placed it into your buffer, you will want to save it in a disk file. This is done with the SAVE BUFFER TO DISK command. Press the S key from command mode and the bottom of the screen will show:

```
Enter save filespec -->
```

You now type in the name you want to save the text under, in standard TRSDOS form. OMNITERM will not supply any default extention. If a disk error occurs while saving the file, such as a full disk or other problem, OMNITERM will try to close the file, so that any text that WAS written will at least be retrievable. You will probably not get all of it after an error, but you may at least get some. OMNITERM will then print the DOS error message and wait for you to press a key before returning to command mode. In no case will the S function alter the contents of the disk I/O buffer, so if you

have a problem saving it to disk, you can try again.

Lowercase Keyboard

The TRS-80 by default generates uppercase letters unless you press the <SHIFT> key, the reverse of a normal typewriter-style keyboard. This is fine for most applications, but occasionally you may want to get lowercase by default and press <SHIFT> for uppercase, as on a typewriter.

The Model III and some Model I operating systems allow you to do this by pressing the <SHIFT> and 0 keys together and thereby toggle back and forth between the two modes. This is the simplest way to solve the problem.

If your DOS or computer does not do this, you can do it by altering your FROM Keyboard translation table so that the Uppercase letters A-Z are changed to lowercase and the lowercase letters a-z are changed to Upper. Thus, when you want the special keyboard, just use OMNITERM with this translation table. A sample of such a table is included on your OMNITERM disk, as the file LOWER/OMT.

Special Command Functions

OMNITERM has a library of built-in special command functions that are used throughout the Special Command table, as described earlier. Many of these are assigned default usages and you may not need to ever change them. Others can only be used if you put them into the table where you want them.

Many of these functions work only on bytes that are from a particular source. This is possible since OMNITERM keeps separate track of the last character received from anywhere, LSTCHR, the last local character (disk or keyboard) LSTLOC, last remote character (comm line) LSTREM and last keyboard character LSTKEY.

You will notice that commands 6 through 16 only work if they are received over the comm line and if they are preceded by an ESCAPE. These are known as Escape sequences and are functions that are only valid after an ESCAPE has been sent. They are used for special screen control functions and are the same commands accepted by Digital VT52 terminals. Thus if you connect to a computer that expects a VT52 terminal (or can be made to do so) OMNITERM will accept these commands. However be aware that OMNITERM is not a true VT52 emulator because of the small (64 by 16) screen size compared to the VT52 (80 by 24).

Special Command Function Summary Table

Cmd #	Command Name	Decimal Default	Function
1	Break	1	Sends true Break to comm line
2	Bell	7	Flashes graphic bell, "beeps"
3	LF Suppression	10	If ON, kills LF if LSTCHR=CR
4	CR/LF Grouping	13	Sends CR/LF pairs, echos LFs
5	Enter Command Mode	64,96	Hit twice, goto Command Mode
6	Wide Character Mode	109	32 Char/Line if LSTREM=ESCAPE
7	Narrow Character Mode	108	64 Char/Line if LSTREM=ESCAPE
8	Cursor Up	65	Cursor up 1 line if LSTREM=ESCAPE
9	Cursor Down	66	Cursor down 1 line if LSTREM=ESCAPE
10	Cursor Left	68	Left 1 space if LSTREM=ESCAPE
11	Cursor Right	67	Right 1 space if LSTREM=ESCAPE
12	Home Cursor	72	Cursor to top left if LSTREM=ESCAPE
13	Clear to EOL	75	Clr to end of line if LSTREM=ESCAPE
14	Clear to EOP	74	Clr to end of page if LSTREM=ESCAPE
15	Clear Screen	106	Clear screen if LSTREM=ESCAPE
16	X/Y Cursor Position	89	Direct cursor move if LSTREM=ESCAPE
17	XOFF (Transmit OFF)	19	Pause buffer output
18	XON (Transmit ON)	17	Resume buffer output
19	Init/Open Buffer	18	Start input to buffer
20	Close Input Buffer	20	Stop input to buffer
21	Ignore but Store	27	Ignore, but use as Last Char
22	Send Auto Signon	5	Start sending auto signon msg
23	Ignore without Store	0	Ignore, do not save
24	Beep without Flash	NONE	Send tone to cassette only
25	Wide Char w/o ESCAPE	NONE	Same as #6, w/o ESCAPE
26	Narrow Char w/o ESCAPE	NONE	Same as #7, w/o ESCAPE
27	Home Cursor w/o ESCAPE	NONE	Same as #12, w/o ESCAPE
28	Clear Screen w/o ESCAPE	12	Clear Screen on Form Feed
29	X/Y Cur Posn w/o ESC	NONE	Same as #16 w/o ESCAPE
30	Horizontal Tab	9	Move to next tab stop
31	CLS From Keyboard	250	Clear Screen on key

Special Command Function Descriptions

1 - Break

The Break function sends what is known as a true Break over the communications line. This is not an ASCII character and has no numeric value to it, but is simply an extended zero signal sent over the interface. Many computer systems use this to interrupt them or get their attention. This function will only work when it is called from a Keyboard key. By default this is assigned to your <BREAK> key.

2 - Bell

When the Bell function is activated, a large graphic bell will briefly flash on the screen and at the same time a "beep" sound will be sent through the cassette port. This by default works when a control-G character is received. This is the ASCII code for BELL. Unless you are in half duplex mode, the character must be received over the comm line for this to work.

3 - LF Suppression

This function, by default set to Line Feed, ASCII 10, can be set ON or OFF from command mode. If OFF, it does nothing. If ON, it will suppress the character it is called from (usually Line Feed) if the previous character processed by OMNITERM (LSTCHR) was a carriage return. If the LSTCHR was anything else, this function will do nothing.

4 - CR/LF Grouping

This function, by default set to Carriage Return, ASCII 13, can be set ON or OFF from command mode. If OFF, it does nothing. If ON, every time OMNITERM gets a CR from any source, it will check where it is from. If it is local, OMNITERM will send a CR over the comm line as the first half of a CR/LF pair, then translate the original GR into a LF and process that normally. If the CR was remote, OMNITERM will send a LF over the comm line to move up the remote terminal's screen and then process the CR normally.

This is used when your TRS-80 is used as the host machine and you must service the remote device by echoing its CRs with LFs and by sending CR/LF pairs instead of just CRs.

5 - Enter Command Mode

This function, by default set to the @ key and the <SHIFT>-@ key (ASCII 64 is @ and <SHIFT>-@ is Accent Grave, ASCII 96) is used to enter command mode from normal mode. When you press this key twice with no other keys intervening, you will enter command mode. It is set to <SHIFT>-@ because some DOS <SHIFT>-0 UPPER/lower case switches also change @ to <SHIFT>-@. If you wanted to use a different key to enter command mode, you could reassign this function to the ASCII code for another key. This function only works when called from a keyboard byte.

6 - Wide Character Mode

When this function is invoked, it will check to see if the last character received was an ESCAPE. If it was not, nothing will happen. If it was, OMNITERM will place the TRS-80 into its 32 character per line wide character mode. This is used primarily when OMNITERM is acting like a VIDEOTEX machine. The default value for this is ASCII 109, the small letter m.

7 - Narrow Character Mode

This function is used to return to the normal TRS-80 64 character per line size. It will act only if the last character received over the comm line was an ESCAPE. The default value is ASCII 108, the small letter l.

8 - Cursor Up

This is another function that requires the ESCAPE code to have been received. It moves the cursor up one line on the screen. The default value is ASCII 65, the letter A.

9 - Cursor Down

Moves the cursor down one line on the screen. Default value is ASCII 66, letter B. Requires ESCAPE to be received first.

10 - Cursor Left

Moves the cursor left one space. Default value is ASCII 68, the letter D. Requires ESCAPE first.

11 - Cursor Right

Moves the cursor right one space. Default value is ASCII 67, the letter C. Requires ESCAPE first.

12 - Home Cursor

This also requires ESCAPE to have been received and its default value is ASCII 72, the letter H. When it is used, the cursor will go to the Home position on the screen - the upper left corner. The contents of the screen will not change.

13 - Clear to EOL

Requires ESCAPE first and the default is ASCII 75, letter K. It clears the screen from the position of the cursor to the end of that same line, thus it Clears to End Of Line.

14 - Clear to EOP

Requires ESCAPE first, the default is ASCII 74, letter J. It clears the screen from the cursor's current position to the end of the screen, thus it Clears to End Of Page.

15 - Clear Screen

Requires ESCAPE first, has a default value of ASCII 106, the small j. It clears the screen and puts the cursor in its Home position in the upper left corner of the screen.

16 - X/Y Cursor Position

This function requires ESCAPE before it and has a default value of 89, an ASCII Y. It lets the cursor go directly to any spot on the screen with one command. After this command is invoked, the next two characters received over the comm line specify first the Y, or vertical and then the X, or horizontal position on the screen that the cursor will move to. The ASCII value of these characters, minus 32, are the Y and X coordinates for the cursor's new position. If either coordinate indicates a location that is not on the screen, the last row or column on the screen in that direction will be used instead. Note that the count of X and Y positions starts from 0,0 which is the upper left corner of the screen.

17 - XOFF (Transmit OFF)

This is the method of pausing OMNITERM's output from the buffer that was briefly mentioned before. This command can only be invoked from the comm line. When it is received, OMNITERM will stop outputting from its buffer at once. When the XON function is invoked, it will continue from where it left off.

This is a very common method for controlling file output speed that you will find to be compatible with many computers. Its default value is 19, an ASCII DC3, Device Control 3. This is the standard control code for this function and you will probably never have to change it. Note that many computer systems accept a Control-S code to pause their transmission to you. This is because the Control-S code is an ASCII DC3 and the remote system is responding just as OMNITERM would with the same code.

18 - XON (Transmit ON)

This function is the companion to XOFF. After the XOFF command has been used, OMNITERM will wait for the XON command before resuming output. It will continue from where it left off when the XOFF command was received. This command is not meant to start output remotely, without any previous XOFF, but merely to resume output after the XOFF command. The default value for this function is 17, ASCII DC1, a Control-Q code. This is standard, just as XOFF is and will probably never need to be changed.

19 - Init/Open Buffer

This command has the same effect as when you enter command mode and press I, for INPUT TO BUFFER. It clears the buffer and everything received thereafter goes into it. If after receiving this command you go to command mode, you will see that INPUT TO BUFFER is ON. This is used so that the remote computer system can open your input buffer when it wants to send you a file. Just as with command mode, if you are already outputting data when this command is received, it will be ignored. When this command is received and acted upon, you will see on your normal mode screen:

*** INPUT BEGUN

This is to inform you that your buffer has been opened. If you want the file being transmitted, it is not a good idea to type anything until it is done, or you may put some stray characters into your buffer. The default value for this function is 18, ASCII DC2. This is a fairly standard use of this code and many systems support it.

20 - Close Input Buffer

This command is usually used as a companion to Init/Open Buffer (special command number 19), but it need not be. It simply closes the input buffer if it is already open, so that no new characters will go into it. You will see on your normal mode screen:

*** INPUT COMPLETED

This means the command has been received and done. This function is used by remote systems when they are finished sending a file. If they did not use it, you might get some stray data at the end of your file (such as the remote system's prompt for your next command) that you would have to edit out later. The default for this function is 20, an ASCII DC4 and is somewhat standard. The function, like all the special command functions assigned by default to Device Control codes, will work only on bytes received over the comm line.

21 - Ignore but Store

This function allows OMNITERM to ignore a character in that it will not send it to the screen, printer, buffer, etc, but will still store it as the most recently processed byte in LSTCHR, LSTREM, LSTLOC and LSTKEY, as appropriate. This is assigned by default to 27, the ASCII code for ESCAPE. This is used when a byte primarily indicates the use of the following byte, as with ESCAPE, which must precede some functions for them to work.

22 - Send Auto Signon

This function causes the Auto Signon message to be sent out. By default, this function occurs on the value for Control-0, ENQ, 5. This can be triggered from the keyboard with Control-0 in the normal way, or can be triggered by the remote system when it sends an ENQ, or Who-Are-You character.

23 - Ignore without Store

When this function is assigned to a byte, that byte is ignored completely by OMNITERM. Whenever it comes in from a device, OMNITERM forgets completely about it and looks for the next byte. It by default assigned to 0, Null, so that you can translate any byte you want to ignore to Null.

24 - Beep without Flash

This is for those who feel that the flashing picture of a bell is distracting, while the "beep" sound through the cassette port (special command number 2 - Bell) is still useful. These people may substitute this function for #2, as it only produces the "beep", without the flashing bell. This command is not assigned by default to any byte.

25 - Wide Character Mode without ESCAPE

This functions identically to special command number 6, Wide Character Mode, except that it does NOT require ESCAPE to be received before its use. It is not assigned by default to any byte.

26 - Narrow Character Mode without ESCAPE

Identical to special command number 7, Narrow Character Mode, except that it does not require ESCAPE. Not assigned a default.

27 - Home Cursor without ESCAPE

Identical to special command number 12, Home Cursor, except that it does not require ESCAPE. Not assigned a default.

28 - Clear Screen without ESCAPE

Like special command number 15, this clears the screen, but it does not require ESCAPE. By default, this is assigned to ASCII 12, Form Feed. Some computer systems expect your terminal to clear the screen upon receipt of a Form Feed character.

29 - X/Y Cursor Position without ESCAPE

Identical to special command number 16, X/Y Cursor Position, except that it does not require ESCAPE. Not assigned a default.

30 - Horizontal Tab

When this function is invoked, the cursor will move forward to the next tab position on the screen and stop there. Tab positions are every 8 characters on the screen. That is screen positions 0, 8, 16, 24, 32 and so on. By default this is assigned to ASCII 9, Horizontal Tab. This code is normally transmitted by the right-pointing arrow key. Unless you are in half duplex mode, the character must be received over the comm line for this to work.

31 - Clear Screen from Keyboard

This is for clearing the screen directly from the keyboard, unlike the other clear screen special commands that are meant to be used when the remote system calls them. It by default works when control-: is pressed, because this special command defaults to 250 decimal, and control-: by default generates the 250 in the control key table.

With the cursor control functions that act without the need for ESCAPE, all cursor commands can be assigned to a byte and work without ESCAPE. The cursor movement commands described in Radio Shack's manuals can be used as well. For example, there is ASCII 30 for Clear to End of Line, 31 for Clear to End of Page, etc.

OMNITERM Utilities

Included on your OMNITERM disk are five programs which may be of use to you when working with OMNITERM. They are:

1) BINHEX/CMD A program to convert binary files to ASCII Hex files, so that they may be transmitted in a standard format.

2) HEXBIN/CMD A program to re-convert the ASCII Hex data created by BINHEX/CMD back to binary.

3) BINERR/CMD Converts any file to a special error-detecting, bit-packed, checksum format. More efficient than BINHEX.

4) ERRBIN/CMD A program to re-convert files made by BINERR which will detect any errors.

5) TEXTED/BAS A line-oriented text editor that can be used to create text to be transmitted by OMNITERM, or to modify received data.

BINHEX/CMD

OMNITERM is primarily designed to transfer text files to remote computers and to receive text files. Often, you may want to send object code or other binary files (such as compressed BASIC programs) to other TRS-80 computers. This can be done by setting OMNITERM so that it will not change any characters during transmission or reception. However, this approach will only work if the other TRS-80 is also using OMNITERM and if the RS-232 interface is using 8 bits of data. Also, if you want to leave the file on a timesharing computer system such as MicroNet or the Source before it is received, these systems will not allow you to transfer binary files through them.

BINHEX/CMD allows you to convert the binary file to a text file of ASCII Hex characters so that you can send the file with much more ease. When you run BINHEX, it will ask you for the source and destination files you wish to use. The source file is the binary file you wish to convert and the destination file will contain the resulting ASCII Hex code.

BINHEX will create the destination file by converting each byte of source code into a two-byte ASCII Hex value and will place no spaces between Hex values in the destination file. At the end of each 60 bytes of destination file (made from 30 bytes of source) and at the end of the file, a carriage return character is inserted. Note that this is the same format used by Lance Micklus' ST-80 terminal program and that Hex files created by BINHEX will be identical with those created by ST-80.

HEXBIN/CMD

This program reconverts the Hex files created by BINHEX back into binary files. When HEXBIN is run, it will ask for source and destination files. The source file should be a Hex file made by BINHEX or another program using the same format and the destination file will be the binary file you wish to recreate.

BINERR/CMD & ERRBIN/CMD

Many times it will be desireable to transfer files with some way to detect if any errors are present in the received file. HEXBIN and BINHEX do not supply this feature since they simply translate each byte of the source file into its hex equivalant, with no provision for error checking. This format is also very wasteful of disk and memory space since a hex file must take up exactly twice as much room as the original file. For these reasons, the simple hex format is not a very satisfactory solution to the problem. BINHEX and HEXBIN are on the OMNITERM disk only for compatiblity with other programs which already use this format.

Much like BINHEX, BINERR translates any file into its own special format. BINERR uses an entirely different format however. It is an error-detecting bit-packed format with checksums at the end of each line and a sum of all checksums at the end of the file. BINERR/CMD is used to create such a file from any source file, either binary or text and ERRBIN/CMD will convert the error-format file back to the original version.

The format created by BINERR packs six bits into each error-formatted byte, as opposed to four for BINHEX. This means that BINERR can store 50% more data in the same space. When ERRBIN is run, it will detect any errors found in the file and print an appropriate message. If this happens, you should re-transmit the file until no errors are found.

The exact format that BINERR and ERRBIN use is described in the appendix.

BINERR and ERRBIN should be used whenever a large file must be transfered over noisy communications lines, or when a binary file is to be moved. You will find it faster and better than BINHEX and HEXBIN and you should use it whenever possible.

TEXTED/BAS

There are many different kinds of data that you may wish to transfer with OMNITERM. You may wish to send BASIC programs, data files, or you may want to pre-compose a message for an electronic mail or bulletin board system. With TEXTED, you can create and modify files before sending them and you can edit files that have been received to remove extra information, such as mistyped commands or system prompts.

TEXTED is a BASIC program and it stores the text in memory, so the size of a file is limited by the amount of memory in your computer. It is a line-oriented text editor, which means it handles the text one line at a time and that you must refer to text by line number. The program is mostly self-explanitory and you should have little problem with it. When you RUN it, you will see a list of the commands and what they do. Just press one of the letters corresponding to what you want to do and the program will either do it, or ask you for more information. The functions available are:

Key	Function
?	Show list of commands
S	Save file to disk
L	Load file from disk
R	Retype line(s)
D	Delete line(s)
I	Insert line(s)
T	Type line(s) out to screen
P	Print line(s) out to printer
Q	Quit program and go to DOS
E	Edit within a line

Note that you do not have to press <ENTER> after one of these keys and that the line numbers start with 0, not 1. The total number of lines in the file is always shown when TEXTED is ready to accept a command (one of the listed keys). Thus the lines in the file go from 0 to the number listed. If, for any command, you specify lines that do not exist, TEXTED will ignore those and use only those that are in the file.

? Listing commands

If you press the question mark, the list of commands will be shown on the screen. So, if you don't know what to do, just press ?.

S Save file to disk

When you are done editing the file and have it just the way you want it, press S and the file will be saved on disk. You will be prompted to enter the filename first.

L Load file from disk

To edit a file that already exists, such as one you have received with OMNITERM, press L to load in the file. You will be asked the file name. After you type it the file will load in.

R Retype line(s)

To replace one or more lines in succession, press R. You will be asked to enter the first and last line numbers you want to retype and then the line numbers of each will print on the screen. Retype each one until TEXTED asks for a new command.

D Delete line(s)

To remove one or more lines from the file, use D. You will have to enter the first and last line numbers that you want to delete. If you only want to remove one line, enter its number as both the first and last line. If you try to delete more than 6 lines, you will be asked for confirmation before it is done. This is to ensure you do not delete a large portion of the file by accident.

After the lines are removed, the line numbers of all those lines following will be moved up into the gap left by the deleted lines. As a result of this, you will never have gaps of line numbers between lines.

I Insert line(s)

To enter in new lines, press I. You will be asked which line number you want the new lines to come after. Enter this. As you type each new line, the lines below will be moved down to make room for the new lines. When you are done, enter the word "END" as your line.

T Type line(s) out to screen

To display lines on the screen, press T. You will be asked for the first and last lines you wish to view. Remember that lines start with 0.

P Print line(s) out to printer

This works the same as T, except it sends the text to the printer.

Q Quit program

After you are done editing and have saved the text to disk, press Q to quit TEXTED. You will be asked if you are sure. Say YES if you are and you will return to "DOS READY" level and the text will be gone from memory. Be careful not to press this by mistake! If you say anything other than YES, you will return to the menu.

E Edit within a line

If there is an error within a line, you can change it, without retyping the entire line, by pressing E. You will have to enter the line number you want to edit. You will be asked to enter the OLDSTRING, or old phrase as it was wrong. You will then enter the NEWSTRING, or what you want the old phrase to be replaced with. For instance, in the line:

```
I think commputers are great because they are fast.
```

To change "commputers" to "computers", the OLDSTRING would be "commputers", or "mm" and NEWSTRING would be "computers", or "m". If there is more than one occurence of OLDSTRING in the line, only the first one will be affected. After the first is changed, you can then change the next one and so on.

Files supplied on the OMNITERM disk

The files supplied on your OMNITERM disk are:

OMNITERM/CMD The OMNITERM program itself.

OMNIXFER/CIM The file transfer program called from OMNITERM. This program cannot be used by itself, but only from within OMNITERM. If you try to call up this program from OMNITERM but it is not available on the disk, OMNITERM will give you an error message.

OMNIXFER/CMD The file transfer program in a form that you can run directly from DOS. This program works exactly the same way that OMNIXFER/CIM does, except that it can be run from DOS without using OMNITERM, and exits back to DOS. Note that this program does not set the UART, so you must have used OMNITERM to set up the UART before running this program.

LOWER/OMT A settings file for use with the Model I that reverses the keyboard case so that lowercase is normally typed, and uppercase is used only when the <SHIFT> key is pressed.

DJNS/OMT The sample settings file for Dow Jones News Service for use with the Model I.

DJNS3/OMT The sample settings file for Dow Jones News Service for use with the Model III.

MNET/OMT The sample settings file for CompuServe system, for the Model I.

MNET3/OMT The sample settings file for CompuServe system, for the Model III.

SOURCE/OMT The sample settings file for the Source and the Model I.

SOURCE3/OMT The sample settings file for the Source and the Model III.

BINHEX/CMD The binary to hex conversion program.

HEXBIN/CMD The hex to binary conversion program.

BINERR/CMD The binary to error format conversion program.

ERRBIN/CMD The error format to binary conversion program.

TEXTED/BAS The BASIC text editor program.

Supplied Setting Files

There are seven /OMT OMNITERM setting file tables on the OMNITERM disk. These are for your own use and can be used as examples of possible OMNITERM setting tables. You will probably want to modify them for your own particular use.

LOWER/OMT

This is identical to the default OMNITERM settings except that upper and lowercase keyboard characters are swapped. Model III owners will not want to use this file since the same effect can be gotten with <SHIFT>-0 (zero).

SOURCE/OMT

This file is meant for using the TRS-80 Model I with the Source, a popular information-service type timesharing computer system. Three changes were made for the Source. First, the <BREAK> key now sends Control-P rather than a true break. This was done by changing 1 in the Special Command Table to 0 to turn off the true Break function and changing 1 in the FROM Keyboard table to 16, the decimal value of Control-P.

Second, the underscore character is translated to a semicolon. The UPI news service on the Source often sends underscores instead of semicolons. This was done by changing 95 decimal, the value of underscore, in the FROM Comm Line table, to 59 decimal, the value of semicolon.

Lastly, UPI news service also at times sends a 26 decimal which causes OMNITERM to home the cursor but not to clear the screen. This is changed to a null by changing 26 in the TO Display table to 00, for Null.

There is also a sample Auto-Signon message for use with Tymnet and the Source. You will want to change the sample password and user number to your own.

SOURCE3/OMT

This is identical to SOURCE/OMT, except that its changes are made from the TRS-80 Model III default table contents. Model III owners will want to use this instead of SOURCE/OMT.

MNET/OMT

This is for use with Compuserve Information Service, or MicroNet. (MNET for short.) Changes are to the R SCREEN REFORMATTING, which is turned OFF and to the <BREAK> key, which now sends a Control-C. Also included is a sample Auto-Signon message for use with MicroNet. You will want to change this to your own user number and password. This table is for use with TRS-80 Model I.

MNET3/OMT

This is the same as MNET/OMT, except that it is for use with the TRS-80 Model III, just as SOURCE3/OMT is.

DJNS/OMT and DJNS3/OMT

These are identical OMNTIERM setting files except that DJNS is for the Model I TRS-80 and DJNS3 for the Model III.

These setting files are meant for use with the Dow Jones News Retrieval Service. Three changes have been made from the standard OMNITERM defaults.

1) A sample auto-signon string for the service when used with Tymnet is included. You will want to change the password to your own.

2) In the FROM Comm Line table, the hex values 11H, 13H, 1CH, and 1EH have all been translated to 00.

   Each time Dow Jones gives a new quote, it sends a 1C hex character, which normally makes OMNITERM home the cursor. This has been changed to 00 so OMNITERM will ignore this and make the screen look neater and easier to read.

   Dow Jones sends a 1E hex each time it is done with a quote and is ready to accept a new stock. This change gets rid of this control character. With this change you may build a file of stocks using TEXTED, and then send them all to Dow Jones at once using OMNITERM's prompted output mode. Just specify a question mark followed by two spaces as the prompt, and each stock name will be sent when Dow Jones is ready for it. This can save you money since Dow Jones charges by the minute.

   Dow Jones sends 11 hex and 13 hex, XON and XOFF codes, that can cause problems when sending a list of stocks to get quotes.

3) Screen Reformatting has been turned OFF. With it turned ON, some column displays from Dow Jones looked messy.

Conclusion

OMNITERM is the software you can use to communicate with almost any equipment your hardware interface will connect to. By accounting for control character incompatibility with translation and special command tables, it allows the remote system to continue using its own standard software, with no customization required. The TRS-80 user can perform all the needed changes within OMNITERM.

You do not have to understand and use all the features in OMNITERM in order to get useful work done with it, but the more you can feel comfortable with, the more flexibility you will have in your communications. If you understand all of OMNITERM's abilities, almost any communications problem can be solved through clever use of OMNITERM's capabilities.

If you can think of some new features you would like incorporated into OMNITERM, or have suggestions for improvements to OMNITERM, please write to Lindbergh Systems. Also, if you note any errors, please contact us.

All registered OMNITERM owners will be notified of changes and improvments in OMNITERM. Our policy allows registered owners to upgrade their programs at cost, currently, $20.

NOTE: The first registered OMNITERM owner to report an error we don't know about will receive the next available upgrade of OMNITERM absolutely FREE. This is to encourage the reporting of errors to us. If you don't tell us about the bugs, we can't fix them. Even if you report an error we already know about, you will receive our undying gratitude (also free).

Finally, we would like to thank the many people who encouraged and aided the production of OMNITERM and this manual, both knowingly and unknowingly. In particular, we would like to thank Robert Alexander, for his help with this manual and Lance Micklus, for unknowingly inspiring us to better things. But if you find anything you don't like, blame us, not them.

Appendix A - Translation Table Default Settings

Unless otherwise noted, all bytes not mentioned are translated to themselves. Only those translated to something else are described here.

TO Display

Input Byte	Output Byte
Dec = 91 *** MODEL I ONLY ***	Dec = 60
Hex = 5B	Hex = 3C
ASCII = Open Square Bracket	ASCII = Less Than Symbol
Dec = 93 *** MODEL I ONLY ***	Dec = 62
Hex = 5D	Hex = 3E
ASCII = Close Square Bracket	ASCII = Greater Than Symbol

The TRS-80 Model I character generator cannot display the square bracket characters, so OMNITERM will display them as corner brackets made from greater and less signs.

TO Comm Line, TO Disk Buffer, TO Printer

No default translations at all.

FROM Keyboard

Input Byte	Output Byte
Dec = 27	Dec = 94
Hex = 1B	Hex = 5E
ASCII = ESCAPE	ASCII = Up-Carat
Keyboard = <SHIFT> <UP-ARROW>	

To send an ASCII Up-Carat, press <SHIFT> <UP-ARROW>. This will generate an ESCAPE, which will be translated to the code for a carat.

Input Byte	Output Byte
Dec = 91	Dec = 27
Hex = 5B	Hex = 1B
ASCII = Open Square Bracket	ASCII = ESCAPE
Keyboard = UP-ARROW	

To send an ESCAPE, press UP-ARROW. This will generate an open square bracket code (in BASIC printed as UP-ARROW and used for exponentiation), which is translated to an ESCAPE.

Control Key Table

Input Byte

Dec = 48
Hex = 30
Keyboard = Control-0

Dec = 49
Hex = 31
Keyboard = Control-1

Dec = 50
Hex = 32
Keyboard = Control-2

Dec = 51
Hex = 33
Keyboard = Control-3

Dec = 52
Hex = 34
Keyboard = Control-4

Dec = 53
Hex = 35
Keyboard = Control-5

Dec = 54
Hex = 36
Keyboard = Control-6

Dec = 55
Hex = 37
Keyboard = Control-7

Dec = 56
Hex = 38
Keyboard = Control-8

Dec = 57
Hex = 39
Keyboard = Control-9

Dec = 58
Hex = 3A
Keyboard = Control-:

Output Byte

Dec = 5
Hex = 5
ASCII = ENQ

Dec = 124
Hex = 7C
ASCII = | (Broken Bar)

Dec = 95
Hex = 5F
ASCII = _ (Underscore)

Dec = 123
Hex = 7B
ASCII = { (Open Brace)

Dec = 125
Hex = 7D
ASCII = } (Close Brace)

Dec = 92
Hex = 5C
ASCII = \ (Backslash)

Dec = 126
Hex = 7E
ASCII = ~ (Tilde)

Dec = 127
Hex = 7F
ASCII = DELETE

Dec = 91
Hex = 5B
ASCII = [(Open Bracket)

Dec = 93
Hex = 5D
ASCII =] (Close Bracket)

Dec = 250
Hex = FA
ASCII = Trigger Keyboard CLS

The digits 0-9 are for the special characters that are not on the standard TRS-80 keyboard. Note that Control-8/Control-9 which are translated to open and close square brackets, are on the open and close parentheses keys. Also

note that Control-0 becomes an ENQ to trigger the auto-signon function. And Control-semicolon (;) becomes 250 decimal to trigger the Clear Screen from Keyboard function.

In addition to the above, all uppercase and lowercase ASCII characters A-Z and a-z are translated to Control A-Z.

FROM Disk Buffer

No default translations at all.

FROM Comm Line

All characters with codes of 80 hex (128 decimal) to FF hex (255 decimal) are translated to codes of value 128 decimal less. That is code 80H becomes 00H, 81H becomes 01H, 145 decimal becomes 17 decimal, 255 decimal becomes 127 decimal.

The effect of this is to "strip off" the 8th data bit of each incoming byte. This allows OMNITERM to properly receive characters sent with 7 data bits and any parity bit, 7 data bits, no parity, and two stop bits, or 8 data bits and no parity all with the same 8 data bits, no parity, and 1 stop bit UART settings.

This means that the OMNITERM user can almost always use these UART settings and does not need to change them to the exact proper settings.

Appendix B - Special Command Table Default Settings

All bytes in the Special Command table not described here are set by default to zero and no function will occur when they are processed by OMNITERM.

Processed Byte	Cmd Number	Name/ASCII value
Dec = 0 Hex = 00	Dec = 23 Hex = 17	Ignore without Store ASCII = NULL
Dec = 1 Hex = 01	Dec = 1 Hex = 01	Break ASCII = None
Dec = 5 Hex = 05	Dec = 22 Hex = 16	Send Auto-Signon ASCII = ENQ
Dec = 7 Hex = 07	Dec = 2 Hex = 02	Bell ASCII = BELL
Dec = 9 Hex = 09	Dec = 30 Hex = 1E	Horizontal Tab ASCII = Horiz. Tab
Dec = 10 Hex = 0A	Dec = 3 Hex = 03	LF Suppression ASCII = Line Feed
Dec = 12 Hex = 0C	Dec = 28 Hex = 1C	CLS without ESCAPE ASCII = Form Feed
Dec = 13 Hex = 0D	Dec = 4 Hex = 04	CR/LF Grouping ASCII = Carriage Rtrn
Dec = 17 Hex = 11	Dec = 18 Hex = 12	XON (Transmit ON) ASCII = DC1
Dec = 18 Hex = 12	Dec = 19 Hex = 13	Init/Open Buffer ASCII = DC2
Dec = 19 Hex = 13	Dec = 17 Hex = 11	XOFF (Transmit OFF) ASCII = DC3
Dec = 20 Hex = 14	Dec = 20 Hex = 14	Close Input Buffer ASCII = DC4
Dec = 27 Hex = 1B	Dec = 21 Hex = 15	Ignore but Store ASCII = ESCAPE
Dec = 64 Hex = 40	Dec = 5 Hex = 05	Enter Command Mode ASCII = @
Dec = 65 Hex = 41	Dec = 8 Hex = 08	Cursor Up ASCII = A

Processed Byte	Cmd Number	Name/ASCII value
Dec = 66 Hex = 42	Dec = 9 Hex = 09	Cursor Down ASCII = B
Dec = 67 Hex = 43	Dec = 11 Hex = 0B	Cursor Right ASCII = C
Dec = 68 Hex = 44	Dec = 10 Hex = 0A	Cursor Left ASCII = D
Dec = 72 Hex = 48	Dec = 12 Hex = 0C	Home Cursor ASCII = H
Dec = 74 Hex = 4A	Dec = 14 Hex = 0E	Clear to EOP ASCII = J
Dec = 75 Hex = 4B	Dec = 13 Hex = 0D	Clear to EOL ASCII = K
Dec = 89 Hex = 59	Dec = 16 Hex = 10	X/Y Cursor Position ASCII = Y
Dec = 96 Hex = 60	Dec = 5 Hex = 05	Enter Command Mode ASCII = ` (Accent)
Dec = 106 Hex = 6A	Dec = 15 Hex = 0F	Clear Screen ASCII = j
Dec = 108 Hex = 6C	Dec = 7 Hex = 07	Narrow Char Mode ASCII = l
Dec = 109 Hex = 6D	Dec = 6 Hex = 06	Wide Character Mode ASCII = m
Dec = 250 Hex = FA	Dec = 31 Hex = 1F	CLS from Keyboard Keyboard = Control-:

Appendix C - The ASCII Character Code

This is the American Standard Code for Information Interchange. It is not the exact same code as used by the TRS-80, since most computer manufacturers, Tandy included, change a few of the less often used characters to their own codes. On the TRS-80, some of the special characters will show differently. You will have to experiment to find them out. Try running this program in TRS-80 Level II BASIC:

```
10 FOR A=0 TO 255 : PRINT A,CHR$(A) : NEXT A
```

This will print a list of the decimal values and their displayed characters on your TRS-80. For more information on the ASCII code, refer to Kilobaud Microcomputing magazine, November 1980, page 129, "All about ASCII".

Dec	Hex	ASCII	Dec	Hex	ASCII	Dec	Hex	ASCII	Dec	Hex	ASCII
0	00	NUL	32	20	SPACE	64	40	@	96	60	`
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	"	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL	39	27	'	71	47	G	103	67	g
8	08	BS	40	28	(	72	48	H	104	68	h
9	09	HT	41	29	)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
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

Appendix D - The EBCDIC Character Code

This code was defined by IBM. You will notice that many values have no character assigned to them. Also, beware that there are many gaps within the letters. You will need this information if you want to create a translation table to be used with the EBCDIC code. Since many of OMNITERM's default function values conflict with EBCDIC, you will have to move these functions to different values in the special command table if you want to work with EBCDIC. For more information on EBCDIC, refer to IBM System/360 Principles of Operation, Appendix F, page 150.3.

Dec	Hex	EBCDIC	Dec	Hex	EBCDIC	Dec	Hex	EBCDIC	Dec	Hex	EBCDIC
0	00	NUL	32	20	DS	64	40	SPACE	96	60	-
1	01	SOH	33	21	SOS	65	41		97	61	/
2	02	STX	34	22	FS	66	42		98	62	
3	03	ETX	35	23		67	43		99	63	
4	04	PF	36	24	BYP	68	44		100	64	
5	05	HT	37	25	LF	69	45		101	65	
6	06	LC	38	26	ETB	70	46		102	66	
7	07	DEL	39	27	ESC	71	47		103	67	
8	08		40	28		72	48		104	68	
9	09		41	29		73	49		105	69	
10	0A	SMM	42	2A	SM	74	4A	CENTS	106	6A	
11	0B	VT	43	2B	CU2	75	4B		107	6B	,
12	0C	FF	44	2C		76	4C	<	108	6C	%
13	0D	CR	45	2D	ENQ	77	4D	(	109	6D	_
14	0E	SO	46	2E	ACK	78	4E	+	110	6E	>
15	0F	SI	47	2F	BEL	79	4F	|	111	6F	?
16	10	DLE	48	30		80	50	&	112	70	
17	11	DC1	49	31		81	51		113	71	
18	12	DC2	50	32	SYN	82	52		114	72	
19	13	TM	51	33		83	53		115	73	
20	14	RES	52	34	PN	84	54		116	74	
21	15	NL	53	35	RS	85	55		117	75	
22	16	BS	54	36	UC	86	56		118	76	
23	17	IL	55	37	EOT	87	57		119	77	
24	18	CAN	56	38		88	58		120	78	
25	19	EM	57	39		89	59		121	79	
26	1A	CC	58	3A		90	5A	!	122	7A	:
27	1B	CU1	59	3B	CU3	91	5B	$	123	7B	#
28	1C	IFS	60	3C	DC4	92	5C	*	124	7C	@
29	1D	IGS	61	3D	NAK	93	5D	)	125	7D	'
30	1E	IRS	62	3E		94	5E	;	126	7E	=
31	1F	IUS	63	3F	SUB	95	5F	NOT	127	7F	"

Dec	Hex	EBCDIC	Dec	Hex	EBCDIC	Dec	Hex	EBCDIC	Dec	Hex	EBCDIC
128	80		160	A0		192	C0		224	E0	
129	81	a	161	A1		193	C1	A	225	E1	
130	82	b	162	A2	s	194	C2	B	226	E2	S
131	83	c	163	A3	t	195	C3	C	227	E3	T
132	84	d	164	A4	u	196	C4	D	228	E4	U
133	85	e	165	A5	v	197	C5	E	229	E5	V
134	86	f	166	A6	w	198	C6	F	230	E6	W
135	87	g	167	A7	x	199	C7	G	231	E7	X
136	88	h	168	A8	y	200	C8	H	232	E8	Y
137	89	i	169	A9	z	201	C9	I	233	E9	Z
138	8A		170	AA		202	CA		234	EA	
139	8B		171	AB		203	CB		235	EB	
140	8C		172	AC		204	CC		236	EC	
141	8D		173	AD		205	CD		237	ED	
142	8E		174	AE		206	CE		238	EE	
143	8F		175	AF		207	CF		239	EF	
144	90		176	B0		208	D0		240	F0	0
145	91	j	177	B1		209	D1	J	241	F1	1
146	92	k	178	B2		210	D2	K	242	F2	2
147	93	l	179	B3		211	D3	L	243	F3	3
148	94	m	180	B4		212	D4	M	244	F4	4
149	95	n	181	B5		213	D5	N	245	F5	5
150	96	o	182	B6		214	D6	O	246	F6	6
151	97	p	183	B7		215	D7	P	247	F7	7
152	98	q	184	B8		216	D8	Q	248	F8	8
153	99	r	185	B9		217	D9	R	249	F9	9
154	9A		186	BA		218	DA		250	FA	
155	9B		187	BB		219	DB		251	FB	
156	9C		188	BC		220	DC		252	FC	
157	9D		189	BD		221	DD		253	FD	
158	9E		190	BE		222	DE		254	FE	
159	9F		191	BF		223	DF		255	FF	

Appendix E - OMNITERM Message Meanings

These messages may show on your normal mode screen from time to time. This Appendix will explain what they mean.

*** ERROR --> I/O BUFFER FULL
*** INPUT TO BUFFER STOPPED

This will happen when you are either loading a disk file into your disk I/O buffer, or when you are using the INPUT TO BUFFER command mode function. The message means that the buffer is full and cannot hold any more data. You will notice that the command mode buffer indicator shows that all bytes in the buffer are used.

If you were loading a file, the file will be closed and your buffer will hold all the data that had been loaded into it, up to the time when the message was printed. You can now send it, or find a way to break up the file into smaller pieces and send them one at a time.

If you were using the INPUT TO BUFFER function, the function will be turned OFF and the information in the buffer will be preserved. You can save this information to disk and continue with more data.

*** ERROR --> PRINTER BUFFER FULL
*** PRINTER TURNED OFF

This message will show when the printer has fallen too far behind the comm line to keep up. OMNITERM can only buffer a total of 2048 bytes before its print buffer becomes full. If this happens, your printer may be turned off or unplugged and OMNITERM is trying to buffer data going to a shut off printer.

When this message shows, the PRINTER flag in command mode will show that the printer is OFF, meaning that no new data is going into the print buffer. Since the buffer still has 2048 unprinted bytes in it, your printer will continue to print these out until the buffer is empty. You can turn the printer back ON at any time, although if the buffer is still full or nearly full, you may soon get this message again.

*** OUTPUT COMPLETE

This is OMNITERM's normal message meaning that the OUTPUT FROM BUFFER function is completed and the entire buffer contents have been sent. After this message, you will see the flag for OUTPUT FROM BUFFER go to OFF.

*** INPUT BEGUN

This message will show when OMNITERM executes special command number 19, Init/Open Buffer. This command opens your disk I/O buffer for input and the message indicates to you that this process has started. If the buffer was

opened by mistake, you may want to go to command mode and turn INPUT TO BUFFER to OFF.

** INPUT COMPLETE

This means that your disk I/O buffer has been closed because OMNITERM has received special command number 20, Close Input Buffer. In command mode, you will see that INPUT TO BUFFER is now OFF.

Improper settings file. Default settings will be used.

Settings (/OMT) files made for one version of OMNITERM cannot be used with other versions of OMNITERM. If you try to load a settings file that won't work properly you will see this message. You may then press any key and OMNITERM will return to normal mode with full default settings in place, just as if you had used the C COLD OMNITERM RE-START special commmand.

Large Picture of a Bell

When a large picture of a bell flashes on your screen, this means that special command number 2, Bell has been done. Normally this means that (by default) a control-G, ASCII BELL character has been received. This code causes the bell to ring on Teletypes, so OMNITERM flashes a picture of a bell and goes BEEP through the cassette port.

Parity, Framing, & Overrun Errors

Parity errors occur when the sum of bits in a byte is not what it is expected to be. This can mean that an incorrect data byte has been sent, that the UART is set for the wrong parity, or that the parity bit itself is wrong.

Framing errors occur when the stop or start bits of a data byte seem to be in the wrong place. This can happen because the UART was interrupted, or because of a bad data byte.

Overrun errors occur when characters are missed because the CPU did not get the old data from the UART before new data came in. This usually is because the computer is trying to do too many things at once, or too fast. If you type on the keyboard at the exact same time data is being received, this will often cause an overrun error since the computer missed the incoming data while processing your key. At the higher baud rates this happens more frequently because there is less time between characters.

It's normal to get a few of each of these errors when using OMNITERM with any communications line that is subject to noise. Most errors do not indicate bad data, but only that the UART thought the data strange. If you get less than 20 errors in each hour, you probably needn't worry. If you get hundreds, your UART is probably set incorrectly. The main purpose of these counts is just to give you a general idea of the quality of your communications.

Appendix F - OMNITERM Speed Limits

OMNITERM can only handle communications up to a certain rate without falling behind and missing characters. Although OMNITERM allows you to configure your UART for any baud rate up to 19200, this does not mean that OMNITERM can take data transmitted continuously at that rate. This means only that the interface hardware can take in the individual bits in each byte at that rate. OMNITERM needs a certain amount of time at the end of each line to scroll up the screen, and a lesser amount of time between each character. Only at high baud rates does this become a problem.

Model III OMNITERM uses the Model III's interrupt hardware, which effectively allows OMNITERM to run at its maximum average speed, which is much higher than the worst-case peak speed, which is the limit for Model I OMNITERM. Model III OMNITERM therefore can run at any speed up to and including 2400 baud, without any need for nulls or other pauses. In certain circumstances it can go as fast as 4800 baud, but this is not recommended for normal use.

Model I OMNITERM can handle any speed up to 600 baud without any special needs. To go beyond this it needs what are known as Nulls at the end of each line. Nulls are time units of the same length as a character that are sent at the end of each line to allow the screen to scroll up. They are needed because Model I OMNITERM, unlike the Model III, cannot buffer incoming characters and average out the time needed for each one. Instead it must completely process each one before the next is received. Since OMNITERM must scroll the screen up at the end of each line (a very time-consuming process) it needs extra time to do this. The number of nulls needed after each line (for Model I OMNITERM only) at various baud rates is shown below.

Baud Rate	# of Nulls	Baud Rate	# of Nulls
50	0	1200	1
75	0	1800	2
110	0	2000	3
134	0	2400	4
300	0	3600	6
600	0	4800	8

Up to 3600 baud, Model I OMNITERM, if it gets at least the proper number of nulls, can do everything as described in this manual. At 4800 baud it only has enough time to do the basic functions of translating bytes and sending them to the correct devices. If you try to run the printer, or do file I/O, at a full 4800 baud with no pauses between characters, OMNITERM will not be able to keep up.

Beyond 4800 baud, neither Model I or Model III OMNITERM can keep up at all with full speed. Although you can configure your UART for 7200, 9600 and 19200 baud, you should not run OMNITERM at these rates unless there is some wait time between characters. For instance, when you type on the keyboard, there is a long wait between characters and OMNITERM could keep up with this, or even something a great deal faster, at up to 19200 baud. Just do not try to run at these speeds with no wait time between characters.

Appendix G - OMNITERM Setting File (/OMT) Contents

When you save OMNITERM's Settings with the S system command, the file that is created has the following contents:

Name	# of Bytes	Description/Use
VRNUMT	2	Current version number
FLTYPE	1	Type of file (1=Settings)
MCCTLW	1	Microconnection Control Word
INTTYP	1	Interface Type: 0=RS-232, 1=MICRO
CTSFLG	1	1=CTS Wait, 0=CTS Ignore
LFFLG	1	1=LF Ignore, 0=LF Honor
REFORM	1	Screen Reformat Value
UIMAGE	1	UART Status Image
PRFLG	1	1=Printer ON, 0=Printer OFF
ECHFLG	1	1=Echo ON, 0=Echo OFF
DUPFLG	1	1=Duplex HALF, 0=Duplex FULL
CRFLG	1	1=CR Suppress, 0=CR Honor
CRLFGF	1	1=CR/LF Grouping ON, 0=OFF
UBDSTO	1	Baud Rate Image
DKIFLG	1	1=Disk Input ON, 0=OFF
DKOFLG	1	1=Disk Output ON, 0=OFF
AUTOF	1	Auto-start flag 1=ON, 0=OFF
PNLEN	1	Phone number length
PNBUF	33	Phone number storage
SIGLEN	1	Signon message length
MSIGST	63	Auto Signon message
KITBL	256	FROM Keyboard translation table
DKTBL	256	FROM Disk translation table
CLTBL	256	FROM Comm Line translation table
DOTTBL	256	TO Display translation table
CLTTBL	256	TO Comm Line translation table
DKTTBL	256	TO Disk translation table
PRTTBL	256	TO Printer translation table
CMDTBL	256	Special Command table
CMDADR	80	Special Command Address table
CTLTBL	256	Control Key Table

Appendix H - Connecting an Amplifier for the Bell

To hear the BEEP sound caused by special command number 2, Bell, you must connect an audio amplifier to the cassette output port of your TRS-80. The easiest way to do this is to buy an audio amplifier, such as that sold by Radio Shack for about $13 and connect it to the audio output from your computer. Other methods include modifying your cassette recorder, or connecting a speaker to the EAR jack on the cassette recorder and putting the recorder into RECORD mode with no cassette by holding in the tape feeler inside the recorder.

In any case, you should hear a BEEP sound when a control-G is received. To test it, put OMNITERM into Half DUPLEX and press control-G on your keyboard (down arrow and G). This should both make the BEEP and cause the picture of the bell to flash on the screen.

If you do not connect the speaker or audio amp to the cassette port, OMNITERM will still flash the picture, but no BEEP will be heard.

Appendix I - BINERR/ERRBIN Format

When BINERR/CMD is run, the output format consists of lines of ASCII characters with values ranging from 32 decimal (an ASCII space) to 95 decimal (an ASCII underscore). This makes a total of 64 possible characters, or 6 binary digits worth. These are created from 6 bit data by adding an offset of 32 to the original data, so that the resulting ASCII characters will be numbers, symbols, and uppercase letters, all of which are printable. Since these characters are the most commonly used in the ASCII set, they are not normally translated or manipulated by timesharing computer systems, and can be safely stored on them. In the following discussion of the contents of these lines, we will assume that the offset of 32 decimal is not present (has not yet been added, or has been subtracted) on each of the bytes.

The first byte is the number of 8-bit data bytes encoded into the line. This can range from 0 to 63 and will be referred to as the NOB (Number of Bytes). If it is 0, this indicates the End Of File (EOF) and is treated differently.

After the NOB comes the data itself, encoded into records of 4 bytes each. Each set of four bytes (storing 6 bits of data each) holds 3 bytes of original 8 bit data. Only the last record in the file may hold less than three data bytes, in which case the remaining empty bits are 0, and the NOB byte is not a multiple of three. Each record is treated as a string of 24 bits, into which the 24 bits of three 8-bit original data bytes are simply packed in sequence. This continues for as many records as are needed to fill out the NOB. Normally this will be for 39 bytes, or 13 records. It can be other lengths as specified in the NOB, but 13 records makes for a total line length of 55 bytes which seems easy to work with.

After the records comes a checksum of all the record bytes in the line. This takes two bytes in the formatted line, and so is 12 bits (MSB,LSB). It is the 12 least-signifgant-bits of a 16 bit checksum of the actual bytes as placed in the data records, including the offset of 32 decimal. The higher 6 of the 12 are in the first byte and the lower 6 in the second.

At the end of the line is a carriage return, followed by as many more lines as are needed to finish the file. Normally the last line will have less than 13 records, and in 2/3 of the cases one or two bytes of the final record will be unused.

After the final line of data comes a line whose first byte is a space (meaning there are 0 data bytes in the line because the ASCII value of space, minus 32, is 0) followed by a two byte checksum in the same format as the checksums at the end of each line. This checksum however is a sum of all previous checksums only.

Appendix J - Trademark Acknowledgments

Apple is a trademark of Apple Computer, Inc.
Atari is a trademark of Atari Inc.
DBLDOS is a trademark of Percom Data Company, Inc.
DOSPLUS is a trademark of Micro-Systems Software, Inc.
Dow Jones News Service is a trademark of Dow Jones and Comapny, Inc.
CP/M is a trademark of Digital Research Inc.
IBM and System/360 are trademarks of International Business Machines
LDOS is a trademark of Lobo Drives International
Lynx is a trademark of Emtrol Systems, Inc.
Microconnection is a trademark of Microperipheral Corp.
MicroNet and CompuServe are trademarks of CompuServe Inc.
NEWDOS and NEWDOS-80 are trademarks of Apparat, Inc.
Prime is a trademark of Prime Computer, Inc.
SmartModem is a trademark of Hayes Microcomputer Products, Inc.
Source is a service mark of Source Telecomputing Corporation
ST-80 is a trademark of Lance Micklus, Inc.
Telenet is a trademark of General Telephone and Electric
Teletype is a trademark of Teletype Corporation
TRS-80, Radio Shack, VIDEOTEX, and TRSDOS are trademarks of Tandy Corporation
Tymnet is a trademark of Tymshare, Inc.
VTOS is a trademark of Virtual Technology

If we have left out anyone here, we apologize, as it is purely unintentional. Please let us know and we will correct it as soon as possible.

Appendix K - Special Printer Drivers

This section is meant only for those with a good background in machine language programming. If you do not have this background we suggest that you ignore this information, as users not wishing to use unusual printer drivers will not need it.

OMNITERM sends all text going to the printer through the *PR system device. This means that special drivers for serial printers, special print formatting, etc., will work easily with OMNITERM.

The requirement for use of a different *PR driver is that it must follow the convention of returning the printer status when a 0 is sent to the *PR device. Whenever a 0 is sent out, the driver should return a byte with bit 7 set high if the printer is busy or the driver will not return immediately if a byte is printed. OMNITERM will wait for bit 7 of this byte to go low before sending data to the printer. If the driver you wish to use does not do this, OMNITERM will either never send data to the printer (bit 7 always high) or will send data to the printer even when a pause will result (bit 7 low). This would result in OMNITERM missing incoming data over the communications line while the printer driver is waiting.

OMNITERM will use the standard *PR printer device ONLY if that device's address (as stored at 4026-4027 Hex) is in RAM space above 4000H. If the *PR device is pointing to the routine in ROM, then OMNITERM will use its own driver instead. This is done because the Model III ROM driver does NOT follow the convention of returning printer status when sent a 0.

Appendix L - How to add your own Special Commands

This section gets into the technical innards of OMNITERM. It is meant only for those with a reasonably strong background in machine language programming. No attempt will be made to define unfamiliar phrases or concepts. If you do not have this background already, it is probably best to ignore this information.

This information, while correct in concept for all past and all likely future versions of OMNITERM, gives addresses correct only for OMNITERM version 1.60. In any future version they WILL change, without doubt, so be sure to check any addendums you may have for your version of OMNITERM. Also, while there is room for additional special commands at this point, as new special commands are added in future versions there may well be less space available for user use. Therefore this information is given on a strict at-your-own-risk basis. Several OMNITERM users have indicated that they would like to add their own special commands for a specific application. This information will allow you to do that, but be aware that any routines written to work with OMNITERM version 1.60 will have to be re-assembled at the least to work with future versions of OMNITERM. If you choose to give your special command a number which is the same as a future standard OMNITERM special command, you may have to change this as well.

OMNITERM Load & Run-time addresses

The first thing that must be known about OMNITERM is that it loads at 6000H and then moves itself down to 5200H for execution. All addresses are based on the OMNITERM run-time base address of 5200H, and not 6000H. It is important to be sure that when you modify OMNITERM's Special Command table, you do it while OMNITERM resides at 6000H (so you can write it back to disk), but to modify the bytes E00H higher than those mentioned in this text, since the run-time addresses are mentioned, not the load addresses.

The usual procedure for adding code onto the end of a program is to load the program, then load the additional code into RAM, and then DUMP it off to disk. This will not work with OMNITERM because OMNITERM uses all the space directly above it for buffer storage. Instead, you must place your routines in high memory, and then lower HIGH$ (also called HIMEM, at 4049H in the Model I, 4411H in Model III) appropriately. OMNITERM will respect space above HIGH$.

How OMNITERM processes Special Commands

As described elsewhere, each time OMNITERM processes a byte it runs it through the appropriate FROM table and then checks it against the Special Command table. The Special Command table, CMDTBL, is part of the OMNITERM setting file whose contents are described in Appendix G. The entire setting file is loaded sequentially by OMNITERM at 84A5H. This makes CMDTBL start at 8CDAH. Bytes are checked through CMDTBL by adding the value of the byte as an offset to CMDTBL. Thus the Special Command number

of any special command assigned to capital A, 41H ASCII, would be at 8CDAH plus 41H, or 8D1BH. CMDTBL can be modified by the OMNITERM user from within OMNITERM. If the contents of this offset byte is zero, then no special command is executed and OMNITERM continues normally.

If the contents are non-zero, then this value, the Special Command number to be executed, is doubled (to make room for a two-byte address in CMDADR), and then added to the address of CMDADR, the Special Command Address Table, located at 8DDAH. The resulting sum is the address of a two-byte pointer to the routine that should be executed to do the Special Command. This routine is NOT a subroutine, and is jumped to, not called.

To summarize, the value of the processed byte is added to CMDTBL at 8CDAH, and the byte at that address is fetched. If zero, nothing happens. If non-zero, the byte is a Special Command Number and is doubled and added to CMDADR at 8DDAH. The resulting sum is the address of a two-byte pointer stored in standard Z-80 format, which is the address of the routine that executes the Special Command. This address is then jumped to.

User Special Commands

At this point the user's routine would take over, do its thing, and then return to OMNITERM with a JP, not a RET. The user will have to pick a Special Command number for his routine. It should be greater than 31, since 31 is the highest Special Command number used by OMNITERM 1.60. However, it should not be greater than 39, since only 80 bytes are allocated to CMDADR, and any Special Command number greater than 39 would index to an area outside of CMDADR. So the only change the user would have to make to OMNITERM itself would be to place an entry in CMDADR pointing to the address of his routine.

Re-entry to OMNITERM

After execution of the user's routine, it should return to OMNITERM. There are only three recommended re-entry points to OMNITERM. They are MINPUT at 56CDH, MBYTE at 56F1H, and MBACK at 56F6H.

MINPUT

At MINPUT the contents of all registers is irrelevant. OMNITERM will check all input devices for characters, and continue from there. This re-entry point is used when the processed original byte (that caused entry to the user's Special Command) is to be ignored and not sent to any device by OMNITERM. Of course, the user's routine may well have done something with this byte, but that is not OMNITERM's concern. This re-entry point is used, for instance, by OMNITERM Special Command number 1, Break. This re-entry point is used because the (by default) control-A that caused the True Break to occur is not to be sent out as a control-A, but is simply used to trigger the function.

MBYTE

At MBYTE, OMNITERM will assume that a byte has been received and processed through the appropriate FROM table, but has not yet been checked against the Special Command table. It will proceed to do this, and go on from there. Register A should have the byte that OMNITERM has supposedly just received from the FROM table. Register IX should contain a 0 if OMNITERM is supposed to think the byte came from the comm line, a 1 if the byte supposedly came from the keyboard, and a 2 if it supposedly came from the disk buffer. All other registers are ignored. This re-entry point is not used by any current OMNITERM Special Command, but it is available for your use.

MBACK

At MBACK OMNITERM will assume that the byte has already been processed through the Special Command table, and any Special Command function to be executed has already been done. This is the most commonly used re-entry point. Register B (not A) should contain the data byte, and register IX should be as above. All other registers are ignored.

OMNITERM Memory Map

Here is a list of the major storage areas of OMNITERM. The addresses are for version 1.60 only. You will probably want to use some of these in your own routines.

Address	Name	Function
56CD	MINPUT	RE-ENTRY POINT
56F6	MBACK	RE-ENTRY POINT
56F1	MBYTE	RE-ENTRY POINT
8451	PRBUFF	NEXT BYTE TO PRINT
8453	PRNEXT	NEXT BYTE TO BUFFER
8455	PRBOTT	BOTTOM OF PRINT BUFFER
8457	DSPBOT	BOTTOM OF DSPBUF
8459	DSPNXT	NEXT BYTE TO BUFFER
845B	IOTOP	NXTBYT HERE (= EMPTY)
845D	PROMPT	PROMPT STORAGE
846D	LAST16	LAST 16 CHARS REC'D
847D	PRMLEN	LENGHT OF PROMPT
847E	PROMW	WAIT FOR PROMPT ON/OFF
847F	OUTIME	MAX TIMER VALUE
8480	OUTWAT	CURRENT TIMER VALUE
8488	IONEXT	NEXT BYTE TO SEND
848A	LSTCHR	LAST CHARACTER SEEN
848B	LSTREM	LAST REMOTE CHARACTER
848C	LSTLOC	LAST LOCAL CHARACTER
848D	LSTKEY	LAST KEYBOARD CHARACTER
848E	CHRMDE	MODE 0=64 CPL, WIDCHR=32
848F	LBTEST	LAST BREAK KEY TEST
8490	RPTCTR	REPEAT COUNTER
8491	DOSTO	OLD *DO DCB STORAGE
8493	PRSTO	OLD *PR DCB STORAGE
8495	SIGLFT	# OF AUTO-SIG BYTES UNSENT
8496	UPARTY	PARITY ERROR COUNT
8498	UOVERN	OVERRUN ERROR COUNT
849A	UFRMNG	FRAMING ERROR COUNT
849C	SCNFLG	SCAN FLAG FOR KISCAN
849D	CURBUF	CURSOR ADDR STORAGE
849F	TOPRAM	HIGH$ STORE (I OR III)
84A5	/OMT File starts with MCCTLW as described	in Appendix G
8F2A	/OMT File ends with CTLTBL as described	in Appendix G
8F4B	CMDBUF	TEXT BUFFER
8F8A	CBUF	STORE SCREEN HERE IN C-MODE
938A	DSKBUF	DISK FILE BUFFER
948A	DSKDCB	DISK DCB
94BC	IOBOTT	LOWEST USABLE I/O BUFF BYTE

From 94BCH onward, RAM is dedicated to I/O and other buffers, up through TOPRAM.

Appendix M - Troubleshooting Guide

Before you call for help:

HAVING TROUBLE DIALING YOUR RADIO SHACK MODEM II??

The Radio Shack Modem II will accept commands only when your UART is configured for 8 data bits, 1 stop bit and no parity. This is the default UART setting for OMNITERM but be sure that you have not changed it if you are having trouble with the Modem II.

HAVING TROUBLE SENDING SCRIPSIT OR BASIC FILES??

Be sure you save the file from BASIC or SCRIPSIT with the ",A" option so it will be in standard ASCII. Otherwise these programs will use a special compressed format that other computers will not understand and that will look like garbage on your screen. See your BASIC or SCRIPSIT manual for details.

TROUBLE LOADING RECEIVED BASIC PROGRAMS??

Getting "DIRECT STATEMENT IN FILE" errors when trying to load a BASIC program? Check for garbage in the file with the TEXTED/BAS program. Often there are stray messages at the beginning or end of a program that you mistakenly saved with the program. Other times long lines in BASIC are broken up into two smaller lines before being sent. You must repair them with the TEXTED/BAS program before you can use the program. Don't forget that files in TEXTED/BAS start with 0, not 1, so check that first line too!

IS YOUR CURSOR FLASHING PAST AT THE BOTTOM OF THE SCREEN??

If your cursor is flashing quickly across the bottom of your screen, and you are unable to do anything else with OMNITERM, this is the classic symptom of RS-232 board hardware problems. Your RS-232 board is almost certainly either missing, broken, or loose. (If you have a Lynx modem addressed to the wrong location and no RS-232 board you will also have this problem, since as far as OMNITERM is concenred your RS-232 board is missing.) This can happen on any model TRS-80. On the Model I the connector between the expansion interface and the RS-232 board is often the problem, and you might be able to fix it by gently moving the board and/or connector. If this does not work the only solution is to take your computer back to Radio Shack for repair. Note that sometimes you can have this problem even when the RS-232 board works correctly with other software. This is usually because some software does not make use of all the functions of the RS-232 board.

CAN'T GET YOUR MODEM TO DIAL THE PHONE??

Your modem won't dial the phone when you use the P DIAL TELEPHONE system command? Remember this is designed to work only with the Microconnection modem with the optional Auto-dial feature. Do not expect it to work with other brands of modem or with a Microconnection without the Auto-dial feature. Some "smart" modems, such as the Hayes and the Radio Shack Modem II, allow you to dial the telephone by sending them characters from OMNITERM's normal mode. Check your modem manual to find out if yours has this feature and how to use it.

HAVING TROUBLE COMMUNICATING WITH COMPUSERVE??

All new CompuServe accounts assume that you are a Radio Shack Vidtex box, and not a computer. As a result it will do strange things on your screen such as eraseing all the characters along the left hand side of the screen. If this

is happening the cure is to tell CompuServe that you are an "other" type terminal. You can do this in the menu called "CompuServe User Information" and then choosing "Change Terminal Type". This will solve the problem.

TROUBLE BOOTING OR RUNNING OMNITERM??

Model III owners, did you CONVERT the OMNITERM disk to double density? OMNITERM is delivered on a Model I disk, and must be converted to double density with the TRSDOS CONVERT utility. You don't have to convert BASIC/CMD or other files that are duplicated on both disks since these are Model I TRSDOS programs that are already duplicated in Model III TRSDOS. Once you have the OMNITERM disk converted for your system, just type "OMNITERM" to run it.

CONFUSED ABOUT CONVERTING OMNITERM TO MODEL III??

Later versions of Model III TRSDOS do not have separate BACKUP/CMD, FORMAT/CMD, and other files, so CONVERT will not ask you about them. This is normal, and does not indicate any problem.

HAVING TROUBLE GETTING YOUR LYNX MODEM TO WORK??

OMNITERM is made to use the Lynx modem only when the Lynx is addressed to the same ports as the Radio Shack RS-232 board. Most Lynx modems are shipped from the factory set to a different address, appropriate for the Lynx communications software. If you want to use OMNITERM with your Lynx you must change the switch settings on the back of the Lynx to address it to the same location as the Radio Shack RS-232 board. This means that you cannot have both a Radio Shack RS-232 board and the Lynx (set to the same address) connected to your TRS-80 at the same time. You can have one or the other, but not both. For more information consult your Lynx manual, or contact the Emtrol (Lynx) factory.

HAVING TROUBLE WITH YOUR HAYES SMARTMODEM??

The Hayes can be dialed only with commands sent to it from the keyboard, as described in the Hayes manual. These commands must be sent in UPPER CASE letters, as the modem will not recognize lower case. Model I TRS-80's normally send uppercase characters unless you press the <SHIFT> key.

HAVING TROUBLE SENDING OR RECEIVING FILES??

You must press the <BREAK> key to go from command mode to normal mode after using the I INPUT TO BUFFER or O OUTPUT FROM BUFFER functions. These functions only set up the process of file transfer. Nothing will actually happen until you press <BREAK> and return to normal mode.

HAVING TROUBLE STARTING THE AUTO-SIGNON??

Be sure you are pressing the <DOWN-ARROW> and 0 (zero) key, not the letter O or the SHIFT key. Also make sure you have enough carriage returns and pauses in your auto-signon string.

HAVING TROUBLE WITH LDOS OR YOUR LNW-80 COMPUTER??

Make sure you have installed the appropriate patches, listed under "Getting Started". If you do not install these patches OMNITERM will not work properly with the LNW-80, or with LDOS on the Model III.

Appendix N - RS-232 Signal Conventions

In order to eliminate confusion, for the purposes of OMNITERM and this manual, the RS-232 signal conventions that we use are as follows:

When OMNITERM, or this manual, says an RS-232 control line is ON, this means that the line is HIGH, or set to a positive voltage. For the purpose of controlling RS-232 devices this means that the RS-232 line is active, or true. For instance if OMNITERM says that the CD (carrier detect) line is ON, this means that the modem is in fact detecting a carrier.

When OMNITERM, or this manual, says an RS-232 control line is OFF, this means that the line is LOW, or set to a negative voltage. For the purpose of controlling RS-232 devices this means that the RS-232 line is disabled, or false. For instance if OMNITERM says that the RI (ring indicator) line is OFF, this means that the telephone is NOT ringing, or at least that the modem doesn't think it is.

Appendix O - How OMNITERM dials the Telephone

OMNITERM uses the RTS control line to dial the telephone. Simply put, OMNITERM uses the Microconnection standard method for dialing the phone, where the phone line is off-hook when RTS is low (ON), and on-hook when RTS is high (OFF). OMNITERM simply brings the RTS line high for 62 milliseconds, and then low for 38 milliseconds, once for each click within a digit. It then waits 500 milliseconds with RTS still low (off-hook) for the between-digit period, and then continues with the next digit. When all digits have been dialed, OMNITERM returns the RTS line to the state it was set to when dialing was started. Normally this is ON (off-hook).

Appendix P - The Christensen Protocol

Complied, enhanced, and edited 1/23/84 by David Lindbergh from:
"Modem Protocol Overview", 1/1/82 by Ward Christensen
"XMODEM File Transfer Protocol", by D. L. Covill
from "Personal Systems", the San Diego Computer Society newsletter
"MODEM/XMODEM Protocol Explained", 1/8/80 by Kelly Smith
from "CP/M News", 1/81

This Appendix (and this Appendix ONLY) is in the PUBLIC DOMAIN.

Questions and comments on the Christensen protocol should be addressed to Ward Christensen, via CBBS/Chicago at (312) 545-8086.

Introduction - What is a protocol and why do we need one?

A protocol is, quite simply, the ritual courtesies exchanged by two separate computer systems in order to insure orderly and correct transfer of information. For short, interactive communications we can do without a formal protocol because the information transferred is visible, and if it looks wrong or we lose it, we can immediately ask for it again.

The situation is different when transferring data or program files. The volume to be transfrerred is greater, thus there is a greater chance of transmission errors. It's hard to tell whether an error has occured (such as in .COM files), and retransmitting the entire file would take too long and might only make things worse.

A protocol provides:

1) Positive control over the process and procedure
2) A means of detecting errors in the transmission process
3) The ability to retransmit any portion that has errors.

Note, however that BOTH parties to the transmission must use the SAME protocol, or the ritual doesn't work. It would be as if the two systems were each speaking different languages.

The Christensen protocol (also known as the MODEM, XMODEM, or CP/M protocol) was defined around 1979 by Ward Christensen as a public-domain protocol for use in hobbyist CP/M based systems. Since then it has become a de-facto standard used in many business and scientific applications with different microcomputer systems including IBM, Apple, and TRS-80, plus some mini and mainframe computer systems.

Outline

Data is sent in 128 byte numbered blocks, with a single checksum byte appended to each block. The receiving computer performs its own checksum as it aquires the incoming data, and upon completing each block compares its result with the checksum for the sending computer. If they match, it returns an ACK to the sender, meaning "recevied OK, send some more". If they don't match, a NAK is returned, meaning "that didn't look right, please send it

again". This process continues until the entire file has been transmitted (or the number of errors causes one of the parties to give up).

Received data is stored in memory, then written to disk every 16 blocks.

Definitions

ASCII	Meaning	Hex	Decimal	Control character
<SOH>	Start Of Heading	01H	1	Control A
<EOT>	End Of Text	04H	4	Control D
<ACK>	Acknolwledgement	06H	6	Control F
<NAK>	Negative Acknowledgement	15H	21	Control U
<CAN>	Cancel	18H	24	Control X

Transmission medium level protocol

Asynchronous, 8 data bits, no parity, one stop bit.

The protocol imposes no restrictions on the contents of the data being transmitted. No control characters are looked for in the 128-byte data messages. Absolutely any kind of data may be sent - binary, ASCII, etc.

Message block level protocol

Each block of the transfer looks like:

<SOH><block number><255-block number><128 data bytes><checksum>

in which:

<SOH> = 01 hex

<block number> = 8 bit binary number, starts at 01, increments by 1, and wraps 0FFH to 00H (not to 01)

<255-block number> = 8 bit ones complement of block number i.e. each bit complemented, as by 8086 "NOT" instruction

<128 data bytes> = 8 bits each

<checksum> = 8 bit value, least significant (toss any carry) 8 bits of the sum of all preceding bytes in the block, including the <SOH>, block number and complement, and all 128 data bytes.

The first three bytes (<SOH>, block number, and complement) of the block will be collectivly referred to as the "header".

On computing the checksum, note that a proper block number byte and its complement will always add to FFH, plus the 01H for the <SOH> will always add to 00H (least significant 8 bits). Therefore the checksum will always be

the same as it would be if only the 128 data bytes themselves were used to generate the checksum. However it is important to compute the checksum from ALL received bytes to verify header integrity. This will be discussed further below.

The last block sent is no different from the others. There is no "short block".

File level protocol

Common to both sender and receiver:

All errors are retried 10 times. After 10 errors a message is printed asking "retry or quit". If the user decides to "quit" the program sends a <CAN>.

CAN, an ASCII control-X, can be used to cancel the transmission.

The protocol may be considered "receiver driven", that is, the sender need not automatically re-transmit until the receiver requests it with a <NAK> or <ACK>, although it may (optionally) re-transmit upon timeout.

Receive program considerations

The receiver has a 10-second timeout. It sends a <NAK> every time it times out. The receiver's first timeout, which sends a <NAK>, signals the transmitter to start sending the first block. Optionally, the receiver could send a <NAK> immediately, in case the sender was ready. This would save the initial 10 second timeout. However, the receiver MUST continue to timeout every 10 seconds in case the sender wasn't ready.

Once into receiving a block, the receiver goes into a one-second timeout for each character and the checksum. If the receiver wishes to <NAK> a block for any reason (invalid header, timeout receiving data, bad checksum, framing or overrun error), it must wait for the line to clear. Since the receiver must wait for the whole block to be sent before sending a reply, there is no point in checking the header as it comes it. You might as well wait until you have the whole block and do it then. See "programming tips" for ideas.

Once the block is received, the receiver should run its own checksum on the data and header, and verify that it matches the received checksum. If it doesn't there must be an error in the block, so send a <NAK> to make the sender re-transmit the block.

Synchronizing: If a valid block number is received, it will be:

1) the expected one, in which case everything is fine;

2) a repeat of the previously received block. This should be considered OK, and only indicates that the receivers <ACK> got glitched, and the sender re-transmitted;

3) any other block number indicates a fatal loss of sync, such as the rare case of the sender getting a line-glitch that looked like an <ACK>. Abort the transmission, sending a <CAN>.

It is critical that the receiver both include the header in the checksum, plus do a separate sum of the header (should add to 00H) to insure that the block number was received correctly. If the block number is hit, it could cause a needless fatal loss of sync, as above. To check the header, sum the <SOH> (01H) with the block number plus the complement of the block number - this result must be zero for a proper transfer (e.g., for block 7, 01+07+F8 = 00).

If any sort of (non-fatal) error is detected in the received block, send a <NAK> to request the sender to re-transmit (up to 10 times).

Send program consideratons

While waiting for transmission to begin, the sender has only a single very long timeout, say one minute. In some implementations of the protocol, the sender has a 10 second timeout before retyring. This is NOT suggested, as it is preferable to let the protocol be completly receiver-driven. This will be compatible with existing programs.

After sending each block, the sender should dump any characters in the UART or other buffers that hold characters that came in while sending. This is in order to insure no noise on the line is interpreted as response from the receiver. The sender should then begin looking for the <ACK> from the receiver. If an <ACK> is received the sender procedes with sending the next block. If any other condition occurs (timeout, any character but an <ACK> received) the sender assumes an error occured and re-sends the same block, with the same block number and data. The block number is important because if the receiver in fact got the block OK and the <ACK> coming back got trashed the receiver can tell that it should ignore the repeated block because it was sent by mistake.

Commonly the data in the file will not end exactly with a full block, that is, the file will not be an exact multiple of 128 bytes in length. In these cases the last block's data must be padded with 0's to fill out the block. It is a characteristic of the Christensen protocol that it preserves end-of-file only as close as the next 128 byte block. This is a holdover from CP/M, which only kept track of EOF to the nearest 128-byte sector.

When the sender has sent all the blocks in the file, it sends an <EOT> instead of another block. It then waits for an <ACK>, resending the <EOT> if it doesn't get one. Again, the protocol could be receiver-driven, with the sender only having the high-level 1-minute timeout to abort.

Data flow example including error recovery

Here is a sample of the data flow, sending a 3-block message. It includes the two most common line hits - a garbaged block, and an <ACK> reply getting garbaged. <xx> represents the checksum byte.

SENDER		RECEIVER	COMMENTS
			Receiver times out after 10 seconds
	<--	<NAK>	Sends NAK on timeout
<SOH> 01 FE -data- <xx>	-->		Sends first block
	<--	<ACK>	ACKs first block
<SOH> 02 FD -data- <xx>	-->		Data line gets hit, block garbaged
	<--	<NAK>	Bad checksum, so NAKs trashed block
<SOH> 02 FD -data- <xx>	-->		Re-sends after NAK
	<--	<ACK>	Good checksum, so ACKs block
<SOH> 03 FC -data- <xx>	-->		Sends block 3
	<--	<ACK>	ACK gets garbaged
<SOH> 03 FC -data- <xx>	-->		No ACK rec'd, so re-sends block 3
	<--	<ACK>	ACKs repeated block
<EOT>	-->		No more blocks coming-- Sender done.
	<--	<ACK>	Receiver done too.

Programming tips

The character receive routine should be called with a parameter specifing the number of seconds to wait. The receiver should first call it with a time of 10, then <NAK> and try again, 10 times.

After receiving the <SOH>, the receiver should call the character receive subroutine with a 1-second timeout, for the remainder of the message and the <checksum>. Since they are sent as a continuous stream, timing out of this implies a serious error, like a glitch that caused, say, 127 characters to be seen instead of 128.

When the receiver wishes to <NAK>, is should call a "PURGE" subroutine, to wait for the line to clear. Recall the sender tosses any characters in its UART buffer immediately upon completing sending a block, to ensure no glitches were misinterpreted.

The most common technique for "PURGE" is to call the character receive subroutine, specifiying a 1-second timeout, and looping back to PURGE until a timeout occurs. The <NAK> is then sent, ensuring the other end will see it.

It is wise to add code to your character receive routine to set an error flag if the UART shows framing or overrun errors. This will help catch a few more glitches - the most common of which is a hit in the high bit of the the byte in two consecutive bytes. The <checksum> comes out OK since counting in 1 byte produces the same result of adding 80H + 80H as with adding 00H + 00H.

Glossary

<BREAK>
The BREAK key on the TRS-80 Keyboard.

<ENTER>
The ENTER key on the TRS-80 Keyboard.

Access
To access a function is to use it, to get at it and to execute or do it.

Acoustic Coupler
A type of modem that connects to the telephone by placing the handset in two rubber cushions so that the sound gets to and from the phone without any direct electrical connection to it. See also Modem.

Alt Mode
See ESCAPE.

ASCII
American Standard Code for Information Interchange. The communications code that the TRS-80 and most other computers use to transfer data. See the Appendix.

Baud
A measure of the speed of transmission of data. The baud rate tells how many bits per second are sent, although not all the bits contain data, since some of them are start, stop and parity bits. See also Bit.

Bit
The fundamental unit of information, a bit (binary digit) is a single 1 or 0 signal, a Yes or a No. There are 8 bits in a byte. See also Byte.

Boot
Boot is short for bootstrap, as in "to pull oneself up by one's bootstraps". It describes the process by which a computer or a computer program starts itself up. To boot a computer or program is to start it.

Break
See True Break.

Buffer
A storage area for information in the computer's memory. Data comes out of the buffer in the same order that it went in, so buffers are used for storing data that will not be processed immediately. OMNITERM has three main buffers, the Disk I/O Buffer, used for file transfers, the print buffer, used for holding characters the printer has not had time to print yet, and the scroll back buffer, which holds information that has scrolled off the top of the screen.

Byte
A unit of memory storage. A byte is eight bits in size and can hold any whole number value from 0 to 255. Normally, one byte will hold one character

of data. See also Character, Data Word, Bit.

CR
Short form of Carriage Return.

Carriage Return
A character with an ASCII value of 13, that is meant to cause the carriage of a Teletype or other printing terminal to return to the left side of the page. On the TRS-80, this causes the cursor to move to the beginning of the next line. See also Character, Data Word, Byte.

Character
Any single letter, number, punctuation mark, etc. that can be typed from the keyboard, put on the screen, or sent over the comm line. See also Data Word, Byte.

Comm Line
Communications line. The hardware interface that OMNITERM uses to connect with other equipment. See also RS232C Interface.

Command
Any action that causes OMNITERM to do something special. See also Function.

Configure
To change something, either hardware or software, to make OMNITERM act differently. To configure OMNITERM is to set it up so that it will do what you want it to do. For example, when you change the translation and special command tables for use with a particular computer, you have configured OMNITERM for that computer.

Control Key
The Down Arrow key on the TRS-80 keyboard. If another key is pressed while the control key is held down, the key will be sent through the Control Key table after going through the FROM Keyboard table. It's used to send special characters not already on the TRS-80 keyboard. See also Character, Byte.

CPU
Central Processing Unit. The essential part of a computer that runs programs and controls the rest of the computer. Often used as a shorthand way of saying "the computer".

Cursor
A small white block on the screen which is in the spot where the next character on the screen will be printed.

DOS
Disk Operating System. The program that loads in other programs, runs the disk drives, takes DIRectories, etc. Some common disk operatings systems for the TRS-80 are TRSDOS, NEWDOS and VTOS.

Data Word
A character as sent over the comm line. It consists of a character up to 8 bits in length, together with start, stop and parity bits, as sent by the UART. See also Byte, Character, Bit.

Dec
Short for Decimal.

Decimal
The usual numbering system used in everyday life. Decimal numbers are in base 10 and use the digits 0,1,2,3,4,5,6,7,8 and 9. See also Hex.

Default
The way things work if you do not specify otherwise. OMNITERM has default values for all functions and if you do not change them, they are what OMNITERM will use.

Disk I/O Buffer
The Input/Output buffer that OMNITERM uses to store information going out over the comm line or coming in over the comm line. This buffer can be loaded from a disk file and its contents can be saved to a disk file. See also Buffer.

Display
The video screen where information is shown.

Driver
The software that operates a particular device such as a disk drive, keyboard, or screen. The driver software translates the desired action (read a byte, print a character) into the specific function appropriate in a particular case.

Dumb Terminal
Sometimes called a glass Teletype. A dumb terminal is a device that simply displays on its screen everything sent to it, and sends each key as it is pressed. It has no functions more advanced than that.

Duplex
Can be either Half or Full. Half Duplex means that characters sent from OMNITERM are not echoed by the remote system and so OMNITERM must directly place these characters on the display itself. Full duplex means characters sent from OMNITERM are echoed back and OMNITERM need not send them to the display, since they will go there anyway when OMNITERM receives their echos.

Echo
To echo characters is to send back out everything that is received. OMNITERM can do this if remote machines need it and many timesharing systems will do this for OMNITERM so you can use Full Duplex. See also Character, Duplex.

ESCAPE
An ASCII code with a value of 27 that is usually used to precede a special sequence of bytes that perform some special function. By default, you can type an ESCAPE by pressing the Up-Arrow key on the TRS-80 keyboard. See also Escape Sequence, Character, Byte.

Escape Sequence
A special series of characters that are preceded by an ESCAPE and perform some special function. See also Character, ESCAPE.

Execute
To do a function, run a program, or perform some action.

Filename
The name by which a disk file is referred. In standard TRSDOS format the filename is 8 characters, a slash, a 3 character extention, a period, an 8 character password, a colon and a disk drive number. All but the original 8 characters are optional and all may be shorter than their maximum length. Also called a Filespec, or file specification. See also Character.

Filespec
See Filename.

Flag
An indicator that shows the state of something. See also Status Flag.

Full Duplex
The process where one computer echos all the characters the other sends. See also Echo, Duplex.

Function
Any action OMNITERM can take. Can be done by pressing a key, or processing a byte. See also Command.

HIMEM
A DOS memory location that stores the location of the last usable byte of memory. OMNITERM uses this to determine where its disk I/O buffer must end. If this value is changed correctly by other drivers you wish to use with OMNITERM, there will be no conflict between them. In the Model I, this is at 4049H. In the Model 3, at 4411H. See also Byte, Hex.

Half Duplex
The process where neither computer echos bytes to the other. Instead, each assumes the duty of displaying the characters it sends locally.

Hex
The base 16, or hexadecimal number system. Used often when working with computers because four bits, exactly half of a byte, fits into one Hex digit. Thus a byte in Hex is always two digits, of 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E and F, corresponding to the decimal numbers 0 through 15. Most high school math books explain the basis of place-value numbering systems and will teach you how to work with different bases. See also Decimal.

I/O Buffer
See Disk I/O Buffer.

LSTCHR
The location in memory where OMNITERM stores the last character it processed, previous to the current one.

LSTKEY
The location in memory where OMNITERM stores the last key that was typed.

LSTLOC
The location in memory where OMNITERM stores the last character generated locally from the keyboard or from the disk I/O Buffer.

LSTREM
The location in memory where OMNITERM stores the last character it received over the comm line.

LF
Short form of Line Feed.

Line Feed
An ASCII character with a value of 10 that is meant to cause the paper on a Teletype to move up to the next line. On the TRS-80, this character causes the cursor to go to the beginning of the next line.

Menu
A list of choices, from which you can pick one. Like a restaurant menu, menus are used by OMNITERM to give you a list of possible functions and allow you to choose among them.

Modem
A device that connects to your RS-232C interface and a telephone that translates the electrical pulses sent out over the interface into tones that are sent over the phone network so you can send data over long distances. The modem can also translate those tones back into pulses so it can receive data sent by another modem elsewhere. Most popular modems run at 300 baud, with a few as fast as 1200 baud. Modem stands for Modulator/Demodulator. See also Acoustic Coupler.

Page
Sometimes OMNITERM will refer to a page. This just means that you can fill the screen with other data which is considered another page of data.

Parity Bit
A bit that can be added to each data word to make the sum of the bits that are 1 in each word either an even or an odd number. This is used for error checking. See also Bit, UART, Data Word.

Print Buffer
The buffer that OMNITERM uses to store characters going to the printer that have not yet been printed. It is 2048 bytes in length. See also Buffer.

Prompt String
A series of characters that tell OMNITERM that it can transmit another line of text. See also String, Character.

RS-232C Interface
This is the hardware that connects your computer to outside equipment, and forms your comm line.

Screen
See Display.

Scroll
When the TRS-80 moves all the lines on the screen up one line, this is called scrolling the screen. This takes a certain amount of time to do, so at very high baud rates OMNITERM needs extra time at the end of each line.

Special Character
Any character not on the standard TRS-80 keyboard, such as the broken bar, tilde, square brackets, etc.

Start Bit
The bit at the beginning of each data word that marks the start of the word.

Status Flag
An item of data in OMNITERM's command mode that shows the current state of a particular function. Status flags tell you exactly how OMNITERM is configured at any given time.

Stop Bit
The bit or bits that mark the end of each data word. These bits last until the next word is sent, which may be an extremely long time. There will always be at least the number of bits set for Number of Stop Bits in the UART.

String
A series of characters one after another. In a string, or line of characters. See also Character.

Timesharing
When one large computer has many people using it at the same time, this is called timesharing, since the computer's time is shared among many users.

Toggle
To toggle something is to change the status of something that has only two states. If it is on, you can "toggle" it off, and if it is off when you "toggle" it, it will go on.

True Break
A true Break is a sort of extended 0 signal sent over the comm line that is usually used to get the attention of the computer connected to the comm line. In OMNITERM, the <BREAK> key on the keyboard will by default send a true Break.

UART
Universal Asynchronous Receiver Transmitter. The device in the RS-232C interface that serializes the data sent to it and attaches start, stop and parity bits to it. The UART also does the opposite of this operation when receiving data.

Wrap Around
When a word is printed at the end of the screen, the first letters will be on the line it started on, but the last ones will be at the beginning of the next line. This is called wrap around, because the words wrap around the edge of the screen.

Index