THIS USER'S MANUAL TELLS YOU HOW AUTOTERM TURNS YOUR TRS-80™ COLOR COMPUTER INTO THE WORLD'S SMARTEST TERMINAL

AND MOST LOVABLE

AUTOTERM CAN ALSO BE USED FOR SIMPLE WORD PROCESSING AND RECORD KEEPING

(TRS-80 is a trademark of Tandy Corporation)

DISK VERSION

NOW HI-RES SCREEN

PLUS XMODEM

EASY TO USE

ON-SCREEN EDITING via cursor. Full prompting and error checking. Key Beep and Error Beebop. Scroll bkwd/fwd while on line. Save/load files while on line. Maintain a disk copy of session. Automatic graphics. True lower case. Screen widths of 32, 40, 42, 50, 64. No split words on screen/printer. Print all or part of text. Search for strings.

PLEASANTLY POWERFUL

Total communications ability, 128 ASCII chars, 1200 baud, etc. Send text, graphics, BASIC, ML. Scan/Edit current data while receiving more data. Any modem. Fully supports D.C. Hayes and others. Any printer, page size, margins, etc. Override narrow text width of received data. Examine/change parameters, KSMs and disk directories at any time. Handles files which are larger than memory.

TRULY AUTOMATIC

Create, edit, print, save and load Keystroke Multipliers (KSMs). KSMs automate almost any activity. Dial via modem, sign-on, interact, sign-off. Perform entire session. Act as a message taker. KSM may include parameter changes, disk operations, editing, time delays, looping, execution of other KSMs, waiting for part-specified responses, branching based upon responses.

PXE Computing

11 Vicksburg Lane • Richardson, TX 75080 • 214/699-7273

IMPORTANT NOTICE

AUTOTERM and the AUTOTERM USER'S MANUAL are sold on an "AS IS" basis without warranty. PXE Computing shall have no liability or responsibility for indirect, special or consiquential damages caused by use of AUTOTERM, such damages including but not limited to loss of business or anticipated profits.
Good data processing procedure dictates that the program should be fully tested by the user in a way that adequately insures that results are satisfactory.

AUTOTERM

AUTOTERM USER'S MANUAL

COPYRIGHT NOTICE

First printing: Dec 1, 1983

PXE Computing
11 Vicksburg Lane
Richardson, TX 75080

AUTOTERM

AUTOMATIC TERMINAL PROGRAM
for the
RADIO SHACK COLOR COMPUTER

USER'S MANUAL

by
Philip Zwart

PXE Computing
11 Vicksburg Lane
Richardson, TX 75080

FEATURES

USER FRIENDLY

- Full prompting
- Easy cancellation of commands
- KEY BEEP indicates acceptance of keyboard entry
- KEY BEEBOP indicates inappropriate keyboard entry
- AUTOMATIC GRAPHICS
- Fast machine-code software
- RESET causes restart without loss of text

PLEASANT SCREEN DISPLAY

- Operating mode, free space, case flags on top line
- Prompting on next line(s)
- Text on remaining lines
- Upper/lower case as dark/light, light/dark, or light/light
- True upper/lower case is optional
- Screen width of 32, 40, 42, 50, or 64 characters
- Word wrap optional
- Scrolls backward & forward while scanning
- Scroll or wrap-to-top while receiving data
- Cursor form signals carriage returns & linefeeds

SIMPLE KEYBOARD OPERATION

- Rapid typing is acceptable
- Arrow keys move cursor & scroll at top or bottom
- Arrow keys repeat when held down
- Shift-zero toggles ALL CAPS
- CLEAR acts as ASCII control key
- Shift-CLEAR initiates all commands
- BREAK cancels any command

EASY ON-THE-SCREEN TEXT EDITING

- Single character CHANGE, DELETE, OR INSERT via cursor
- Search for substring

CONVENIENT OPERATING COMMANDS

- Memory ON/OFF during SEND/RECEIVE
- Print, save, delete all or part of memory
- Easily maintain a disk copy of the session
- Display disk directory
- Handle files larger than memory
- Enter any graphics character via hex code
- Send/receive text, BASIC, MACH LANG, PICTURE data, etc.
- Display/change user options

TOTAL COMMUNICATIONS ABILITY

- Baud rate - 110, 150, 300, 600, 1200
- Duplex - Full or Half
- Parity - Off, Mark, Space, Odd, or Even
- Stop bits - One, two, etc.
- Scan & edit text while receiving additional text
- ASCII control key, line break, & backspace

FEATURES

(Continued from previous page)

FLEXIBLE PRINTER CONTROLS

Page length, width, and all margins
Line spacing as single, double, triple, etc.
Word wrap is optional
Page pause is optional
Linefeed after carriage return is optional
Printer control codes may be embedded in text
Optionally override the narrow width of received text

ADVANCED KEYSTROKE MULTIPLIERS

Any sequence of any keys on keyboard
May include other keystroke multiplier keys
May include any system commands
May include time delays & looping

AUTOMATIC DIALOGUE ABILITY

Wait for next response from the other computer
Branching based on alternate responses
Self-test mode

REQUIRED EQUIPMENT

RADIO SHACK COLOR COMPUTER

Memory - 32K to 64K
DISK BASIC is required.

DISK DRIVE

Needed for loading AUTOTERM into computer. Optionally used to load &/or save text files, user's keystroke multiplier definitions, BASIC programs in ASCII or BINARY form, MACHINE LANGUAGE programs, PICTURE data.

OPTIONAL EQUIPMENT

MODEM & CABLE

Used to communicate with other computers over the telephone. Modem should provide asynchronous communication at one of the following baud rates: 110, 150, 300, 600, or 1200. Modem is not required when communicating directly with another computer via the serial port.

PRINTER & CABLE

Used to print formatted copies of text. AUTOTERM is compatible with any printer which operates with the Color Computer. AUTOTERM also accomodates those printers which fail to be compatible with the Color Computer because the printer requires a linefeed to be sent after each carriage return.

TABLE OF CONTENTS

TABLE OF CONTENTS

(Continued from previous page)

LIST OF FIGURES

LIST OF TABLES

LIST OF EXAMPLES

KEYSTROKE MULTIPLIERS

COMPUTER DIALOGUES

Modem II is a registered trademark of Tandy Corporation.

AUTOTERM

UPGRADE FROM DISK VERSION 3. TO DISK VERSION 4.

NEW FEATURES

BACKSPACE CHARACTER ALWAYS ACTIVATED

START/TRANSFER ADDRESSES NOT NEEDED FOR ML PROGRAMS

XMODEM PROTOCOL FOR ERROR FREE TRANSMISSION

1. BACKSPACE CHARACTER ALWAYS ACTIVATED

This version of AUTOTERM deletes the previous character whenever a BACKSPACE character is sent by you or received from the other computer. This is suppressed during XMODEM operation.

The previous version of AUTOTERM would permit a received BACKSPACE character to delete the previous character only when that BACKSPACE had been sent by you and echoed by the other computer within one-fifteenth of a second.

2. START/TRANSFER ADDRESSES NOT NEEDED FOR ML PROGRAMS

A MACHINE LANGUAGE program which has been received and is held in memory can be saved without knowing the START and TRANSFER ADDRESSES. Merely, use the cursor to define the START and STOP points of the data to be saved to disk.

Be careful to position the cursor precisely over the very first byte of the MACHINE LANGUAGE program when specifying the START point and precisely over the very last byte of the MACHINE LANGUAGE program when specifying the STOP point. You can check the results by loading the program back in by using the LOAD command. When AUTOTERM loads the program back into memory, it should recognize it as MACHINE LANGUAGE and precede it with the START and TRANSFER ADDRESSES as described in section 4.2.5 of the user's manual.

The previous version of AUTOTERM required you to insert the START and TRANSFER ADDRESSES ahead of any MACHINE LANGUAGE program which was being saved to disk. This was described in section 4.1.5 of the user's manual. This information is no longer required.

3. XMODEM PROTOCOL FOR ERROR FREE TRANSMISSION

The XMODEM protocol is used to overcome the problem of line noise causing received data to differ from the sent data. It is commonly used by bulletin boards and time share networks to download data. AUTOTERM is capable of receiving data via XMODEM or sending data via XMODEM. AUTOTERM does not perform repeated disk reading or writing during the XMODEM transmission itself. Before initiating XMODEM action, be sure that the unused memory buffer space (displayed in the upper right of your screen) is large enough to accomodate all the data being received or sent.

RECEIVING DATA VIA XMODEM

When you receive data, AUTOTERM is playing the role of the "receiving computer". The "sending computer" will require you to indicate that you wish to receive data via XMODEM. This usually involves a series of responses on your part.

For example, COMPUSERVE lets you select from several protocols when downloading data. You enter "1" to choose XMODEM protocol. Then COMPUSERVE indicates that it is ready to send by giving you the message "Starting XMODEM transfer".

If the "sending computer" is running AUTOTERM, then it will say "READY TO SEND VIA XMODEM".

In any case, once the "sending computer" is ready to send, then you should type <SHIFT-CLEAR>, and then type <X> for XMODEM. The top line of your screen will show:

```
XMODEM: TYPE S=SEND OR R=RECEIVE
```

Now type <R> for "RECEIVING". AUTOTERM will change your screen to show:

```
RECEIVING XMODEM DATA

BLOCK # 001
```

AUTOTERM will send a signal to the "sending computer" to tell it to start sending data. As data is received, the BLOCK # is increased and the data is displayed on the lower portion of your screen. The display form is the same as when ULD=1, except that CARRIAGE RETURN, [←] , and LINEFEED, [↓] , do not start a new line on the screen.

When the transmission is completed, AUTOTERM will be ready to save the data to disk. You will be asked for the filename. The screen will show:

```
RECEIVING XMODEM - TYPE FILENAME
& PRESS <ENTER>
```

At this point, you should type the filename to be used for saving the data to disk. If you do not specify an extension, then AUTOTERM will use /ATM for text data and /BIN for binary data, such as a MACHINE LANGUAGE program. For BASIC programs, you should specify /BAS as the extension.

If an error condition arises during the save to disk, then an error message will be displayed. When you press <BREAK> to continue, you will again be asked for a filename. This lets you try to save the data as often as you wish. If you are repeatedly unable to save the data and you don't want to try anymore, then press <BREAK> when asked for the filename. You will be returned to SEND/RECEIVE operation and your data will be held in memory.

If something goes wrong while the data is being sent to you, then press <BREAK> for a few seconds to leave XMODEM and return to SEND/RECEIVE operation. The data received up to that point will be retained in your memory buffer. The other computer may still be sending more data. If so, you can type <CLEAR-C>, i.e. an ASCII CONTROL C. This is usually recognized by the other computer as a signal to halt XMODEM operation.

SENDING DATA VIA XMODEM

When you send data, AUTOTERM is playing the role of the "sending computer". The "receiving computer" should be running AUTOTERM or some other terminal program which is able to handle receiving data via the XMODEM protocol.

When in SEND/RECEIVING mode, type <SHIFT-CLEAR> and then <X>. The top line of your screen will show:

```
XMODEM: TYPE S=SEND OR R=RECEIVE
```

Now type <S> for "SENDING". AUTOTERM will change your screen to show:

```
XSENDING A BLOCK OR FILE - TYPE
BLOCK # OR <C> OR <F>
```

You now specify what data is to be sent by giving a block # or using the cursor or giving a filename. This is done the same way as for transmitting a block or file. See section 3.6.1. of the user's manual.

After you have specified what data is to be sent, AUTOTERM will prepare for XMODEM transmission. You screen will show:

```
SENDING XMODEM DATA

BLOCK # 001
```

At the same time, AUTOTERM will send the message "READY TO SEND VIA XMODEM" to the other computer. When this message is received by the other computer, the operator of the other computer should trigger his machine to begin receiving via XMODEM. His computer will send a signal to tell AUTOTERM to start sending the data. As AUTOTERM sends the data, your screen will show the BLOCK # increasing and will display the data in the form used for ULD=1, except that CARRIAGE RETURN, ⊟ , and LINEFEED, ⊠ , do not start a new line on the screen. When the transmission is completed, AUTOTERM will return to SEND/RECEIVING mode.

You can halt the transmission and return to SEND/RECEIVING operation by pressing <BREAK> for a few seconds.

If the other computer sends an ASCII CONTROL C during the transmission, then AUTOTERM will halt the transmission and return to SEND/RECEIVING mode.

BRIEF DESCRIPTION OF XMODEM PROTOCOL

Under XMODEM protocol, the data is sent in a series of blocks of 128 bytes. A block is retransmitted whenever the "receiving computer" senses an error in the transmission of that block. Here are some of the details.

The "receiving computer" sends a NAK character (ASCII NEGATIVE-ACKNOWLEDGE = Hex 15) to tell the "sending computer" to start sending the first block.

The "sending computer" sends 132 bytes of data for each block. The first byte is the SOH character (ASCII START-OF-HEADING = Hex 01). The second byte is the block number. The third byte is the complement of the block number. The next 128 bytes are the data. And the 132nd byte is the checksum. It is calculated by adding up the 128 bytes of data and discarding any carry bits.

The "receiving computer" checks the data just received. The first byte should be a Hex 01, the second and third bytes should be the correct block number and complement. The 132nd byte, the checksum byte, is checked by independently recalculating a checksum byte using the 128 bytes of data received. If this recalculated checksum byte is the same as the checksum byte received from the "sending computer", then the data is assumed to be error free. If the checksum bytes disagree, then the data contains some errors.

If the 1st, 2nd, 3rd, and 132nd bytes are correct, then the "receiving computer" sends an ACK character (ASCII ACKNOWLEDGE = Hex 06) to signal that the next block should be sent. If any of the 1st, 2nd, 3rd, or 132nd bytes is incorrect, then the "receiving computer" sends a NAK character to signal that the previous block should be sent again.

The blocks of data are sent in the above manner until the entire set of data has been transmitted successfully. When the last block has been sent successfully, the "sending computer sends an EOT character (ASCII END-OF-TEXT = Hex 04) to signal that all the data has been sent. The "receiving computer" replies by sending an ACK, and the XMODEM transmission is completed.

If the last block of data does not contain exactly 128 bytes, then the "sending computer" makes it contain exactly 128 bytes by adding SUB characters (ASCII SUBSTITUTE = Hex 1A). These are discarded by the "receiving computer", after it has received the EOT character.

A successful XMODEM transmission does not actually gaurantee that the data has been sent without any errors. It is possible for the checksum bytes to agree even though the block contains errors. The chance of an erroneous block being accepted is one in two hundred and fifty-six.

ERROR CONDITIONS DURING XMODEM SENDING OR RECEIVING

AUTOTERM looks for several peculiar situations during XMODEM operation. These situations are taken as an indication that XMODEM operation has failed, and an error message is displayed on your screen. You are asked to type <BREAK> to trigger a return to SEND/RECEIVE operation.

Data received before the error condition occurs is retained in your memory buffer. You may wish to try to salvage this data and save it. If you don't want it then use the DELETE command (see section 2.7 of the user's manual) to eliminate it from memory.

Here are the error messages and their associated meanings.

TEN RE-TRANSMISSIONS - The current block of 128 bytes of data has been sent ten times and some type of error occurred every time. This could be caused by a very, very poor phone connection, or the block numbering became mismatched, or there may be some sort of incompatibility between AUTOTERM and the other program. You should try to do the whole transmission over again, rather than give up and assume incompatibility.

FIVE DELAYS OF 20 SECONDS - While receiving XMODEM data, AUTOTERM has sent an ACK or NAK and received no response for approximately 20 seconds, and this has happened five times. This could be caused by a broken connection or by the other computer having stopped its XMODEM operation.

NO REPLY FOR 30 SECONDS - While sending XMODEM data, AUTOTERM has received no ACK or NAK from the other computer for approximately 30 seconds. This could be caused by a broken connection or by the other computer having stopped its XMODEM operation.

EXCEEDS BUFFER - While receiving XMODEM data, the available space in your memory buffer has dropped below approximately 300 characters. You might try again after deleting all the data from your memory buffer. If this does not give you enough room, you may want to use the low-resolution version of AUTOTERM (LOADM"A64"), which gives you a larger memory buffer.

AUTOTERM

UPGRADE FROM DISK VERSION 4. TO DISK VERSION 5.

NEW FEATURES

KEYSTROKE MULTIPLIERS (KSM) CAN BE PLACED ON AUTOTERM DISK

AUTOMATIC LOADING OF SET OF KEYSTROKE MULTIPLIERS

AUTOMATIC EXECUTION OF DESIGNATED KEYSTROKE MULTIPLIER

TIMED EXECUTION OF AUTOTERM

PRINTER BAUD RATE IS NOW A USER PARAMETER

XMODEM OPERATION IS IMPROVED

J & M PARALLEL PRINTER PORT IS SUPPORTED

RADIO SHACK RS232 PAK IS SUPPORTED

RADIO SHACK MODEM PAK IS SUPPORTED

PRINTING WHILE ON LINE (J&M PORT, RS232 PAK, MODEM PAK)

OPERATION ON COLOR COMPUTER 3 VIA COLOR COMPUTER 2 MODE

1. KEYSTROKE MULTIPLIERS (KSM) CAN BE PLACED ON AUTOTERM DISK

The new AUTOTERM disk provides several grans of extra file space for data or other programs. It can be used to store your favorite KSMs. This enables you to take advantage of AUTOTERM's AUTO-LOAD and AUTO-EXECUTE capabilities described below.

To place your KSMS on the AUTOTERM disk; remove the protective tab, save your KSMs on the disk (see 4.1 in the User's Manual), and replace the protective tab.

CAUTION: Do not KILL or RENAME "A32", "A64", "A64H", or "AX". All four files are needed for AUTOTERM.

2. AUTOMATIC LOADING OF SET OF KEYSTROKE MULTIPLIERS

When AUTOTERM loads into your computer, it immediately looks on its own disk for a file named AKSM/KSM. If this file is found, it is automatically loaded into your KSM memory space.

Your KSMs become an integral part of AUTOTERM provided they are saved on the AUTOTERM disk under the filename AKSM/KSM. Whenever AUTOTERM is loaded, it will include your KSMs.

Be sure to use all upper-case letters in the filename AKSM/KSM. If the file name contains lower-case letters, it will not be automatically loaded by AUTOTERM.

3. AUTOMATIC EXECUTION OF DESIGNATED KEYSTROKE MULTIPLIER

When AUTOTERM loads your AKSM/KSM file, it looks at your very first KSM. If the first KSM is for "1" (1:==:), then AUTOTERM goes to EDITING TEXT mode and executes that KSM. If the first KSM is for "2" (2:==:), then AUTOTERM goes to SEND/RECEIVE mode and executes that KSM.

Your first KSM can execute other KSMs. This gives you a great deal of automatic start-up capability. By executing other KSMs, the first KSM could effectively set user parameters, print a reminder message, display a banner, dial the phone via an autodial modem, sign-on to a bulletin board, get messages, sign-off, hang-up, etc. All of the above or any part of it could be performed automatically when AUTOTERM is loaded.

4. TIMED EXECUTION OF AUTOTERM

The AUTOTERM disk contains ATIMER, a BASIC program for setting up delayed execution of AUTOTERM. To use this feature merely type RUN "ATIMER". You will be asked for the program name (A32, A64, or A64H), the time of execution, and the current time, which will be updated every second. When the execution time is reached, AUTOTERM will load and run. Pressing (BREAK) will cancel ATIMER.

CAUTION: Unattended operation involves some risks. You must weigh the convenience against the risks.

ATIMER will display a warning regarding the risks involved with unattended operation of a terminal program. Unfortunately, we can never be sure that AUTOTERM, a complicated program, would contain no bugs. Moreover, your computer, modem, or telephone could unexpectedly fail to operate properly. Or the telephone service could fail to properly route the phone call.

Some of the possible causes of failure are:

1. Program bug.
2. Computer memory chip errors.
3. Computer overheating.
4. Power interruptions or fluctations.
5. Modem malfunction.
6. Equipment at other end malfunctions.
7. Phone call mishandled by telephone service.
8. Phone call interrupted by incoming call or lifting of receiver at another extension.

Some of the undesirable results are:

1. AUTOTERM could execute at the wrong time or not at all.
2. An incorrect phone number could be repeatedly dialed. The party called would be very upset.
3. The time share session could fail to be properly terminated. Connect charges could be high.
4. The phone could be left connected for a long period. Long distance charges could be high.
5. The intended operation could fall into a loop and be repeated many times. The connect charges and/or long distance charges could be high.

5. PRINTER BAUD RATE IS NOW A USER PARAMETER

The following user parameter has been added:

PBD PRINTER BAUD RATE J&M, 300, 600, 1200, 2400, 4800, 7200, or 9600

When AUTOTERM loads into your computer, it sets PBD equal to the value currently held in your computer. You can change this parameter at any time.

6. XMODEM OPERATION IS IMPROVED

When sending data via XMODEM, AUTOTERM no longer sends the message "READY TO SEND VIA XMODEM". This message conflicted with some bulletin boards and with Compuserve's addition of the CRC option.

7. J & M PARALLEL PRINTER PORT IS SUPPORTED

If you have a J & M parallel printer port on your disk controller, then AUTOTERM will recognize it during start-up. In such a case, the Printer Baud parameter will be set to J&M. If you do not want to use the port, merely reset PBD to the desired buad rate to be used with your printer via the serial port. If you have a J & M parallel printer port and AUTOTERM fails to recognize it, then set PBD equal to J&M. This will cause AUTOTERM to try to use the parallel port.

When using the J & M parallel printer port, you can print while on line. See PRINTING WHILE ON LINE below.

8. RADIO SHACK RS232 PAK IS SUPPORTED

If you have a Radio Shack RS232 PAK connected, then AUTOTERM will use it for all communication. In the Multi-pak Interface, position the RS-232 PAK next to the Disk Controller PAK. Communication baud rate (CBD) can be 110, 150, 300, 600, 1200, 2400, 4800, 7200, or 9600.

When using the RS232 PAK, you can print while on line. See PRINTING WHILE ON LINE below.

9. RADIO SHACK MODEM PAK IS SUPPORTED

If you have a Radio Shack MODEM PAK connected, then AUTOTERM will recognize it during start-up. The MODEM PAK will be used for all communication.

CAUTION: The MODEM PAK will not transmit data when the CONNECT SWITCH button is UP. Be sure to avoid trying to send data in SEND/RECEIVE mode with this button UP. Wait until you have made the connection and pressed the button DOWN. If you try to send data with the button UP, then AUTOTERM will lock-up. If this happens accidentally, you can recover by pressing the RESET button on the rear of your computer, and returning to SEND/RECEIVE mode by typing "2" at the MAIN MENU.

When using the MODEM PAK, you can print while on line. See PRINTING WHILE ON LINE below.

10. PRINTING WHILE ON LINE (J&M PORT, RS232 PAK, MODEM PAK)

Use of the J & M Parallel Printer Port or the Radio Shack RS232 PAK or the Radio Shack MODEM PAK enables AUOTERM to print data while on line. When you are in SEND/RECEIVE operation, you can toggle the printer ON or OFF by the CLEAR-DOWN-ARROW, that is, while holding down the CLEAR key, press the DOWN-ARROW key. The upper right corner of the screen will show a "P" when the printer is ON.

Your keystroke multipliers can include toggling the printer ON/OFF. The wedge notation for the CLEAR-DOWN-ARROW is (CDA).

When AUTOTERM is printing on line, it will print data in the format in which it is received from the other computer. Your settings for printer baud rate, PBD, and printing LINEFEED after CARRIAGE RETURN, PLF, will be applied. However, all other printer parameters, such as margins and page length, will be ignored.

When the printer is turned on, it will receive characters after they have been shown on the screen. Your printer does not need to print as fast as the data is received. AUTOTERM sends characters to the screen as quickly as it can. Characters are sent to the printer only when the printer is ready and the screen is caught up. AUTOTERM keeps track of how far the printer is behind.

CAUTION: Although your printer need not print fast, it must be set to a high enough baud rate. If you are not using a J & M Parallel Printer Port, then you should have your printer baud, PBD, set at least twice as high as your communications baud, CBD. In particular, if you are communicating at 1200 baud, then your printer baud should be at least 2400. Otherwise, AUTOTERM, while sending a character to the printer, may miss a character that is coming from the other computer. A setting of PBD=9600 is the most desirable for printing while on line.

The printer can fall far behind when the high resolution screen display is being used. For faster screen action and more printer time, set ULD = 1, 2, or 3.

11. OPERATION ON COLOR COMPUTER 3 VIA COLOR COMPUTER 2 MODE

This version of AUTOTERM has all internal jumps, etc. set so that it is compatible with operation on the Color Computer 3 when running in the Color Computer 2 mode. When the Color Computer 3 is first turned on or reset, it is placed in a mode which imitates the Color Computer 2. Be sure that your machine is in this mode before loading AUTOTERM.

When exiting AUTOTERM from the MAIN MENU, do the following:

1. Type (SHIFT-BREAK) to leave AUTOTERM.
2. Type POKE 113,0 and (ENTER) to prepare for RESET.
3. Press the RESET button on the back of the computer.

This page intentionally left blank.

1. INTRODUCTION

The AUTOTERM program is devoted to two objectives:

1. EASY OPERATION - The user is guided by simple prompt messages and is quickly alerted to unacceptable entries. Screen display, etc. may be pleasantly tailored to the user's needs.
2. FLEXIBLE AUTOMATION - Automatic operation can be molded to each user's individual needs. Any series of keyboard actions, including time delays and alternative branching, may be automated.

Hopefully, the EASY OPERATION will enable inexperienced individuals to make use of AUTOTERM. The FLEXIBLE AUTOMATION should appeal to more advanced users. However, the automation may also benefit beginners. For example, an experienced AUTOTERM user can automate the sign-on dialogue for accessing a news service. A beginner could then make use of the automatic operation to sign-on to the news service. The beginner would not need to learn the sign-on procedure.

The overall operation of AUTOTERM is shown in figure 1.1, p. 9. The MAIN MENU enables you to select one of the three operating modes; TEXT PROCESSING, INTELLIGENT TERMINAL OPERATION, or KEYSTROKE MULTIPLIER DEFINITION. You can easily jump from one operating mode to another. Merely, use the <BREAK> key to trigger a return to the MAIN MENU, and then select the next desired mode. The RESET button also causes a return to the MAIN MENU.

Notice that TEXT PROCESSING and INTELLIGENT TERMINAL OPERATION share a common set of text, namely, the combination of all text and data accumulated during both operations. You can conveniently use the TEXT PROCESSING mode for off-line preparation of text or data, and then jump to INTELLIGENT TERMINAL OPERATION for going on-line and sending the previously prepared material. After completion of an on-line session, you can conveniently jump back to TEXT PROCESSING for editing and printing of selected portions of the session. As you jump between operating modes, all accumulated text and KEYSTROKE MULTIPLIER DEFINITIONS are retained in the computer.

The TEXT PROCESSING mode can handle most of your short word processing tasks. ON-THE-SCREEN editing makes it very easy to prepare error-free personal and business letters, school compositions, meeting summaries, etc. You can also create and edit your own calendar of events, club roster, log book, personal diary, etc. The results of your efforts can be printed and reprinted as desired, and can be preserved on disk. If you wish to use AUTOTERM in this manner, then read sections 2, 4.1.1, 4.2.1, and 5 of this manual. You can skip the other sections, which deal with on-line operation, special data files, and automatic operation. However, you may wish to also read sections 6.1 and 6.2, which show how to automate the typing of your commonly used phrases.

INTELLIGENT TERMINAL OPERATION is straightforward. You can immediately communicate with another computer. You should first

read sections 3.1, 3.2, and 3.3. Generally, one will find it more appealing to read through all of section 2, even though on line operation is the primary goal. If you intend to send &/or receive binary data, then be sure to read sections 3.7, 3.8, and all of section 4.

If you are a first-time user of timesharing, then you may have difficulty making connection with the other computer. Section 3.2 contains a list of trouble symptoms and possible causes.

The printed copy of an on line session usually has an undesirable format. Sign-on and sign-off data appears at the beginning and end. The body of the text is sprinkled with question/answer dialogue. There is no paging control to produce standard 11" pages. The left margin is non-existent. The right margin is excessive. Frequently, the output is formatted for a screen width of 40 or 32 characters. You may wish to edit the text before printing it. You may want to take a close look at the printer controls which are available in AUTOTERM. These are described in section 5.4. You should experiment with the OMIT CARRIAGE RETURNS option. With OCR=Y, your printer does not receive most of those CARRIAGE RETURNS which were sent by the other computer to accomodate a narrow screen display. This is described in section 5.3.

You may be a regular user of one or more computer services, such as Compuserve, Dow Jones News, The Source, local bulletin boards, etc. If so, KEYSTROKE MULTIPLIERS can speed-up your computer access. Automation of your sign-on and subsequent querries can eliminate some of the drudgery. It will certainly save you time and money. It may also enable others to perform your computer runs without learning all the details.

The KEYSTROKE MULTIPLIER system is described in sections 6 and 7. This material is more difficult to read than the other sections. It is worth the extra effort! Many examples are included. You can make use of the examples without fully understanding the technical details. You can use any example "as is" except for modification of key items, such as account number, password, and telephone number. For instance, if you have a Radio Shack Modem II, then use example 7.9 to acheive automatic dialing.

It may take a while for you to become familiar with all the features of AUTOTERM. Please don't hesitate to experiment. Use the "U" command to change various user options, and then observe the effects. Most users would rather learn by experience, than spend long hours studying the user's manual. AUTOTERM can readily accomodate the "learn-by-doing" approach.

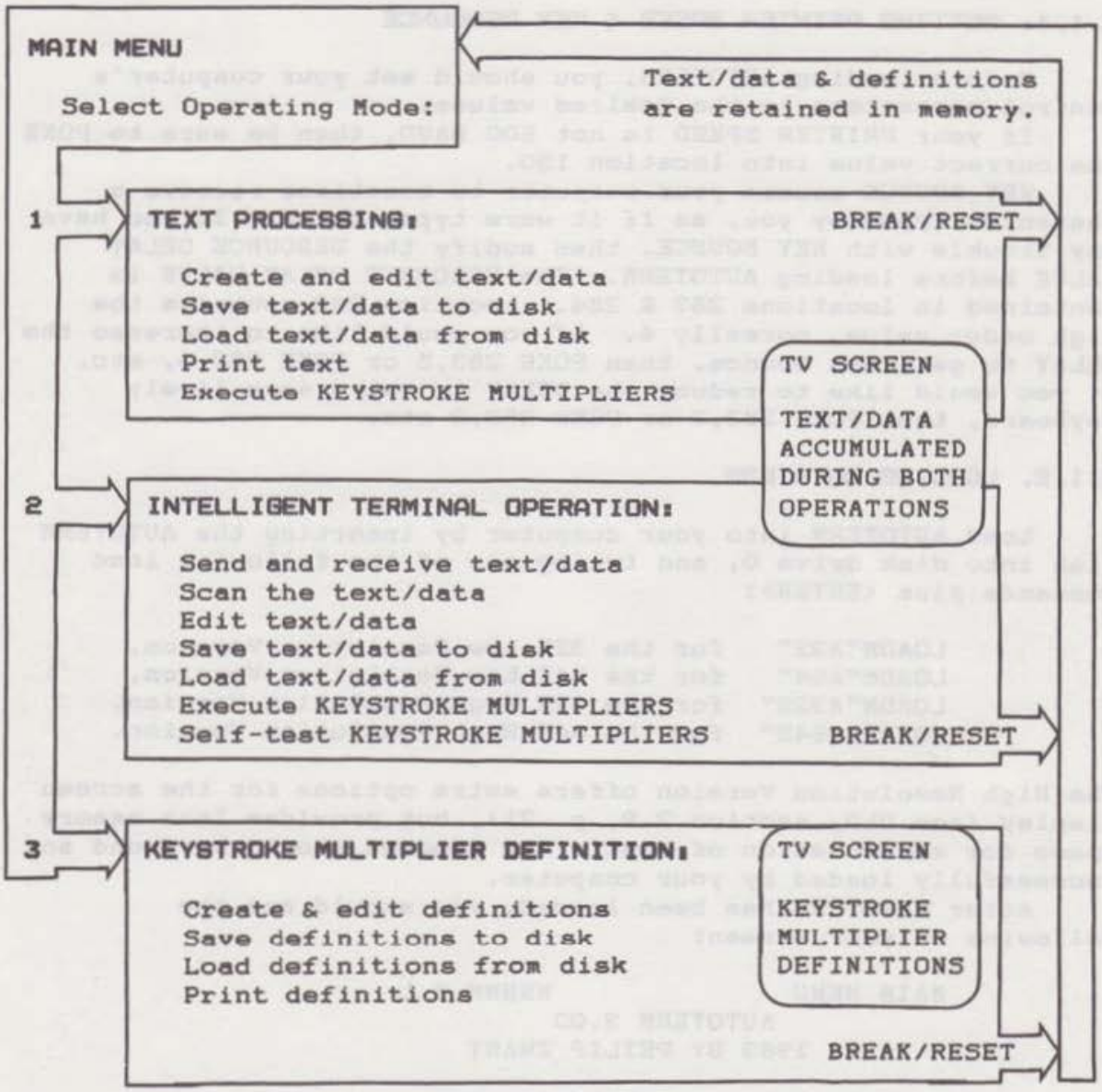

Figure 1.1
AUTOTERM OPERATION

2. TEXT PROCESSING: A STEP-BY-STEP EXAMPLE

2.1. GETTING STARTED

2.1.1. SETTING PRINTER SPEED & KEY DEBOUNCE

Before loading AUTOTERM, you should set your computer's control parameters to the desired values.

If your PRINTER SPEED is not 600 BAUD, then be sure to POKE the correct value into location 150.

KEY BOUNCE causes your computer to sometimes receive a character typed by you, as if it were typed twice. If you have any trouble with KEY BOUNCE, then modify the DEBOUNCE DELAY VALUE before loading AUTOTERM. The DEBOUNCE DELAY VALUE is contained in locations 283 & 284. Location 283 contains the high order value, normally 4. If you would like to increase the DELAY to get less bounce, then POKE 283,5 or POKE 283,6, etc. If you would like to reduce the DELAY to get a more lively keyboard, then POKE 283,3 or POKE 283,2 etc.

2.1.2. LOADING AUTOTERM

Load AUTOTERM into your computer by inserting the AUTOTERM disk into disk drive 0, and typing one of the following load commands plus <ENTER>:

```
LOADM"A32"    for the 32K Low Resolution Version,
LOADM"A64"    for the 64K Low Resolution Version,
LOADM"A32H"   for the 32K High Resolution Version,
LOADM"A64H"   for the 64K High Resolution Version.
```

The High Resolution Version offers extra options for the screen display (see ULD, section 2.9, p. 21), but provides less memory space for accumulation of text. The program should be found and successfully loaded by your computer.

After AUTOTERM has been loaded, you should see the following on your screen:

```
MAIN MENU                    NNNNN M L
          AUTOTERM 3.OD
    (C) 1983 BY PHILIP ZWART

<CLEAR> ACTS LIKE "CONTROL KEY"
<SHIFT-CLEAR> PRECEEDS COMMANDS
<BREAK> CANCELS COMMAND OR MODE
<RESET> RETURNS TO MAIN MENU
<SHIFT-RGHT-ARROW> INSERTS SPACE
<SHIFT-LEFT-ARROW> DELETES CHAR
OTHER ARROWS MOVE THE CURSOR

TYPE <1> OR <2> OR <3> FOR MODE:
1 = TEXT PROCESSING
2 = INTELLIGENT TERMINAL
3 = KEYSTROKE MULTIPLIER DEFN'S
```

Whenever the MAIN MENU is displayed, you can exit AUTOTERM by typing <SHIFT-BREAK>. The five digit number in the upper right tells you how many characters of memory are available for additional text storage. This number depends on your computer's total memory size.

Choose TEXT PROCESSING by typing the single keystroke <1>. Your screen should now show:

```
EDITING TEXT                NNNNN M L
```

At this point, you have no text accumulated in memory. The flashing cursor is waiting for action.

The "L" at the upper right of the screen signals that the keyboard creates LOWER CASE characters. Type <SHIFT-ZERO> to get a "U" for UPPER CASE ONLY. Type <SHIFT-ZERO> again to get back to LOWER CASE.

The "M" near the "L" says that SAVE-TO-MEMORY is turned ON. This means that all text is saved to memory as it is typed. You cannot turn this OFF unless you are in SEND/RECEIVE mode. See section 3.1, p. 23.

2.2. TYPING TEXT WITHOUT <ENTER>

The cursor is at the upper left. Type the following sentence without using the <ENTER> key (Don't worry about typing mistakes!):

Now IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID OF THEIR COUNTRY.

Notice that AUTOTERM automatically starts a new line on the screen when there is no more room on the current line. Your screen should show:

```
EDITING TEXT                NNNNN M L
NOW IS THE TIME FOR ALL GOOD MEN
 TO COME TO THE AID OF THEIR
COUNTRY.
```

The lower case letters, "ow", are displayed in reverse video. You have several options for the display of upper/lower case. See ULD in section 2.9, p. 21.

Notice that the word "COUNTRY" has been shifted to the third line. This shifting of words is done to avoid breaking a word between two lines on the screen. It is called WORD WRAP. This wrapping action can be supressed. See SWW in section 2.9.

Also, notice that the number in the upper right has been decreased. The text has been stored in memory and the amount of available space has been reduced.

If your speaker volume is turned up, then you heard a "BEEP" as each key was typed. The "KEY-BEEP" can be turned-off. See BEP in section 2.9. The KEY-BEEP may slow down your typing if you are extra fast.

2.3. MOVING THE CURSOR

Now let's move the cursor around on the screen. Type the <UP-ARROW> once. The cursor has moved up one line to the SPACE between "COME" and "TO". Type the <UP-ARROW> once again. The cursor is now on the top line over the "H". Now type the <UP-ARROW> once again. If your speaker is turned up, you heard the sound of "BOP". The cursor could not move upward from its present position. So your <UP-ARROW> was an unacceptable key and was ignored by AUTOTERM. The "BOP" was sounded to let you know about the key being ignored. This "BOP" can be turned off. See BOP in section 2.8, p. 20.

Next type the <LEFT-ARROW> once. The cursor has moved one space to the left and is positioned over the "T". Now hold down the <LEFT-ARROW> key. The cursor moves to the left until it reaches the left end of the line. Then you hear a series of BOPS until you stop pressing the key. Holding down any ARROW key is the same as typing that key repeatedly.

Next type one <DOWN-ARROW> to move the cursor to the second line. Now use the <RIGHT-ARROW> to move the cursor to the right end of the second line. Notice that when you move the cursor beyond the single SPACE which follows the word "THEIR", the cursor jumps to the start of the next line. The cursor can be positioned over only those characters which are in your text. You typed a single space after the word "THEIR". The 2nd, 3rd, and 4th SPACEs are not part of your text. They are present on the screen because the "COU" of "COUNTRY" was automatically moved to the 3rd line on the screen.

The <RIGHT-ARROW> can cause the cursor to jump from the right end of the current line to the left end of the next line. However, the next line must be a continuation of the current line. If a CARRIAGE RETURN or LINEFEED is pesent, then the <RIGHT-ARROW> is ignored. The <LEFT-ARROW> behaves in a similar manner.

You need to experience one more effect of the ARROW keys. Type <SHIFT-UP-ARROW>. That is, hold the <SHIFT> key down while typing the <UP-ARROW> key once. The cursor has moved to the start of the text. Now type <SHIFT-DOWN-ARROW>. The cursor has moved to the end of your text and is waiting for more action.

2.4. USING THE <ENTER> KEY

With the cursor positioned just after the ".", type the <ENTER> key. The cursor has moved to the left hand end of the fourth line. The <ENTER> key acts as a CARRIAGE RETURN. It causes the start of a new line.

If you are preparing text for eventual printing at your printer, then you would not need to use <ENTER> at the end of each printed line. When AUTOTERM prints your text, it will supply the CARRIAGE RETURNS which are needed to keep the text within the selected margins. The only time you should use the <ENTER> key, is when you want to force the start of a new line. You would use <ENTER> at the ends of paragraphs and when forcing blank lines between sections of text.

Let's generate some short lines of text. Type <4> <ENTER>, that is, type the key <4> and then type the key <ENTER>. Your screen now looks like this:

```
EDITING TEXT              NNNNN M L
NOW IS THE TIME FOR ALL GOOD MEN
 TO COME TO THE AID OF THEIR
COUNTRY.
4
```

Continue typing with <5> <ENTER> <6> <ENTER> and so on, all the way up to sixteen, that is, up to <1> <6> <ENTER>. Your screen now displays:

```
EDITING TEXT              NNNNN M L
COUNTRY.
4
5
6
7
8
9
10
11
12
13
14
15
16
```

Notice that the text has automatically scrolled upward to provide room for your next line. The original first two lines of text are being held in memory, even though they have been moved off the screen.

Let's move the cursor around some more. Hold down the <UP-ARROW>. The cursor moves line-by-line to the top of the screen, and then forces the text to scroll backwards until it has reached the top line of the text in memory. Now you can see those original two lines of text. But you can't see the line numbered "16". It has been moved off the bottom of the screen. It is being held in memory. Hold down the <DOWN-ARROW>. The cursor moves downward and eventually forces the text to scroll upward until the cursor is positioned at the next available space for entering text. That is, your screen is back to looking like our last illustration of the screen display.

Now type the <UP-ARROW> one time. Next type the <RIGHT-ARROW> one time. The cursor is positioned over the "6" on line 16. There is another character saved in memory after the "6". It is a CARRIAGE RETURN which was generated when you typed <ENTER> after <1> <6>. You can see it by typing one more

<RIGHT-ARROW>. The cursor does not move to the right. It stays in the same location. However, it flashes more slowly and takes the form - . The cursor is now positioned over the CARRIAGE RETURN rather than the "6". Type the <LEFT-ARROW> one time. The cursor has been re-positioned over the "6". It is flashing faster and has its normal form.

Let's add a little more to the text. Type the <DOWN-ARROW> to move the cursor to the next line. Notice that it automatically goes to the start of the line, because there is no text on the line. Type the following:

```
THIS IS THE END OF THE EXAMPLE.<ENTER>
```

Your screen now looks like this:

```
EDITING TEXT            NNNNN M L
4
5
6
7
8
9
10
11
12
13
14
15
16
THIS IS THE END OF THE EXAMPLE.
```

2.5. FINDING A STRING OF CHARACTERS

If you have a large amount of text, then scrolling is a slow way to look for something. The "FIND" command enables you to quickly find any string of characters. Suppose that we would like to find the string "15".

Type <SHIFT-UP-ARROW> to position the cursor at the top of the text. Now type <SHIFT-CLEAR>, that is, hold down the <SHIFT> key and type <CLEAR>. AUTOTERM has been placed in the COMMAND MODE, and is waiting for you to type the key for the desired command. The top of the screen shows:

```
TYPE COMMAND OR <?>     NNNNN M L
```

Notice that the top line is telling you that AUTOTERM is waiting for you to type the single key for the desired command or the key <?>. If you type <?>, the screen will display a list of the AUTOTERM commands, and AUTOTERM will wait for you to type <BREAK> to return to your text.

The <BREAK> key can be used to cancel any command at any

time before its completion.

Type the key <F>, for the FIND command. The top two lines of the screen now show:

```
FIND COMMAND - TYPE STRING
& PRESS <ENTER>
```

Type <1> <5> & <ENTER>. AUTOTERM will search through the text until it finds the string "15". It then positions the cursor over the first character of the string.

When using the FIND command, you can specify as many as fourteen characters. AUTOTERM will start the search at the current cursor position. If the string is not found, then the cursor will be positioned at the end of the text and a BOP will be sounded. If you type <ENTER> without typing any string, then AUTOTERM will use the previously specified string. This lets you conveniently search for each occurence of a string.

The ASTERISK may be used to indicate a wild card character. For example, the string J*INGLE would cause AUTOTERM to stop at JANGLE, JINGLE, & JUNGLE, as well as JBNGLE, JCNGLE, etc.

Your string may include the CARRIAGE RETURN and/or LINEFEED characters. Merely, represent them by typing the five characters <ENT> or <LFD>, respectively. This is the WEDGE NOTATION, discussed more thoroughly in section 6, p. 46.

2.6. CHANGING, DELETING, & INSERTING

First, we'll do some CHANGING of the text. The cursor should be positioned over the "1" of "15". Now type the key <2>. The "1" has been changed to a "2", and the cursor has moved to the next character of text, the "5". Now type the <DOWN-ARROW> two times, so that the cursor is positioned over the "H" of "THIS". Type the three keys <I> <M> <E>. The word "THIS" has been changed to "TIME", and the cursor is positioned over the next character of text, the SPACE.

Your screen should now show:

```
EDITING TEXT            NNNNN M L
4
5
6
7
8
9
10
11
12
13
14
25
16
TIME IS THE END OF THE EXAMPLE.
```

Now let's DELETE a character or so. Type <SHIFT-LEFT-ARROW>, that is, hold down the SHIFT key and type the <LEFT-ARROW> key. The SPACE has been deleted, and the cursor is positioned over the next character, the "I". Now hold down the <SHIFT-LEFT-ARROW> for a short while. Several characters have been deleted, because holding down the arrow is like typing it repeatedly. We will assume that "IS THE END" has been deleted.

You can delete a CARRIAGE RETURN, just as easily. Use the <LEFT-ARROW> to move the cursor over to the left end of the line. Your screen should show:

```
EDITING TEXT                NNNNN M L
4
5
6
7
8
9
10
11
12
13
14
25
16
TIME OF THE EXAMPLE.
```

Type the <UP-ARROW> two times. The cursor is now over the "2" of "25". Type the <RIGTH-ARROW> once to put the cursor over the "5" of "25". Now type the <RIGHT-ARROW> one more time to put it over the CARRIAGE RETURN. The cursor should be flashing slowly and have the form ⊟ . Now type <SHIFT-LEFT-ARROW>. The CARRIAGE RETURN has been deleted. Your screen looks like this:

```
EDITING TEXT                NNNNN M L
4
5
6
7
8
9
10
11
12
13
14
2516
TIME OF THE EXAMPLE.
```

The cursor is positioned over the "1" of "2516".

Finally, let's do some INSERTING. Type <SHIFT-RIGHT-ARROW> one time. A SPACE has been inserted ahead of the "1", and the cursor is positioned over the SPACE. Hold down the <SHIFT-RIGHT-ARROW> for a short while. More SPACES have been inserted. We will suppose that five more SPACES have been inserted, giving a total of six SPACES between the "5" and the "1". The cursor is positioned at the start of these SPACES. Now type the keys <6> <6> <7> <7> <8> <8>. Your screen shows:

```
EDITING TEXT                NNNNN M L
 TO COME TO THE AID OF THEIR
COUNTRY.
4
5
6
7
8
9
10
11
12
13
14
25667788[1]5
TIME OF THE EXAMPLE.
```

Notice that inserting text involves two operations, namely, inserting SPACES and changing the SPACES to the desired text.

Now let's insert a CARRIAGE RETURN after the "7" and before the "8". Type the <LEFT-ARROW> two times, to position the cursor over the "8" of "78". Type <SHIFT-RIGHT-ARROW>, to insert a SPACE. Type <ENTER>, to change the SPACE to a CARRIAGE RETURN. Your screen now displays:

```
EDITING TEXT                NNNNN M L
COUNTRY.
4
5
6
7
8
9
10
11
12
13
14
256677
[8]816
TIME OF THE EXAMPLE.
```

2.7. DELETING A BLOCK OF TEXT

If you want to delete a large section of text, you can use the "DELETE BLOCK/FILE" command. We will delete the following block of text:

```
8816
TIME O
```

Type <SHIFT-CLEAR>, and then <D> for the DELETE BLOCK/FILE command. The top two lines of the screen show:

```
DELETING A BLOCK OR FILE - TYPE
BLOCK # OR <C> OR <F>
```

The text does not contain any block labels, so you cannot type a block number. The <F> is for deleting files from the disk. We want to use the cursor to specify the block. Type <C>. The top two lines now show:

```
DELETING A BLOCK OR FILE - PUT
CURSOR AT START & TYPE <ENTER>
```

Follow the instructions. Use the ARROW keys to position the cursor over the first "8" of "8816". Now type <ENTER>. The top two lines of the screen now show:

```
DELETING A BLOCK OR FILE - PUT
CURSOR AT STOP  & TYPE <ENTER>
```

Use the ARROW keys to position the cursor over the "O" in "TIME OF". Type <ENTER>. The block is deleted and the screen now shows:

```
EDITING TEXT              NNNNN M L
 TO COME TO THE AID OF THEIR
COUNTRY.
4
5
6
7
8
9
10
11
12
13
14
256677
F THE EXAMPLE.
```

Now let's add a pair of BLOCK MARKERS and use them when deleting a block. Use the ARROW keys and the

<SHIFT-RIGHT-ARROW> to insert :START1: and :STOP1: so that your screen displays:

```
EDITING TEXT                NNNNN M L
 TO COME TO THE AID OF THEIR
COUNTRY.
4
5
6
7
8
9
10
11
:START1:12
13
14
2:STOP1:56677
F THE EXAMPLE.
```

The markers say that BLOCK #1 starts at the "1" of "12" and stops at the "2" of "256677".

Now type <SHIFT-CLEAR>. Then type <D> for the command key. The first two lines show:

```
DELETING A BLOCK OR FILE - TYPE
BLOCK # OR <C> OR <F>
```

Now type the key <1>, to delete block #1. Your screen now shows:

```
EDITING TEXT                NNNNN M L
NOW IS THE TIME FOR ALL GOOD MEN
 TO COME TO THE AID OF THEIR
COUNTRY.
4
5
6
7
8
9
10
11
56677
F THE EXAMPLE.
```

Notice that the BLOCK MARKERS have been deleted along with the block.

2.8. CONTROL CHARACTERS & GRAPHICS CHARACTERS

The ASCII coding applies to 128 characters. Some of these are rarely used and are usually not seen in text. Table 8.1, pp. 76-77, shows how each ASCII character is typed at the keyboard and displayed on the screen.

Notice that some characters are typed by using the <CLEAR> key. Most of these are the ASCII CONTROL characters. Such characters have special meaning. For example, the FORMFEED character (FF) causes the start of a new page during printing. If you want to add a FORMFEED to your text, then type <CLEAR-L>. That is, hold down the <CLEAR> key and type the <L> key. Any FORMFEED contained in your text will be displayed on your screen as , a COMMA with reverse background.

Text can also involve 128 other characters, which are thought of as GRAPHICS characters. They can be added to your text by use of the HEX CODE command.

Suppose you would like to add a graphics character having color number 5 (buff) and pattern number 9 (checkered). The numeric code for this graphics character is 201, that is, 128+16*(5-1)+9. The equivalent two-digit hexadecimal number is C9. Here is what you do to enter the hexadecimal code C9.

Position the cursor at the point where you want to put the graphics character and type <SHIFT-CLEAR>. AUTOTERM will ask for the command key. Respond by typing the key <H> for HEX CODE. The top line of your screen will display:

TYPE TWO HEX DIGITS & <ENTER>

At this point, you should type the keys <C> <9> and <ENTER>. The graphics character for C9 will be put into your text.

The graphics character CC can be used to embed printer control codes within the text. This is described in section 5.3, p. 43.

2.9. USER OPTIONS

You can change certain aspects of AUTOTERM operation. Your "options" are categorized as GENERAL, COMMUNICATIONS, & PRINTING. The COMMUNICATIONS options are described in section 3.9, p. 31, and the PRINTING options in section 5.4, p. 45. Here are the GENERAL options. Each item is preceded by its three-letter abreviated code. The codes are ordered alphabetically.

BEP KEY-BEEP Y or N Preset to Y

BOP KEY-BOP Y or N Preset to Y

SWW SCREEN-WORD-WRAP Y or N Preset to Y

WORD WRAP avoids breaking a word at the end of a line on the screen. If the word does not fit on the line, then the entire word is shifted to the next line.

ULD UPPER/LOWER CASE DISPLAY 1 to 3 Preset to 1

The display of upper/lower case is affected by ULD. Values 4 thru 8 are available with the High Resolution Versions of AUTOTERM. They use the high resolution screen to display upper/lower case. These modes result in slower scrolling and keyboard action.

1 = Normal background for upper case. Reverse background for lower case.

2 = Normal background for upper and lower case. Screen does not distinguish between upper and lower case.

3 = Reverse background for upper case. Normal background for lower case.

4 = True upper/lower case with screen width of 32 chars.

5 = True upper/lower case with screen width of 40 chars.

6 = True upper/lower case with screen width of 42 chars.

7 = True upper/lower case with screen width of 50 chars.

8 = True upper/lower case with screen width of 64 chars.

You can examine and/or change the setting of any of these by the "USER OPTION" command. For example, suppose you'd like to change the screen display to type 3, that is, reverse background for upper case and normal background for lower case.

Type <SHIFT-CLEAR>. Next, type <U>. The screen shows:

```
EDITING TEXT                NNNNN M L
TO CHANGE ANY OPTION TYPE ITS
3-LETTER CODE & <ENTER> □

GENERAL:
BEP=Y   BOP=Y   SWW=Y   ULD=1

COMMUNICATIONS:
IGD=Y   IGL=Y   IGN=Y   SCR=Y
CBD=300   DUP=FULL   PAR=EVEN
STP=1     XNF=OOOO   SLF=N

PRINTING:
PGL=66  PGW=80  PGP=N   LSP=1
TMG=6   BMG=6   LMG=6   RMG=6
CNP=Y   PLF=Y   PWW=Y   OCR=N
```

While changing an option, you can press <BREAK> at any time, to return to your text with no change. Now type the three letters <U> <L> <D>. If you make a typing error, you can use the <LEFT-ARROW> to backspace and cancel typed characters. Next press <ENTER>.. After you have typed the <ENTER>, AUTOTERM will check that you have typed an acceptable code. If not, you will hear a "BOP", and the cursor will remain placed after your third character. If you have typed <U> <L> <D> <ENTER> properly, then the screen shows:

```
EDITING TEXT                    NNNNN M L
CHANGING AN OPTION - TYPE ITS
NEW VALUE & <ENTER>

GENERAL:
BEP=Y   BOP=Y   SWW=Y   ULD=

COMMUNICATIONS:
IGD=Y   IGL=Y   IGN=Y   SCR=Y
CBD=300   DUP=FULL   PAR=EVEN
STP=1     XNF=0000   SLF=N

PRINTING:
PGL=66  PGW=80  PGP=N   LSP=1
TMG=6   BMG=6   LMG=6   RMG=6
CNP=Y   PLF=Y   PWW=Y   OCR=N
```

AUTOTERM is waiting for you to fill-in the new value for ULD. Now type <3> and <ENTER>. Your screen will show your text again. The text is displayed in mode 3 in accordance with the new setting for UPPER/LOWER CASE DISPLAY.

3. INTELLIGENT TERMINAL OPERATION

3.1. BEFORE MAKING THE CONNECTION

AUTOTERM is put into INTELLIGENT TERMINAL operation by the selection of mode #2 from the MAIN MENU. See section 2.1, p. 10, for instructions on loading AUTOTERM and using the MAIN MENU. After you have selected mode #2, the top line of your screen should show:

SEND/RECEIVING NNNNN M L

AUTOTERM is now keeping track of the serial I/O port. Anything received over the port will be displayed on your screen. You have not yet connected to the computer. So nothing should be happening on your screen. If characters are being accumulated on your screen, then some piece of equipment is sending data (or what looks like data) to your serial I/O port. Perhaps your printer is still connected to the port, or your modem is picking up noise from somewhere and sending signals.

The M at the upper right of the screen indicates that SAVE-TO-MEMORY is turned ON. All characters being sent or received are saved in memory as accumulated text. Use the <CLEAR-UP-ARROW> to toggle the SAVE-TO-MEMORY between ON and OFF. Usually you will want to have it set to ON.

Before actually tying into the other computer, you may wish to set some communications parameters. The most common parameters, which might concern you, are BAUD RATE, DUPLEX, and PARITY. See section 3.9, p. 31, for a description of these parameters and a few others. You can change their settings by using the "USER OPTIONS" command. An example of this command is given in section 2.9, p. 20. If you don't know the proper values for these parameters, then leave them as they are. They are set to the most likely values.

If you are using a modem, then its DUPLEX switch, if present, should be set to FULL. This will supress any echoing of characters by the modem. If the other computer does not echo the characters you send, then you will not see any display of the characters you type. In such a case, set AUTOTERM's DUPLEX parameter to HALF, that is, DUP=HALF.

3.2. MAKING THE CONNECTION

Now try to establish connection with the other computer. Dial the other computer if necessary. If the other computer expects to receive certain characters from you, then type these characters. Some computers expect to receive one or more CONTROL CHARACTERS. Use the <CLEAR> key as a CONTROL key. Other computers will send you some greeting before expecting you to send anything. If this is the case with the computer you are contacting, then simply wait to see the other computer's greeting on your screen.

Be sure to send the correct characters by typing them correctly. In particular, if the other computer expects to

receive all upper case, then you should have the SHIFT-LOCK turned on. That is, the letter "U" should be present in the upper right hand corner of your screen. If an "L" appears there, then type <SHIFT-ZERO> to change to "U" for UPPER-CASE-ONLY.

If you wish to send a "LINE BREAK", then type <SHIFT-BREAK>. Don't type <BREAK> without the <SHIFT>. If you type <BREAK>, you will be taken back to the MAIN MENU. If this happens accidentally, then immediately type <2> to get back to INTELLIGENT TERMINAL operation. Your accumulated text will not be lost from memory, and you will still be connected to the other computer.

Everything sent by the other computer is shown on your screen. Everything you type is displayed on your screen and is sent to the other computer. The <LEFT-ARROW> key sends a BACKSPACE to the other computer. The other ARROW keys, the <BREAK>, and the <SHIFT-CLEAR> affect AUTOTERM, but do not cause anything to be sent to the other computer.

If you are not sending and receiving as described above, then something has gone wrong. Here are a few possibilities:

SYMPTOM - The carrier light on your modem is not turning-on. You don't seem to be making contact with the other computer.
CAUSE #1 - The other computer is not answering the phone. Check this by using a normal hand held telephone to dial the number. You should hear a shrill tone when the other computer answers.
CAUSE #2 - Your modem is set to ANSWER when it should be set to ORIGINATE. Look for an ANS/ORIG switch on your modem. If present, it should probably be set to ORIG.
CAUSE #3 - Your modem is set to the wrong speed or BAUD RATE. Look for a BAUD RATE switch or a CHAR/SEC switch. If present, it should probably be set to 300 BAUD or 30 CHAR/SEC.
CAUSE #4 - Your modem is not turned ON, or not plugged into the electrical outlet.

SYMPTOM - The carrier light on your modem is turned on, but the other computer is not responding to your initial sign-on characters. It's as if the other computer is receiving nothing from you or is receiving the wrong characters.
CAUSE #1 - AUTOTERM's PARITY may be incorrectly set. Try changing the PAR to other options. See PAR, p. 31. If PAR=EVEN is not working, then the next best bet is PAR=MARK.
CAUSE #2 - AUTOTERM's COMMUNICATION BAUD RATE may be incorrectly set. Try changing the CBD to other options. See CBD, p. 31. CBD=300 is the most common setting.

SYMPTOM - Characters sent by the other computer are being received correctly, but characters typed by you are apparently misunderstood by the other computer.
CAUSE - AUTOTERM's PARITY is incorrectly set. Try changing the PAR to other options. See PAR, p. 31. If PAR=EVEN is not working, then the next best bet is PAR=MARK.

SYMPTOM - Characters sent by the other computer are being received correctly by you and characters typed by you are apparently being received correctly by the other computer. However, the characters typed by you are not being displayed on your screen.
CAUSE - AUTOTERM's DUPLEX is set to FULL, DUP=FULL, when it should be set to HALF, DUP=HALF. Change the setting. See DUP, p. 31.

SYMPTOM - Characters sent by the other computer are being received correctly by you, and characters typed by you are apparently being received correctly by the other computer. However, characters typed by you appear double on your screen.
CAUSE #1 - AUTOTERM's DUPLEX is set to HALF, DUP=HALF, when it should be set to FULL, DUP=FULL. Change the setting. See DUP, p. 31.
CAUSE #2 - Your modem's DUPLEX switch is set to HALF, when it should be set to FULL. Look for a DUPLEX switch on your modem. If you find it, then set it to FULL.

SYMPTOM - Characters sent by the other computer are sometimes displayed as graphics block characters on your screen.
CAUSE - AUTOTERM's PARITY is incorrectly set to OFF, PAR=OFF. Try changing PAR to other options, such as MARK, PAR=MARK. See PAR, p. 31.

3.3. SCANNING WHILE CONNECTED

As your session with the other computer progresses, your text will be moved off the screen. It will be retained in memory, provided the SAVE-TO-MEMORY has been left ON. You may wish to review some of the previous portions of the session. AUTOTERM enables you to scan the entire session without losing contact with the other computer. Moreover, characters sent by the other computer while you are scanning, will be accumulated in memory. These additional received characters will be displayed on your screen when you return to SEND/RECEIVING operation.

Enter the SCAN/RECEIVING mode by using the <UP-ARROW> or <SHIFT-UP-ARROW> to move the cursor away from the end of the text. If SAVE-TO-MEMORY is OFF, then you must use the <SHIFT-UP-ARROW>. The top line of the screen will show:

```
SCAN/RECEIVING          NNNNN M L
```

You can use the ARROW keys to move the cursor anywhere in the accumulated text. You can scan any part of the session.

While SCAN/RECEIVING, you can make use of other operating modes. You are able to edit the accumulated text. You can use the disk drive to save all or part of the text, or to load additional text. See section 3.5, p. 26.

You cannot send characters to the other computer while in SCAN/RECEIVING operation. However, you can exit SCAN/RECEIVING and return to SEND/RECEIVING operation by positioning the cursor

beyond the end of text. This is most easily done by typing <SHIFT-DOWN-ARROW>.

3.4. EDITING WHILE CONNECTED

You may desire to delete, add to, or change some of the accumulated text without losing contact with the other computer. AUTOTERM enables you to do this via EDIT/RECEIVING operation. You can edit the accumulated text. Your editing commands will not be sent to the other computer. The characters sent by the other computer while you are editing, will be accumulated in memory.

While in SEND/RECEIVING or SCAN/RECEIVING operation, type <SHIFT-CLEAR>. AUTOTERM will ask for your command. Type <E>. The top line of your screen will show:

```
EDIT/RECEIVING          NNNNN M L
```

Now you can edit the text in the various ways described in section 2.6, p. 15.

During EDIT/RECEIVING operation, the ARROW keys act the same as in text processing. In particular, the <SHIFT-LEFT-ARROW> deletes characters, the <SHIFT-RIGHT-ARROW> inserts SPACES, and the <SHIFT-UP-ARROW> moves the cursor to the start of the text. The <SHIFT-DOWN-ARROW> positions the cursor immediately after the end of the text. This does not take you out of EDIT/RECEIVING operation.

You can leave EDIT/RECEIVING operation by typing <BREAK> one time. This will place AUTOTERM into SEND/RECEIVING or SCAN/RECEIVING operation. To get back to SEND/RECEIVING operation, you may need to type <SHIFT-DOWN-ARROW>, depending on whether or not the cursor is positioned beyond the end of text.

You can easily delete a block of text while SEND/RECEIVING or SCAN/RECEIVING. Merely, type <SHIFT-CLEAR> and then type <D>. See section 2.7, p. 18, for a complete description of the DELETE BLOCK command. During the deleting, characters sent by the other computer will be accumulated in memory, as occurs with SCAN/RECEIVING and EDIT/RECEIVING.

3.5. USING THE DISK DRIVE WHILE CONNECTED

Whenever you are in SEND/RECEIVING or SCAN/RECEIVING operation, you can use AUTOTERM's commands for SAVING-TO-DISK, LOADING-FROM-DISK, MAINTAINING-A-SESSION-COPY, GETTING-A-DISK-DIRECTORY, KILLING-A-FILE, and TRANSMITTING-A-FILE. Merely type <SHIFT-CLEAR> and then type <S>, <L>, <M>, <G>, <D>, or <T>. These commands are described in section 4, p. 33.

When AUTOTERM is performing disk operations, it cannot receive characters sent by the other computer. You should try to use the disk commands only during periods when the other computer is waiting for a response from you. Then you will not miss any characters sent by the other computer.

3.6. SENDING A BLOCK OR FILE OF TEXT OR DATA

You may wish to send one or more blocks or files of text or data during your session with the other computer. Usually you will want to prepare the text ahead of time, before making connection with the other computer. AUTOTERM makes it very convenient to do this.

You can use the TEXT PROCESSING option to prepare your blocks of text or data. It is very easy to correct typing errors as-you-go, and to make other changes after proofreading. The data can be saved on disk and recalled while on line, or just before calling the other computer. Or the data can be held in your accumulated text while you return to the MAIN MENU and select the INTELLIGENT TERMINAL option.

3.6.1. TRANSMITTING THE BLOCK OR FILE

Let's assume that you are connnected to the other computer and that the data to be sent is contained in a disk file or in your accumulated text. You can now use the TRANSMIT BLOCK/FILE command to send the data to the other computer.

While in SEND/RECEIVING operation, type <SHIFT-CLEAR>. AUTOTERM will ask for the command letter. Type <T> for TRANSMIT. The top two lines of your screen will show:

```
SENDING A BLOCK OR FILE - TYPE
BLOCK # OR <C> OR <F>
```

If the data to be sent is contained in your accumulated text, then you should either type its block number, or, if thblock is not marked, type <C> and place the cursor at the START and STOP points. This process is the same as that ddearibed for DELETE BLOCK/FILE in section 2.7, p. 18.

If the data to be sent is contained in a disk file, then you should type <F>. In this case you will be prompted to type the filename. The procedure for entering a filename is the same as described in section 4.1.1, p. 33. The file must consist of text or a BASIC program in ASCII form. BINARY DATA must be loaded into the computer and transmitted as a block.

After your data has been sent to the other computer, you are still connected and can continue the session. You can send additional blocks or files at the appropriate points in the session.

You can use the <BREAK> to halt AUTOTERM's transmission of data. It may be necessary to hold down the <BREAK> key for a few seconds. Any other keys typed during BLOCK/FILE TRANSMISSION will be ignored by AUTOTERM.

If you are sending a block of binary data, then you should be sure to have the PARITY set to OFF and SEND-LINEFEED-AFTER-CARRIAGE-RETURN set to NO, i.e. PAR=OFF and SLF=N. See section 3.9, pp. 31-32. The other computer, which is receiving your binary data, should also have PARITY turned OFF. If the other computer is using AUTOTERM, then it should have PAR=OFF, IGD=N, IGL=N, & IGN=N, to avoid losing any of the

binary data. Furthermore, you should make use of the CHECKSUM command when sending &/or receiving binary data. See section 3.7, p. 29.

3.6.2. VISUAL CHECK OF TRANSMISSION SPEED

You can visually check that AUTOTERM is transmitting your data. The upper left corner of your screen changes background with each 100 characters transmitted.

If you are operating at 30 characters per second, then the upper left corner should change every 3 or 4 seconds. If it is changing every 10 seconds, then AUTOTERM is delaying for one-fifteenth second after each character is transmitted. AUTOTERM is waiting for the other computer to echo each character, but the other computer is not echoing. In such a situation, you should set DUP=HALF just before transmitting the block. Then change back to DUP=FULL immediately after transmitting the block.

3.6.3. AUTOMATIC LINE-AT-A-TIME TRANSMISSION

Many computer services use a "LINE-AT-A-TIME" proceedure for submission of messages, programs, etc. You would type a single line of your message and end this line with a CARRIAGE RETURN, i.e. <ENTER>. The computer service replies with one or more characters to indicate that it is ready for your next line. This process is repeated for each line of your message. Usually the computer service's reply ends with a specific character, which can be taken as a "prompt" for you to type your next line. You can use AUTOTERM for such LINE-AT-A-TIME transmission by suitably setting the XNF option.

As an example, suppose that the computer service supplies a line number followed by a COLON as an indication that it is ready for the next line of your message. You would like to transmit your entire message as a single block of text. You can acheive this by using the CARRIAGE RETURN (Hex OD) as the TRANSMISSION-OFF CHARACTER and the COLON (Hex 3A) as the TRANSMISSION-ON CHARACTER. That is, set XNF=3AOD. When you prepare your message off line, be sure to type <ENTER> at the end of each line of text. When you get on line, you merely issue the transmit block command. AUTOTERM will send your first line of text and halt after sending the CARRIAGE RETURN. When AUTOTERM receives a COLON from the other computer, then AUTOTERM will send your second line of text and again halt after sending the CARRIAGE RETURN. When AUTOTERM receives the next COLON, it will send the third line of your message, etc. This process will continue until the entire message has been transmitted.

When sending data by the LINE-AT-A-TIME proceedure, it is usually not necessary to set DUP=HALF, as described in section 3.6.2. The other computer will probably continue to echo each character, because it is assuming that you are sending the message manually.

3.7. CHECKSUMMING A BLOCK OF TEXT OR DATA

Errors can arise during the transmission of a block of text or data. If such transmission errors are noticed, then the block can be transmitted again, as often as necessary, until no errors occur. It is difficult to reliably make a visual check of text to catch incorrect characters. It is virtually impossible to visually check binary data for errors. The CHECKSUM proceedure offers a method of spotting transmission errors.

Each character or byte of data has an equivalent two-digit hexadecimal value. A CHECKSUM of a block of text or data, is obtained by the following proceedure:

1. At the start, the CHECKSUM is set equal to zero.
2. The CHECKSUM is rotated left one bit. Bit 7 is rotated to bit 0.
3. The character is added to the rotated CHECKSUM. If a CARRY is generated, then one is added to the result. This gives the new CHECKSUM.
4. Steps 2 and 3 are repeated for each character.

There are 256 possible values for the CHECKSUM of a block of text or data.

The sender can calculate the CHECKSUM for the block before it is transmitted. Then the receiver can calculate the CHECKSUM for the block which has been received. If no transmission errors occurred, then the sender's CHECKSUM should be the same as the receiver's CHECKSUM. If transmission errors occur, then it is very very unlikely that the two CHECKSUM values would agree.

When you are transmitting or receiving a block of text or data, you should use the CHECKSUM to check for transmission errors. This is especially important when working with a block of binary data.

You can use the CHECKSUM command to calculate the CHECKSUM of any block of text or data. This command can be used in any of the three operating modes. However, it is most frequently used during INTELLIGENT TERMINAL operation.

Type <SHIFT-CLEAR>. AUTOTERM will ask for the command letter. Type <C> for CHECKSUM A BLOCK. You will be prompted to type the block number or, if the block is not marked, to place the cursor at the START and STOP points. This process is the same as described for DELETE BLOCK in section 2.7, p. 18. The CHECKSUM for your block of data will be calculated. The top two lines of your screen will show:

```
BLOCK CHECKSUM VALUE IS "NN"
TYPE <BREAK> TO CONTINUE
```

NN is the two-digit hexadecimal value of the CHECKSUM. At this point, you should write down the CHECKSUM value and then press the <BREAK> key. You can then send the CHECKSUM value to the other computer for verification.

3.8. SENDING & RECEIVING PICTURE DATA

Pictures are displayed on the TV screen by means of Block Graphics Characters or by Extended Graphics Pixel Controls.

3.8.1. BLOCK GRAPHICS PICTURES

AUTOTERM conveniently displays Block Graphics Pictures. The Block Graphics Characters are handled the same as text. You can scroll through the picture and move the cursor through the picture. You can edit the picture by use of the Hex Code command described in section 2.7, p. 19. However, you should avoid inserting or deleting characters, because this would put the picture out of alignment. That is, two horizontal lines of the picture would fail to line up or match up properly. A picture composed of Block Graphics Characters should be preceded by a CARRIAGE RETURN. This insures that the first line (and each subsequent line) starts at the left side of your screen.

Block Graphics Characters can be sent and/or received in the same way that text is sent and/or received. When sending/receiving Block Graphics, you must set PARITY to OFF, i.e. PAR=OFF. See PAR, p. 31.

3.8.2. EXTENDED GRAPHICS PICTURES

AUTOTERM can send and receive Extended Graphics Pictures, but is unable to display these pictures. These pictures consist of binary data corresponding to special pixel controls. AUTOTERM's screen display of this binary data is a mixture of text and block graphics.

This binary data can be treated like a machine language program. The START ADDRESS corresponds to the memory location which contains the first element of data when the picture is held in memory. The TRANSFER ADDRESS can be set to the same value as the START ADDRESS. Actually, the TRANSFER ADDRESS has no meaning because a picture is not executed.

The binary data for an Extended Graphics Picture can be loaded from disk by AUTOTERM before it is sent, and it can be saved to disk after it is received. This is described in sections 4.1.6, p. 35, and 4.2.6, p. 38.

When sending/receiving Extended Graphics Pictures you must set PAR=OFF, IGD=N, IGL=N, IGN=N, & SLF=N.

3.8.3. AUTOMATIC BLOCK GRAPHICS MODE

AUTOTERM automatically adjusts to the receipt of block graphics. Whenever the three characters ESC G 4 are received, AUTOTERM switches to a mode where the PARITY bit (see PARITY, p. 31) is included as part of the character representation. This causes the graphics picture to be displayed on the screen during its receipt, as well as during later scrolling or recall from disk. AUTOTERM returns to the normal treatment when the three characters ESC G N are received, or whenever you cycle through the main menu.

3.9. USER OPTIONS

You can change certain aspects of AUTOTERM operation. This is done via the USER OPTION command. Section 2.8, p. 20. contains a description of the use of this command.

Here are the items which specifically affect INTELLIGENT TERMINAL operation. Each item is preceded by its abreviated three letter code. The codes are arranged alphabetically.

CBD COMMUNICATIONS BAUD RATE 110, 150, 300, 600, or 1200 Preset to 300

300 BAUD equals 30 characters per second, etc. If CBD=110, then the number of STOP BITS is usually set to two, i.e. STP=2. When CBD=110, AUTOTERM cannot send characters simultaneously with the receipt of characters. At 110 BAUD, you should not attempt to send characters to the other computer while it is sending characters to you.

DUP COMMUNICATIONS DUPLEX HALF or FULL Preset to FULL

FULL causes AUTOTERM to expect the other computer to return an echo of each character sent by AUTOTERM. This echo should occur within one-fifteenth second after AUTOTERM has sent the character. Otherwise, the sent character will not be saved in memory and will not be displayed on the screen.

HALF causes AUTOTERM to save and display each character sent to the other computer. AUTOTERM will not wait for an echo. If an echo is received, then the character will be put on the screen a second time. If double characters appear on your screen, then you should change the DUPLEX from HALF to FULL.

IGD IGNORE DEL Y or N Preset to Y

Some timeshare services use the ASCII DEL character (Hex 7F) as a DELAY character or FILLER character. In such cases, it is desirable for the character to be ignored by AUTOTERM. "Y" causes the DEL character to be ignored when received from the other computer. Use IGD=N when receiving binary data.

IGL IGNORE LINEFEED Y or N Preset to Y

Most timeshare sevices send a LINEFEED after each CARRIAGE RETURN. These LINEFEEDS result in extra blank lines if they are not ignored by AUTOTERM. "Y" causes the first LINEFEED after a CARRIAGE RETURN to be ignored. "Y" has no affect on other characters received. Use IGL=N when receiving binary data.

IGN IGNORE NULL Y or N Preset to Y

Some timeshare services use the ASCII NULL character (Hex 00) as a DELAY or FILLER character. It is desirable for the character to be ignored by AUTOTERM. "Y" causes the NULL character to be ignored when received from the other computer. Use IGN=N when receiving binary data.

PAR PARITY OFF, EVEN, ODD, MARK, or SPACE Preset to EVEN

EVEN is appropriate for most computer services. Some may require another setting, such as MARK. In particular, PAR=MARK

should be used with computer services which require seven data bits rather than eight. The PAR setting affects the bit pattern of only those characters being sent. It has little effect on received characters. The parity bit of received characters is ignored and set to zero when PAR is equal to EVEN, ODD, MARK, or SPACE. However, PAR=OFF causes the parity bit to be treated as part of the character representation. PAR=OFF is required when sending and/or receiving binary data or graphics data.

SCR SCROLLING WHILE RECEIVING Y or N Preset to Y

You may find that scrolling makes it difficult to read text as it is being received. Use "N" to inhibit this scrolling. When the bottom of the screen is reached, the text will continue at the top.

SLF SEND LINEFEED AFTER CARRIAGE RETURN DURING BLOCK TRANSMITION Y or N Preset to N

Some computer services require that you send a LINEFEED after each CARRIAGE RETURN during transmission of a block of data. Choose "Y" in such cases. Be sure to use SLF=N when sending binary data.

STP STOP BITS 1 thru 125 Preset to 1

"1" causes a single STOP BIT to be sent after the eight data bits. "2" causes two STOP BITS to be sent, etc. If CBD=110, then STP should be set to two or higher. "1" is appropriate for all other BAUD rates. Values greater than one (or two) do not hinder transmission, other than to slow it down because of the extra amount of delay between characters. Sometimes this delay is desirable because the other computer is unable to accept data at full speed. You can "retard" the transmission speed by 10% by adding an extra STOP BIT, STP=2, or by 50% by adding ten extra STOP BITS, STP=11. Some computer services require two STOP BITS and seven data bits. This pattern can be obtained from AUTOTERM by setting STP=1 and PAR=MARK.

XNF TRANSMISSION ON/OFF CHARACTERS 0000 thru FFFF Preset to 0000

This four digit hexadecimal number specifies "TRANSMISSION-ON" and "TRANSMISSION-OFF" characters used by AUTOTERM during the block transmission described in section 3.6. A value of "0000" has no affect on block transmission. For any other value, the first two digits specify the hexadecimal value for the TRANSMISSION-ON character, and the last two digits specify the hexadecimal value for the TRANSMISSION-OFF character. While transmitting a block of data, AUTOTERM halts transmission whenever it sends or receives the TRANSMISSION-OFF character. After halting, AUTOTERM does not resume transmission until the TRANSMISSION-ON character is received from the other computer. A setting of XNF=1113 corresponds to standard XON/XOFF transmission protocol. Other settings can be used to acheive other desired behavior, such as LINE-AT-A-TIME transmission. See section 3.6.3, p. 28.

4. USING DISK DRIVES

The AUTOTERM commands for SAVE BLOCK and LOAD BLOCK give you the ability to save all or part of your text to disk and to load text from disk. Either of these actions can be performed during TEXT PROCESSING, SEND/RECEIVING, SCAN/RECEIVING, or KEYSTROKE MULTIPLIER DEFN. However, you should be careful when performing disk operations during SEND/RECEIVING or SCAN/RECEIVING. AUTOTERM will not capture incomming characters from the other computer during disk operations.

4.1. SAVING TO DISK

4.1.1. SAVING TEXT

We will assume that you wish to create a file on disk that will contain all or part of your accumulated text. This section or block of text may or may not be marked with pairs of block markers, such as, :START1:, :STOP1:, :START2:, :STOP2:, etc. Section 2.7, p. 18, gives a detailed example of working with a block of text, unmarked or marked. You may find it helpful to review that section before proceeding.

Type <SHIFT-CLEAR>. The top of the screen will show:

```
TYPE COMMAND OR <?>    NNNNN M L
```

Now type the key <S> for SAVE BLOCK. The top two lines of the screen will show:

```
   SAVING A BLOCK OF TEXT - TYPE
ONE DIGIT BLOCK # OR <C>
```

If your section of text has been marked, then type the single key for its number. Otherwise, type <C> and you will be prompted to use the cursor to specify the START and STOP points for the block. In either case, after you have specified the block, the top two lines of your screen will show:

```
   SAVING A BLOCK - TYPE FILENAME
& PRESS <ENTER>
```

At this point you should type a filename of at most eight letters and/or digits. You may include an optional three-character extension preceded by a SLASH. If no extension is specified, then AUTOTERM will use ATM for the type, unless the data is a BASIC program in BINARY form or a machine language program. See sections 4.1.4 and 4.1.5. You may specify a disk drive number preceded by a COLON. If no drive number is specified, then AUTOTERM will use drive 0. You can use the <LEFT-ARROW> key to cancel previously typed characters.

When you have typed the filename correctly, then press the <ENTER> key. AUTOTERM will create the desired file and write the data to the file. The file will have been created on disk in the same form as a BASIC program in ASCII format. This

format is readable by many word processors. After the file has been created, you will be returned to your previous operating mode.

If your filename corresponds to a file that is already present on the disk, then your text will be appended to the existing file. If you type <ENTER> with no filename, then AUTOTERM will use the filename which you had last entered with a previous SAVE command.

If a disk I/O error occurs, then your screen will show:

```
SAVING A BLOCK - I/O ERROR
TO RECOVER PRESS <BREAK>
```

You should then type <BREAK> to return to your previous operating mode.

If there is not enough free space on the disk, then your screen will show:

```
SAVING A BLOCK - NO SPACE
TO RECOVER PRESS <BREAK>
```

You can then type <BREAK> to return to your previous operating mode.

4.1.2. SAVING KEYSTROKE MULTIPLIER DEFINITIONS

You can save your KEYSTROKE MULTIPLIER DEFINITIONS on disk. Merely select operating mode #3 when at the MAIN MENU. This will cause your KEYSTROKE MULTIPLIER DEFINITIONS to be displayed on the screen. You can then follow the procedure described in section 4.1.1 to save a block of text. However, since you are in operating mode #3, you will be saving all or part of your KEYSTROKE MULTIPLIER DEFINITIONS. If you do not specify an extension via SLASH, then KSM will be used.

4.1.3. SAVING A BASIC PROGRAM IN ASCII FORMAT

AUTOTERM can be used to receive BASIC programs which are sent in ASCII form from another computer. AUTOTERM can also be used to load BASIC programs which have been saved on disk in ASCII form by the SAVE "----",A command. See section 4.2.3, p. 37. In either case, the BASIC program is contained in your accumulated text. It can be edited as described in sections 2.6 and 2.7, pp. 15-19.

Such a BASIC program can be saved to diskette as described in section 4.1.1. Disk files created in this manner can be read by the LOAD "----" command.

4.1.4. SAVING A BASIC PROGRAM IN BINARY FORMAT

AUTOTERM can be used to receive BASIC programs which are sent in BINARY form from another computer. AUTOTERM can also be used to load BASIC programs which have been saved on disk in BINARY form by the SAVE "----" command. In either case, the

program is contained in your accumulated text as BINARY data. It appears on the screen as a mixture of text and graphics characters.

Such a BASIC program can be saved to disk as described in section 4.1.1, provided it is preceeded by the four characters which AUTOTERM creates when loading such files. These four characters are; a graphics hexadecimal "FF", two "B"'s, and a CARRIAGE RETURN. These four characters must be the first four characters of the block. They enable AUTOTERM to recognize that you want the block to be saved on diskette in the format appropriate for a BASIC program in BINARY form. AUTOTERM will use BIN for the extension if none is specified via the SLASH. Disk files created in this manner can be loaded by the LOAD "----" command.

If a BASIC program in BINARY form has been received from another computer, then you can easily add the above mentioned four characters. Merely type the <SHIFT-RIGHT-ARROW> four times to insert four SPACES ahead of the data. Then use the "H" command (see section 2.8, p. 20) to change the first SPACE to a graphics hexadecimal "FF". That is, type <SHIFT-CLEAR> <H> <F> <F> <ENTER>. Then type <B> <B> to change the second and third SPACES to "B"'s, and then type <ENTER> to change the fourth SPACE to a CARRIAGE RETURN.

4.1.5. SAVING A MACHINE LANGUAGE PROGRAM

(START/TRANSFER ADDRESSES no longer required. See p. 6.1.)

AUTOTERM can be used to receive MACHINE LANGUAGE programs from another computer. AUTOTERM can also be used to load MACHINE LANGUAGE programs which have been saved on diskette by the SAVEM command. See section 4.2.5, p. 38. In either case, the MACHINE LANGUAGE program is contained in your accumulated text as BINARY data. It appears on the screen as a mixture of text and graphics characters.

Such a MACHINE LANGUAGE program can be saved to diskette as described in section 4.1.1 provided it is preceded by the eighteen characters of information which AUTOTERM creates when loading such files. The eighteen characters consist of three parts:

1) A graphics hexadecimal "FF", an "M", an "L", and a CARRIAGE RETURN.
2) An "S", an "=", four characters representing the START ADDRESS in hexadecimal, and a CARRIAGE RETURN.
3) A "T", an "=", four characters representing the TRANSFER ADDRESS in hexadecimal, and a CARRIAGE RETURN.

The first four characters enable AUTOTERM to recognize that you want the block to be saved as a MACHINE LANGUAGE program. The next fourteen characters enable AUTOTERM to store the START and TRANSFER ADDRESSES in the header block of the disk file. AUTOTERM will use BIN for the extension if none is specified via the SLASH. Disk files created in this manner can be loaded by the LOADM command.

If the MACHINE LANGUAGE program has been received from another computer, then you can easily add the above mentioned eighteen characters. Merely use the <SHIFT-RIGHT-ARROW> to

insert eighteen SPACES. Then change the SPACES to the desired characters by typing over the SPACES. The H command (see section 2.8, p. 20) is used to enter the graphics hexadecimal "FF". You must type <SHIFT-CLEAR> <H> <F> <F> <ENTER>.

4.1.6. SAVING PICTURE DATA

PICTURE DATA consisting of Block Graphics Characters (see section 3.8.1, p. 30) can be saved to cassette the same as text, as described in section 4.1.1, p. 33.

PICTURE DATA consisting of Extended Graphics Data (see section 3.8.2) can be treated the same as a MACHINE LANGUAGE program. The START ADDRESS is the memory location which contains the first element of data when the PICTURE is held in memory. The TRANSFER ADDRESS has no meaning because a PICTURE is not executed. A PICTURE is stored in memory, and then observed by switching to the proper screen graphics mode.

AUTOTERM can be used to receive the BINARY data for an Extended Graphics PICTURE from another computer. AUTOTERM can also be used to load the BINARY data for an Extended Graphics PICTURE from a disk file which was created by using SAVEM to save the data as a MACHINE LANGUAGE program. In either case, the Extended Graphics Data is contained in your accumulated text. It appears on the screen as a mixture of text characters and graphics characters. AUTOTERM can save the data on a disk file as if it is a MACHINE LANGUAGE program. See section 4.1.5. The resulting disk file can be loaded by the LOADM command.

4.2. LOADING FROM DISK

4.2.1. LOADING TEXT

We will assume that you have a file of text on diskette and you wish to APPEND this text to your accumulated text. (If you wish to REPLACE your accumulated text, then you should first delete your accumulated text by using the DELETE BLOCK command.)

Type <SHIFT-CLEAR>. The top of the screen will show:

```
TYPE COMMAND OR <?>    NNNNN M L
```

Now type the key <L> for LOAD BLOCK. The top two lines of the screen will show:

```
LOADING FROM DISK- TYPE FILENAME
& PRESS <ENTER>
```

At this point you should type the filename of the file you wish to load from disk. You can use the <LEFT-ARROW> key to cancel previously typed characters. After you have typed the filename correctly, then press the <ENTER> key. AUTOTERM will read the file. The text will be appended to your accumulated text, and you will be returned to your previous operating mode. If you do not specify an extension, then ATM will be assumed.

If there is not enough space in memory to hold the entire

file, then AUTOTERM will load as much as possible. It will then display on your screen:

```
LOADING FROM DISK- NO MORE SPACE
TO RECOVER PRESS <BREAK>
```

You can then type <BREAK> to return to your previous operating mode. You can load the remainder of the file. Merely delete some of the accumulated text and then execute the LOAD command again, using <ENTER> without a filename. AUTOTERM will load more text from the file, starting at the last stopping point.

If a disk I/O error occurs while the file is being read, then your screen will show:

```
LOADING FROM DISK- I/O ERROR
TO RECOVER PRESS <BREAK>
```

You should type <BREAK> to return to your previous operating mode.

4.2.2. LOADING KEYSTROKE MULTIPLIER DEFINITIONS

You may have created one or more disk files of KEYSTROKE MULTIPLIER DEFINITIONS as described in section 4.1.2. You can easily load these files into AUTOTERM'S KEYSTROKE MULTIPLIER DEFINITIONS. Merely select operating mode #3 at the MAIN MENU. This puts AUTOTERM into the KEYSTROKE MULTIPLIER DEFINITION operating mode. Next follow the procedure described in section 4.2.1. The KEYSTORKE MULTIPLIER DEFINITIONS which are on the disk file will be appended to the KEYSTROKE MULTIPLIER DEFINITIONS which are currently in memory. If you do not specify an extension via the SLASH, then KSM will be assumed.

4.2.3. LOADING A BASIC PROGRAM IN ASCII FORMAT

AUTOTERM can be used to edit a BASIC program which has been saved in ASCII format by the SAVE "----",A command. The program can be loaded into the accumulated text by the procedure described in section 4.2.1. The program can be edited and then saved on diskette as described in section 4.1.3, p. 34. The program can also be sent to another computer as a block of text by use of the TRANSMIT BLOCK command, section 3.6, p. 27.

4.2.4. LOADING A BASIC PROGRAM IN BINARY FORMAT

AUTOTERM can be used to load a BASIC program which has been saved by the SAVE "----" command. Merely follow the procedure described in section 4.2.1. Be sure to specify the file extension. The BASIC program will be appended to your accumulated text. The BASIC program will appear on your screen as a mixture of text and graphics characters because it consists of BINARY data. The program will be preceded by:

```
□BB
```

The ↑ is a graphics "FF". This character tells you that the subsequent text material is BINARY data. The "BB" tells you that the subsequent BINARY data is a BASIC program in BINARY form. The characters "□BB" are followed by a CARRIAGE RETURN to make them more visually apparent.

You cannot conveniently edit a BASIC program when it is in BINARY form. However, you can send it to another computer by use of the TRANSMIT BLOCK command as described in section 3.6, p. 27. This may be more appealing than transmitting the program in ASCII form becuase the BINARY form is shorter.

4.2.5. LOADING A MACHINE LANGUAGE PROGRAM

The procedure described in section 4.2.1 can be used to load a MACHINE LANGUAGE program which has been saved on disk by the SAVEM command. The file extension must be specified. The program will be appended to your accumulated text. It will appear on your screen as a mixture of text and graphics characters because it is BINARY data. It will be preceded by:

```
□ML
S=NNNN
T=MMMM
```

The □ is a graphics "FF". It tells you that the subsequent text is BINARY data. The "ML" tells you that the subsequent BINARY data is a MACHINE LANGUAGE program. The "NNNN" is the START ADDRESS in hexadecimal. The "MMMM" is the TRANSFER ADDRESS in hexadecimal.

You can edit the START and TRANSFER ADDRESSES of a MACHINE LANGUAGE program. You should be sure that the new addresses are consistent with proper operation of the program. You can send a MACHINE LANGUAGE program to another computer by the TRANSMIT BLOCK command, section 3.6, p. 27.

4.2.6. LOADING PICTURE DATA

AUTOTERM can be used to load PICTURE DATA from disk. If the PICTURE consists of Block Graphics Characters (see section 3.8.1, p. 30) then it can be loaded as text, provided it had been saved as text as described in section 4.2.1.

If the PICTURE consists of Extended Graphics Data (see section 3.8.2, p. 30), then the PICTURE should have been saved by the SAVEM command as if it were a MACHING LANGUAGE program as described in section 4.1.6, p. 36. This data can be loaded from disk in the manner described in section 4.2.5. The START ADDRESS represents the memory location where the start of the PICTURE is stored in memory when loaded by the LOADM command. The TRANSFER ADDRESS is present, but has no meaning.

You do not actually see the PICTURE when it is appended to your accumulated text. You merely see the BINARY data. You cannot conveniently edit this data. However, you can send it to another computer by using the TRANSMIT BLOCK command. See section 3.8.2., p.30.

4.3. CONTINUOUS DISK SAVES OF SESSION DATA

AUTOTERM offers a convenient method of continuously maintaining a disk file of your session. The "MAINTAIN" command lets you easily copy to disk only the not-yet-saved portion of your current session.

The MAINTAIN command does the following:

1. Previously saved data is deleted from your computer's memory.
2. Newly accumulated data is appended to the disk file and retained in your computer's memory.
3. A :SAVED: marker is added to your text.

Figure 4.1 illustrates the effect of the first and second executions of MAINTAIN. Subsequent executions have an effect similar to the second execution.

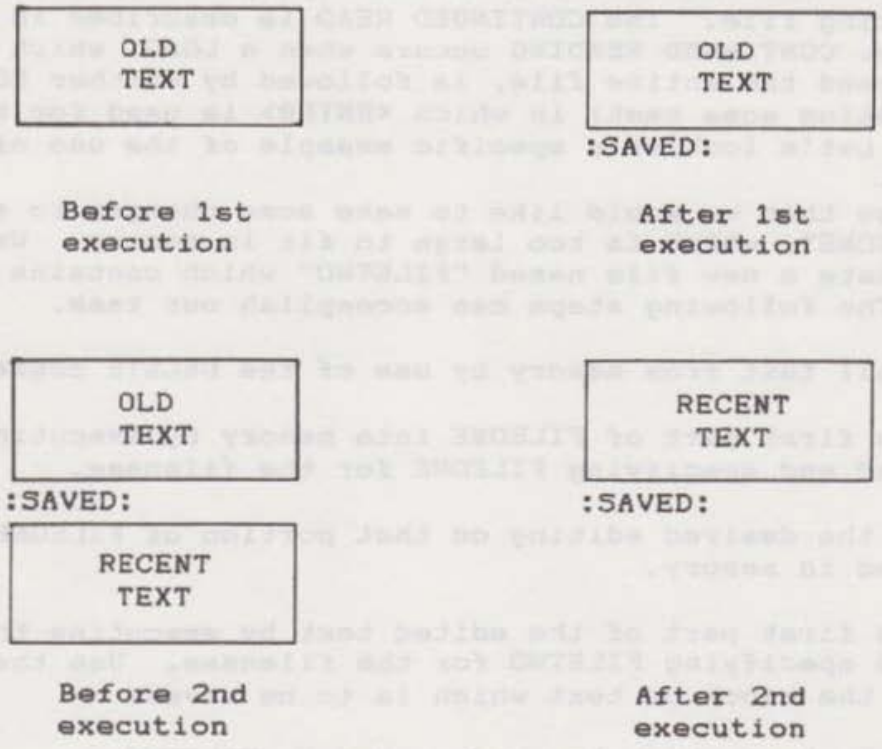

Figure 4.1
THE "MAINTAIN" COMMAND

To execute the MAINTAIN command, type <SHIFT-CLEAR>, and then type <M>. The top two lines of your screen will show:

```
MAINTAINING TEXT - TYPE FILENAME
& PRESS <ENTER>
```

At this point you can specify a filename or merely type <ENTER>

to specify "same filename as last time". AUTOTERM will search for the :SAVED: marker. If the marker is found, then all data before the marker will be deleted, all data after the marker will be appended to the disk file, and the marker will be placed at the end of the data. If the marker is not found, then all the data will be appended to the disk file, and the marker will be placed at the end of the data.

If an I/O ERROR occurs or there is not enough room on the disk, then the data in memory is not altered. That is, no data is deleted and the marker is not moved (or created). Such situations are handled as described in section 4.1.1 above.

4.4. FILES LARGER THAN MEMORY

AUTOTERM'S ability to do an APPEND to a file and to do a CONTINUED READ of a file make it possible to handle files which cannot fit into memory. The APPEND is described in section 4.1.1 above. APPENDING occurs automatically whenever you SAVE to an existing file. The CONTINUED READ is described in section 4.2.1 above. CONTINUED READING occurs when a LOAD, which was unable to read the entire file, is followed by another LOAD (after deleting some text) in which <ENTER> is used for the filename. Let's look at a specific example of the use of these operations.

Suppose that we would like to make some changes to a file named "FILEONE", which is too large to fit in memory. We would like to create a new file named "FILETWO" which contains the changes. The following steps can accomplish our task.

1. Delete all text from memory by use of the DELETE command.

2. Load the first part of FILEONE into memory by executing the LOAD command and specifying FILEONE for the filename.

3. Perform the desired editing on that portion of FILEONE which is contained in memory.

4. Save the first part of the edited text by executing the SAVE command and specifying FILETWO for the filename. Use the cursor to specify the block of text which is to be saved.

5. Delete the text which has been saved to FILETWO.

6. Load the next portion of FILEONE by executing the LOAD command, and merely typing <ENTER> when asked to type the filename. This will cause a new portion of FILEONE to be loaded into memory. The new portion will continue at the point where the previous load stopped.

7. Perform desired editing of the newly loaded text.

8. Save the next part of edited text to FILETWO by executing the SAVE command and typing <ENTER> when asked for the filename.

9. Delete the part of text which was saved to FILETWO.

Repeat steps 6 thru 9 until the entire file has been loaded, edited, and saved.

Judicious use of APPENDING and CONTINUED READING can achieve extensive manipulation of files which are larger than memory. You can pull a section from one file and insert it into another file, creating a new file in the process. You can merge selected sections from several files. Such operations may require repeated use of procedures such as that outline above.

4.5. GETTING THE DIRECTORY OF A DISK

The GET-A-DIRECTORY command enables you to see the granule contents of a disk. Type <SHIFT-CLEAR>, and then type <G>. The top two lines of your screen will display:

```
GETTING A DIRECTORY - TYPE
DRIVE # □
```

At this point you should type the single digit for the drive number, i.e. <0>, <1>, <2>, or <3>. Your screen will show the contents of the diskette in the format shown below.

```
DRIVE 0 - FREE 41
PRESS <BREAK> TO RECOVER
FILEONE/ATM  17  FILETWO/BIN  12
FILETHRE/ATM 22  FILEFOUR/KSM 11
FILEFIVE/BIN  9  FILESIX/BIN   7
FILESEVN/ATM  3
```

Of course, the filenames and extensions will be the actual filenames on the disk. Notice that the free space appears on the top line.

If there are more files than can be listed on the screen, then the 2nd line will read

```
PRESS <ENTER> FOR NEXT PAGE
```

You can type <ENTER> to see a continuation of the list, or you can type <BREAK> to recover.

4.6. KILLING A FILE

A file can be deleted from a disk by use of the DELETE command. Type <SHIFT-CLEAR>, and then type <D>. Your screen will show:

```
DELETING A BLOCK OR FILE - TYPE
BLOCK # OR <C> OR <F>
```

At this point you should type <F> to indicate that you wish to

specify a filename. Your screen will then show:

```
DELETING A FILE - TYPE FILENAME
& PRESS <ENTER>
```

You can now type the name of the file which you would like to delete from the disk.

4.7. COPYING A FILE FROM CASSETTE TO DISK

Your AUTOTERM disk contains a program which can be used to copy files created by the CASSETTE VERSION OF AUTOTERM onto disk, so that they can be loaded by the DISK VERSION OF AUTOTERM.

While in DISK BASIC type LOADM"TAPEDISK". After the program has been loaded, type EXEC to execute it. Your screen will show:

```
LOADING A FILE FROM CASSETTE
TYPE FILENAME AND <ENTER>
```

You can type the filename, or merely type <ENTER> to indicate that the next file on cassette is to be loaded. Next the screen will show:

```
PUT UNIT INTO PLAY AND
PRESS <ENTER>
```

Put your tape unit into "PLAY" and then press <ENTER>. The file will be read into memory. Next your screen will display:will display:

```
READY TO SAVE TO DISK
TYPE FILENAME & <ENTER>
```

After you have typed the filename and <ENTER>, the data will be saved on disk. If the filename already exists, the data will be appended to the file.

5. USING THE PRINTER

The AUTOTERM PRINT command makes it very convenient to satisfactorily print all or part of your accumulated text or KEYSTROKE MULTIPLIER DEFINITIONS. You are free to choose page size, margins, and line spacing. WORD WRAP is applied to avoid splitting words at the ends of lines. There are additional options, such as the ability to pause at the end of each page, PGP, or to do continuous printing in the midst of a page, CNP.

Printing can be done during TEXT PROCESSING operation and during KEYSTROKE MULTIPLIER DEFN operation. You cannot use the printer during SEND/RECEIVING or SCAN/RECEIVING operation, because the serial I/O port is being used to connect to the other computer.

5.1. PRINTING A BLOCK OF TEXT

We will assume that you wish to print all or part of your accumulated text. This section of text may or may not be marked with a pair of block markers, such as, :START1:, :STOP1:, :START2:, :STOP2:, etc. Section 2.6, p. 17, gives a detailed example of working with a block of text, unmarked or marked. You may find it helpful to review that section before proceeding.

Type <SHIFT-CLEAR>. The top of the screen will show:

```
TYPE COMMAND OR <?>    NNNNN M L
```

Now type the key <P> for PRINT BLOCK. The top two lines of the screen will show:

```
PRINTING A BLOCK OF TEXT - TYPE
SINGLE DIGIT BLOCK # OR <C>
```

If your section of text has been marked, then type the single key for its number. Otherwise, type <C> and you will be prompted to use the cursor to specify the START and STOP points for the block.

In either case, after you have specified the block, AUTOTERM will start printing at the START point. Printing will continue until the STOP point is reached.

At any time you can press the <BREAK> key to cancel the PRINT BLOCK command. However, if your printer is not turned-on, then AUTOTERM will become inaccessible while waiting for the printer. If you cannot turn-on the printer, then you can press the RESET BUTTON on the back of your computer. You will be returned to the MAIN MENU, and your accumulated text will be preserved.

5.2. EMBEDDING PRINTER CONTROL CODES

Your printer may have extra "intelligent" abilities, such as printing boldface. Usually, you tell the printer to perform such actions by sending "printer control codes" to the printer.

When these printer control codes are included in the text, they upset the chaaracter count used for margins and word wrap. This can be avoideed by use of Hex CC, provided PWW=Y. The following does not apply when PWW=N.

AUTOTERM treats the graphics block character Hex CC as a marker for the inclusion of printer control codes. You can enclose any set of characters between a pair of Hex CC's. The Hex CC's will not be sent to the printer. The enclosed characters will be sent to the printer but will not be counted in calculations of margins and word wrap.

For example, suppose that your printer changes to boldface when it receives the two characters ESCAPE and G. You could send these two characters to the printer without upsetting margins and word wrap, by enclosing then between Hex CC's. This would appear on your TV screen as:

■;G■

The ■'s are the graphics block characters for Hex CC. Each is created by typing <SHIFT-CLEAR> <H> <C> <C> <ENTER>. See section 2.7, p. 19. The ; is the screen representation of ESCAPE. It is obtained by typing <CLEAR-BREAK>. See table 8.1, p. 76.

5.3. OVERRIDING THE NARROW WIDTH OF RECEIVED TEXT

Most computer services format their text for a particular sized display screen. The screen width rarely matches the width you'd like to use on the printed page. Output which is formatted for a screen width of 32 or 40 characters, has an especially unappealing appearance on standard sized paper. The text width is much too narrow.

Usually, the computer service uses extra CARRIAGE RETURNS to force the text to fit on the screen. AUTOTERM can suppress these "artificial" CARRIAGE RETURNS while the text is being sent to your printer. Merely, set OMIT CARRIAGE RETURNS to YES, i.e. OCR=Y. See section 5.4.

You might want to do some test printouts to see the results of OCR=Y. Be sure to have PWW=Y, as well. Otherwise, some words will be broken by the right hand margin.

The OCR=Y causes AUTOTERM to suppress all CARRIAGE RETURNS except those that are preceded or followed by a COLON, SPACE, CARRIAGE RETURN, LINEFEED, or FORMFEED. This rule avoids suppression of those CARRIAGE RETURNS which are likely to be desirable. When a CARRIAGE RETURN is supressed, it is replaced by a SPACE. However, an extra SPACE is inserted when the CARRIAGE RETURN is preceded by a PERIOD.

You may find that a few CARRIAGE RETURNS are being supressed when you'd like to keep them. You can do a small amount of editing to cause those few CAARRIAGE RETURNS to be kept. For example, merely insert a SPACE ahead of a CARRIAGE RETURN to render it non-artificial.

5.4. USER OPTIONS

You can change certain aspects of AUTOTERM operation. This is done via the USER OPTION command. Section 2.8, p. 20, contains a description of the use of this command.

Here are the items which specifically affect the PRINT BLOCK action. Each item is preceded by its abreviated three letter code. The codes are arranged alphabetically.

BMG BOTTOM MARGIN (lines) 0 to 125 Preset to 6

CNP CONTINUOUS PRINTING Y or N Preset to N

Normally, a new page is started at the beginning of printing. A "Y" causes printing to act as a continuation of the previous printout. A new page is not started. Printing begins at the next line on the current page.

LMG LEFT MARGIN (spaces) 0 to 125 Preset to 6

LSP LINE SPACING (lines) 1 to 125 Preset to 1

"1" causes SINGLE SPACING. "2" causes DOUBLE SPACING, etc.

OCR OMIT CARRIAGE RETURNS Y or N Preset to N

"Y" causes the suppression of "artificial" CARRIAGE RETURNS, that is, those not preceded or followed by a COLON, SPACE, CARRIAGE RETURN, LINEFEED, or FORMFEED.

PGL PAGE LENGTH (lines) 0 to 255 Preset to 66

"0" causes paging to be ignored.

PGP PAGE PAUSE Y or N Preset to N

"Y" causes printing to stop at the end of each page. The next page will not be started until you press <ENTER>.

PGW PAGE WIDTH (characters) 1 to 255 Preset to 80

PLF PRINT LINEFEED AFTER CARRIAGE RETURN Y or N Preset to N

"Y" causes a LINEFEED character to be sent to the printer after each CARRIAGE RETURN. Most printers automatically perform a LINEFEED after receiving a CARRIAGE RETURN. You should use "N" with these printers. Otherwise, you will have an extra space between lines.

PWW PRINTER WORD WRAP Y or N Preset to Y

"Y" causes AUTOTERM to avoid spliting a word between two lines during printing. If the entire word does not fit at the end of a line, then the word is moved to the start of the next line.

RMG RIGHT MARGIN (spaces) 0 to 125 Preset to 6

TMG TOP MARGIN (lines) 0 to 125 Preset to 6

6. BUILDING KEYSTROKE MULTIPLIERS

KEYSTROKE MULTIPLIER DEFINITIONS can be created and edited by selecting operating mode #3 at the MAIN MENU. These DEFINITIONS can also be printed, saved on disk, and loaded from disk. Merely use the PRINT, SAVE, or LOAD command while in operating mode #3.

For convenience, the abreviation KSM will be used to represent the term KEYSTROKE MULTIPLIER.

6.1. A SIMPLE EXAMPLE

Load AUTOTERM and enter the TEXT PROCESSING mode of operation, as described in section 2.1, p. 10. Use <SHIFT-ZERO> to obtain UPPER-CASE-ONLY on the keyboard. This will simplify our presentation of examples. Now type the following text:

THIS IS A DEMONSTRATION OF A KEYSTROKE MULTIPLIER.

Type <ENTER> at the end of the line, that is, after the PERIOD.

EXAMPLE 6.1. TYPING YOUR NAME

We will create a KSM DEFINITION for the key <1>. Type <BREAK> to return to the MAIN MENU. Then type <3>. Your screen's top line shows:

```
KEYSTROKE MULTIPLIERS  NNNNN M L
```

Now type the following:

```
FIRST DEMO
1:==:
JOHN JONES
:ND:
```

Type <ENTER> at the end of each line. You may want to type your own name in place of "JOHN JONES". The very first line is REMARK. It is not actually part of your KSM DEFINITION. The KSM DEFINITION starts with 1:==: and ends with :ND:. You can place REMARKS on separate lines between your KSM DEFINITIONS.

Next press <BREAK> to return to the MAIN MENU, and then type <1>. The screen now shows that you are back in the TEXT PROCESSING mode of operation.

Now type <SHIFT-CLEAR>, and then type <1>. AUTOTERM will act as though you had typed <J> <O> <H> <N> < > <J> <O> <N> <E> <S>. The screen will show:

```
TEXT PROCESSING        NNNNN M L
THIS IS A DEMONSTRATION OF A
KEYSTROKE MULTIPLIER.
JOHN JONES
```

Type <SHIFT-CLEAR> and then <1>. The screen will now show:

```
TEXT PROCESSING          NNNNN M L
THIS IS A DEMONSTRATION OF A
KEYSTROKE MULTIPLIER.
JOHN JONESJOHN JONES
```

6.2. USING THE <ENTER> KEY

We will assign to the keys <2> and <3>, KSM DEFINITIONS which are similar to that for <1>. Type <BREAK> and then <3>. Now type the following:

```
SECOND DEMO
2:==:
JOHN JONES<ENT>
:ND:
```

Press <ENTER> at the end of each line. Next type the following:

```
THIRD DEMO
3:==:
JO
HN JONES
<ENT>
:ND:
```

Be sure to type an <ENTER> at the end of each line, that is, after the "O", the ":", the "O", the "S", the ">", and the ":". Now type <BREAK> and then <1> to return to the TEXT PROCESSING mode. You have created KSM DEFINITIONS for the keys <2> and <3>. Let's see what they do! The cursor should be at the end of the line which reads:

```
JOHN JONESJOHN JONES
```

Now type <ENTER>. The cursor should be at the start of the next line, which is blank. Type <SHIFT-CLEAR> and then type <2>. AUTOTERM will act as though you had typed <J> <O> <H> <N> <SPACE> <J> <O> <N> <E> <S> <ENTER>. The screen now shows:

```
TEXT PROCESSING          NNNNN M L
THIS IS A DEMONSTRATION OF A
KEYSTROKE MULTIPLIER.
JOHN JONESJOHN JONES
JOHN JONES
```

The cursor is located at the start of a new line, which is blank. The five characters "<ENT>", at the end of the KSM DEFINITION for the key <2>, caused AUTOTERM to type the <ENTER> key.

Now type <SHIFT-CLEAR> and then type <3>. The screen now shows:

```
TEXT PROCESSING        NNNNN M L
THIS IS A DEMONSTRATION OF A
KEYSTROKE MULTIPLIER.
JOHN JONESJOHN JONES
JOHN JONES
JOHN JONES
```

The cursor is located at the start of the next line, which is blank. The KSM DEFINITION for the key <3> gives the same results as that for the key <2>. The use of the <ENTER> key during the creation of the KSM DEFINITION for the key <3> had no effect on the results.

Generally, you can use the <ENTER> key to make your KSM DEFINITIONS more readable. The <ENTER> key causes a CARRIAGE RETURN to be placed in the text of the KSM DEFINITION, but this CARRIAGE RETURN is ignored by AUTOTERM during the execution of the KSM. If you want to include <ENTER> as part of your KSM SEQUENCE, then represent it by the five characters "<ENT>".

EXAMPLE 6.2. SINCERELY YOURS, ETC.

Type <BREAK> and then <3> to get to KEYSTROKE MULTIPLIER DEFN mode. Now type the following:

```
******SINCERELY
4:==:
SINCERELY YOURS,<ENT>
<ENT>
<ENT>
<ENT>
JOHN JONES<ENT>
:ND:
```

Use the <ENTER> at the end of each line. This KSM DEFINITION enables the key <4> to generate this phrase commonly used at the end of a letters. You may want to add some spaces ahead of the "SINCERELY" and the "JOHN", to cause the phrase to be placed away from the left margin of the page.

6.3. MOVING THE CURSOR

The previous examples illustrate the use of <ENT>. This will be referred to as the WEDGE NOTATION for the <ENTER> key. Table 8.3, p. 81, shows the WEDGE NOTATION for other keys. Your KSM DEFINITIONS can include use of the arrow keys. Merely represent them as indicated in table 8.3.

EXAMPLE 6.3. A FOURTEEN-LINE SCROLL

This simple KSM may make it a little more pleasant to read through your text. The <Z> key is used as a GET-NEXT-PAGE type of command. When the command is issued, the text is scrolled

forward until the previous bottom line is moved to the top of the screen. When reading the text, merely type <SHIFT-CLEAR> and <Z>, whenever you reach the bottom of the screen. Then continue reading near the top of the screen. The KSM simply involves typing the <DOWN-ARROW> key fourteen times.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******SCROLLING                     Remark.
Z:==:
<DAR><DAR><DAR><DAR><DAR>           Do five <DOWN-ARROW>'s.
<DAR><DAR><DAR><DAR><DAR>           Do five <DOWN-ARROW>'s.
<DAR><DAR><DAR><DAR>                Do four <DOWN-ARROW>'s.
:ND:
```

Notice that explanatory comments have been placed on the right. This format will be used in the remaining examples.

6.4. LOOPING VIA :GOTO--:

A KSM may include branching via LABELS and GOTOS.

A LABEL consists of a pair of letters placed between COLONS and situated on a line by themselves, such as, :AA:. LABELS are ignored and passed over during execution of the KSM. They have effect only when referred to by a GOTO command.

A GOTO consists of GOTO followed by two letters with all six letters placed between COLONS and situated on a line by themselves, such as :GOTOAA:. A GOTO causes the KSM to continue execution at the designated LABEL point.

Branching typically occurs during COMPUTER DIALOGUES. You will see LABELS and GOTOS in several of the examples of section 7. A KSM which is not a COMPUTER DIALOGUE can involve branching, if it triggers repetitive action. Such action requires some form of looping. The next example shows the use of a LABEL and a GOTO to acheive repetitive action by looping.

EXAMPLE 6.4. THE BOUNCING CURSOR

We can modify Example 6.3 to cause the cursor to bounce repeatedly after the fourteen-line scroll. We merely add a LABEL, a few UP-ARROWS, a few DOWN-ARROWS, and a GOTO.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

*****BOUNCING                       Remark.
Z:==:
<DAR><DAR><DAR><DAR><DAR>           Do five <DOWN-ARROW>'s.
<DAR><DAR><DAR><DAR><DAR>           Do five <DOWN-ARROW>'s.
<DAR><DAR><DAR><DAR>                Do four <DOWN-ARROW>'s.
:AA:                                Our label.
<UAR><UAR><UAR><UAR><UAR>           Do five <UP-ARROW>'s.
<DAR><DAR><DAR><DAR><DAR>           Do five <DOWN-ARROW>'s.
:GOTOAA:                            Go repeat the bounce.
:ND:
```

When this KSM is executed, the cursor moves down fourteen lines. Then the cursor moves up five lines, down five lines, up five lines, etc. The up and down motion is caused by the loop from :AA: to the :GOTOAA:. This loop repeats until the action is interrupted by you pressing the <BREAK>.

6.5. USING TIME DELAYS

A KSM DEFINITION can include the specification of TIME DELAYS. Table 8.3, p. 81, shows how the WEDGE NOTATION can be used to represent a TIME DELAY. Let's modify example 6.3, p. 48, so that it repeats itself every few seconds.

EXAMPLE 6.5. A FOURTEEN-LINE SCROLL EVERY 6.5 SECONDS

Let's start with Example 6.3, p. 48. We will add a TIME DELAY after the fourteenth <DOWN-ARROW> and follow this with a :GOTOAA:. :AA: will be placed at the start of the KSM DEFINITION. This will cause the entire sequence to be repeated.

KEYSTROKE MULTIPLIER DEFINITION	EXPLANATION
******SCROLL & DELAY	Remark.
Z:==:	
:AA:	
<DAR><DAR><DAR><DAR><DAR>	Do five <DOWN-ARROW>'s.
<DAR><DAR><DAR><DAR><DAR>	Do five <DOWN-ARROW>'s.
<DAR><DAR><DAR><DAR>	Do four <DOWN-ARROW>'s.
<06S><.5S>	Delay 6.5 seconds.
:GOTOAA:	Do it all again!
:ND:	

Notice that the six and one-half second delay is acheived by performing two delays, one for six seconds and the other for one-half second. Of course, the specific delay period should be matched to your own reading speed.

6.6. CALLING SYSTEM COMMANDS & KEYSTROKE MULTIPLIERS

A KSM DEFINITION can include the use of SYSTEM COMMANDS and/or other KSMS. The <SHIFT-CLEAR> key is specified as <SCL>. See table 8.3, p. 81. The following few examples demonstrate this. Example 6.6 uses the command for CHANGING USER OPTIONS. Example 6.7 uses the PRINT BLOCK command. Example 6.8 uses the FIND command. Example 6.9 causes <SHIFT-CLEAR> <z> to act like <SHIFT-CLEAR> <Z>. Example 6.10 displays a MENU and has AUTOTERM await the user's command.

EXAMPLE 6.6. SETTING COMMUNICATIONS PARAMETERS

If you access several different time-share services, you may find it necessary to reset one or more of AUTOTERM's USER OPTIONS when you switch to a different service. These "re-settings" could be acheived by a single keystroke. The

following KSM DEFINITION for the key <1> causes the BAUD RATE to be set to 300, the DUPLEX to be set to HALF, and the PARITY to be set to MARK.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******SET COMM PARAMS                Remark.
1:==:
<SCL>U                              Execute U command.
CBD<ENT>300<ENT>                    Set CBD to 300.
<SCL>U                              Execute U command.
DUP<ENT>HALF<ENT>                   Set DUP to HALF.
<SCL>U                              Execute U command.
PAR<ENT>MARK<ENT>                   Set PAR to MARK.
:ND:
```

EXAMPLE 6.7. UNATTENDED PRINTING OF PORTIONS OF TEXT

Suppose that you wish to print selected portions of a large amount of text which resulted from an extended time-share session. Normally, if you are printing separate portions of text, you would have to wait for each section to be completely printed before giving the command to print the next session. A more appealing approach would be to mark all the START and STOP points before doing any printing, and then execute a single command which tells the computer to print the selected portions. This would allow you to attend to other activities while the printing is taking place. The following KSM DEFINITION for the key <1> provides you with such a command. It is assummed that you have marked as many as five blocks with :START1:, :STOP1:, :START2:, :STOP2:, etc.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******PRINT BLOCKS                   Remark.
1:==:
<SCL>P1                             Print 1st section.
<SCL>P2                             Print 2nd section.
<SCL>P3                             Print 3rd section.
<SCL>P4                             Print 4th section.
<SCL>P5                             Print 5th section.
:ND:
```

Notice that the KSM DEFINITION includes all the typing that is done manually when printing text. The print command is issued by "<SCL>P". Next the block number is typed. If some of the blocks have not been marked, then nothing gets printed by that particular command. However, the other blocks will get printed as desired.

EXAMPLE 6.8. REMOVING EXCESSIVE BLANK LINES

You may find that the text generated by an on line session contains many blank lines. You may wish to remove the excessive blank lines before printing a copy of the session. Let's look at a KSM for removing all blank lines. Afterward, we'll discuss how to modify it to eliminate only the multiple blank lines.

We will assume that a blank line results from a CARRIAGE RETURN being followed immediately by either a LINEFEED or another CARRIAGE RETURN. The following KSM performs two tasks; all CARRIAGE RETURN-LINEFEED pairs are changed to single CARRIAGE RETURNS, and all pairs of CARRIAGE RETURNS are changed to single CARRIAGE RETURNS.

KEYSTROKE MULTIPLIER DEFINITION	EXPLANATION
******REMOVE BLANK LINES	Remark.
1:==:	
:AA:	
<SUA>	Go to start of text.
<SCL>F<EN	Find a
T><LF	CARRIAGE RETURN
D><ENT>	& LINEFEED.
<DAR><SLA>	Eliminate the LINEFEED.
<SUA>	Go to start of text.
<SCL>F<EN	Find a
T><EN	pair of
T><ENT>	CARRIAGE RETURNS.
<SLA>	Eliminate one of them.
:GOTOAA:	Do this repeatedly!
:ND:	

This KSM invlolves a couple of tricks. First, we must be careful to type the five characters <, E, N, T, & >, by splitting them between two lines of text. If we put them all on the same line, then AUTOTERM would interpret them as the wedge notation for a <ENTER>. We don't want to type an <ENTER> when specifying the string to be found. We want to type the five characters for the corresponding wedge notation. The same applies when we are typing <, L, F, D, & >, to tell the FIND command to look for a LINEFEED. In the KSM DEFINITION, the <, L, F, D, & > must be spread over two lines of text.

The second trick is related to the behavior of the FIND command. If the desired string is not found, then the cursor is positioned at the end of the text. Our KSM will continue operating after all blank lines have been deleted, but it will merely BOP repeatedly without additional changes to the text. This is because the cursor is positioned at the end of the text whenever a <SHIFT-LEFT-ARROW> is executed.

The above KSM could be modified to search for CARRIAGE RETURN-LINEFEED-LINEFEEDS and triple CARRIAGE RETURNS. This would result in the elimination of all multiple blank lines, but preserve sengle blank lines.

EXAMPLE 6.9. UPPER/LOWER CASE COMMAND KEY

You may wish to associate a KSM with both the upper case and lower case versions of a given letter. It is not necessary to have two expressions of the sequence. You can merely define the lower case command key so as to execute the upper case command key. For example, the following KSM DEFINITION causes the lower case key, <z>, to generate the same keystroke sequence as the upper case key, <Z>.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******Z=Z                           Remark.
z:==:
<SCL>Z                              Do <SHIFT-CLEAR> & <Z>.
:ND:
```

EXAMPLE 6.10. DISPLAYING A MENU

You can make it convenient for yourself and others to use your KSMS by displaying a MENU of options. Suppose that you have created a set of KSM DEFINITIONS for accessing a few computer services. Suppose that <1> corresponds to calling Compuserve and signing-on, <2> is for Dow Jones News, and <3> is for a local bulletin board. The following KSM for the key <M> would display these options and take the proper action when the user responds with <1>, <2>, <3>, or <BREAK>.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******MENU FOR SIGN-ONS             Remark.
M:==:                               "M" for MENU.
<ENT><ENT>                          Two blank lines.
PLEASE PRESS KEY FOR SELECTION:     Instructions.
  1   = COMPUSERVE<ENT>
  2   = DOW JONES<ENT>
  3   = BULLETIN BOARD<ENT>
BREAK= NONE OF THESE<ENT>
<ENT>                               One blank line.
<SCL>                               Prepare for command.
:ND:
```

Notice the last keystroke is <SHIFT-CLEAR>. This puts AUTOTERM into the mode where it is waiting for a command key to be typed. If the user types <1>, <2>, or <3>, then the associated KSM will be executed. If the user types <BREAK>, then the command mode will be canceled.

This method can also be used for prompting the user to perform specific manual tasks. For instance, at a certain point a KSM might say "PLACE DISK #3 INTO DISK DRIVE NUMBER ZERO. THEN TYPE "Z" TO CONTINUE." The KSM for <Z> would handle the next stage, such as saving certain blocks of data to disk and issuing the next prompt.

6.7. CLEARING-OUT ALL PENDING KEYSTROKE MULTIPLIERS

When a KSM executes another KSM, the first KSM is held "pending". The first KSM resumes action after the second has completed its action. However, the second KSM may execute a third KSM as part of its action. In such a case, we would have two KSMS held pending, while the third is executing. When the third is completed, the second would resume execution and there would be only one pending KSM. AUTOTERM can tolerate at most nine simultaneously pending KSMS.

Example 7.11, p. 70, shows how a COMPUTER DIALOGUE can give rise to pending KSMS. A series of KSMS are executed to dial, sign-on, get the news, and sign-off. However, if an error condition occurs during dialing, we would not want to continue with sign-on, etc. The :CLEAR: can handle our problem.

The :CLEAR: statement can be included in a KSM DEFINITION. It must appear on a line by itself. :CLEAR: causes all pending KSMS to be discontinued. The following example demonstrates the use of :CLEAR:.

EXAMPLE 6.11. HANDLING AN ERROR CONDITION

An error condition could occur before or during an on line session which is being performed by a COMPUTER DIALOGUE. The error condition would be recognized when none of the anticipated responses are received from the other computer or from an autodial modem. Example 7.4, p. 62, demonstrates the recognition of an error condition. The following KSM DEFINITION shows what action might be taken.

KEYSTROKE MULTIPLIER DEFINITION	EXPLANATION
******ERROR HANDLING	Remark.
Q:==:	
ERROR<ENT>	Flag the error.
:CLEAR:	Cancel pending KSMS.
<ENT><02S>	Send a CARRIAGE RETURN.
<SBR><ENT><02S><SBR><ENT><02S>	Send some LINE BREAKS.
DISC<ENT>	Try to sign-off.
<02S>+++<02S>	Try to tell MODEM
ATH0<ENT><02S>	to hang-up.
:ND:	

The last three lines, where the modem is told to hang-up, are intended for a D. C. Hayes Smartmodem. The action would be different for some other modem.

7. BUILDING COMPUTER DIALOGUES

A COMPUTER DIALOGUE DEFINITION is a KEYSTROKE MULTIPLIER DEFINITION which includes responses from the other computer. When the KSM is being executed, the action is periodicaly halted to wait for the other computer's reply.

Your KSM DEFINITION controls the periodic halts. You can specify what the other computer's responses will be, and what your computer will do in turn. You can also list several alternate responses by the other computer, along with the corresponding actions to be taken by your computer.

The KSM SELF-TEST command enables you to test the operation of your COMPUTER DIALOGUE without actually going on line with the other computer. This is described in section 7.7, p. 73.

7.1. QUOTES INDICATE OTHER COMPUTER'S RESPONSE

Generally, "the other computer's response" consists of those characters received from the other computer and displayed on your screen. This includes CARRIAGE RETURNS and LINEFEEDS, which are displayed via the cursor slowly flashing in the appropriate form. This does not include characters ignored by AUTOTERM because of a YES seting for the IGNORE-DEL, IGNORE-LF-AFTER-CR, and/or IGNORE-NULL controls described in section 3.9, p. 31.

Throughout this section, we will assume that IGD=Y, IGL=Y, and IGN=Y. This means that our expression of "the other computer's response" will not contain any ASCII NULL or DEL characters, and will not include the first LINEFEED received immediately after a CARRIAGE RETURN has been sent or received.

The following example shows how QUOTE marks are used to indicate the other computer's responses.

EXAMPLE 7.1a. SIGNING-ON TO COMPUSERVE

Signing-on the COMPUSERVE network begins with dialing the computer, and typing CONTROL-C after the computer answers the phone. Then the computer sends a CARRIAGE RETURN, two LINEFEEDS, and the phrase "User ID: ". After you have entered your user I.D. and a CARRIAGE RETURN, the computer sends a LINEFEED and the phrase "Password: ". At this point, you enter your password and a CARRIAGE RETURN.

The following KSM DEFINITION automates this process. After the computer answers the phone, you would merely type <SHIFT-CLEAR> and then type <1>.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******COMPUSERVE SIGN-ON #1         Remark.
1:==:
#                                   Send a CONTROL-C.
"<ENT><LFD>USER ID: "               Asking for USER ID.
98765,321<ENT>                      Enter my ID.
"<LFD>PASSWORD: "                   Asking for PASSWORD.
ABCDEF<ENT>                         Enter my PASSWORD.
:ND:
```

The CONTROL-C is displayed on your screen as # , a NUMBER SIGN in reverse. It is obtained by typing <CLEAR-C>. CONTROL-C corresponds to the ASCII character ETX. See table 8.1, p. 76. Notice that each response by the other computer is started on a new line. Also, each reply by your computer is started on a new line. This "STARTING ON A NEW LINE" is required whenever a KSM DEFINITION includes responses by the other computer.

The above example is somewhat unrealistic. Normally it is not desirable to specify all the characters in the responea of the other computer. The next section presents a more reasonable version of this example.

7.2. HYPHEN INDICATES "DON'T CARE" CHARACTERS

It is not reasonable to try to specify the entire response of the other computer. Frequently, you do not know the entire response. You know only the last few characters which indicate that the other computer wants you to respond. Sometimes, you may be unsure of the exact form of the last one or two characters. Is the QUESTION MARK or COLON followed by a SPACE? Is some non-printing control character being sent by the other computer at the end of its question? You don't have to worry about such details. You can use a HYPHEN at the start and/or end of your specification of the other computer's response. This tells AUTOTERM that some "don't care" characters may be present at the beginning and/or end of the other computer's response.

7.2.1. THE LEADING HYPHEN

A HYPHEN may be placed immediately after the leading QUOTATION MARK. This LEADING HYPHEN indicates that any number of characters may be sent by the other computer before it finally sends the specified response characters.

EXAMPLE 7.1b. COMPUSERVE SIGN-ON, IMPROVED BY LEADING HYPHEN

The KSM DEFINITION of example 7.1a can be simplified by the use of LEADING HYPHENS.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******COMPUSERVE SIGN-ON #2         Remark.
1:==:
[#]                                 Send a CONTROL-C.
"-ID: "                             Asking for USER ID.
98765,321<ENT>                      Enter my ID.
"-[RD]: "                           Asking for PASSWORD.
ABCDEF<ENT>                         Enter my PASSWORD.
:ND:
```

Now AUTOTERM will look for the four characters "ID: ", when waiting for the opportunity to type the USER I.D. The "ID: " may be preceeded by any number of other characters. At the next step, AUTOTERM will wait for "rd: ", before replying with the password.

7.2.2. THE TRAILING HYPHEN

A HYPHEN may be placed immediately before the final QUOTATION MARK. This TRAILING HYPHEN indicates that as many as two characters may come after the designated characters. The other computer may send two, one, or no characters after those specified.

EXAMPLE 7.1c. COMPUSERVE SIGN-ON, IMPROVED BY TRAILING HYPHEN

Let's use some TRAILING HYPHENS in an improved version of example 7.1b.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******COMPUSERVE SIGN-ON #3         Remark.
1:==:
[#]                                 Send a CONTROL-C.
"-ID:-"                             Asking for USER ID.
98765,321<ENT>                      Enter my ID.
"-[RD]:-"                           Asking for PASSWORD.
ABCDEF<ENT>                         Enter my PASSWORD.
:ND:
```

Notice that a TRAILING HYPHEN has been used to avoid specifying whether or not the other computer's repsponse contains a SPACE at the end.

Let's look at another example of a KSM DEFINITION which controls a sign-on dialogue.

EXAMPLE 7.2a. SIGNING-ON TO DOW JONES NEWS

The sign-on to the DOW JONES NEWS SERVICE via TYMENET involves several steps:

1) You dial the computer, and wait for it to answer with a LINEFEED and the phrase "please type your terminal identifier ". You then type the letter "A".

2) The computer types a CARRIAGE RETURN, a LINEFEED, and the phrase "please log in: ". You type the six characters "DOW1;;".
3) The computer responds with the phrase "to[>] host is online", followed by three CARRIAGE RETURN & LINEFEEDS and the phrase "WHAT SERVICE PLEASE????", and a CARRIAGE RETURN and LINEFEED. [>] denotes the ASCII RS character. You reply with the four characters "DJNS" and a CARRIAGE RETURN.
4) The computer answers with a LINEFEED, the phrase "ENTER PASSWORD ", a CARRIAGE RETURN, a series of characters typed over each other, a CARRIAGE RETURN, a DC1, three SPACES, and a DC3. You respond by typing your password and a CARRIAGE RETURN. We'll assume that the password is "ABCDEF".

The following KSM DEFINITION automates this interchange.

KEYSTROKE MULTIPLIER DEFINITION	EXPLANATION
******DOW JONES SIGN-ON #1	Remark.
1:==:	
"-[IER]-"	Prompt from TYMENET.
A	My answer.
"-[IN]:-"	Next TYMENET prompt.
DOW1;;	My answer.
"-?<ENT>-"	Next prompt.
DJNS<ENT>	My response.
"- [3]-"	Prompt from DOW JONES.
ABCDEF<ENT>	My response.
"- [3]-"	Prompt from DOW JONES.
:ND:	

Notice that the ASCII DC3 character is represented as [3], the digit "3" in reverse. It is obtained by typing <CLEAR-S>. See table 8.1, p. 76.

7.3. PAUSE INDICATES END-OF-RESPONSE

AUTOTERM does not continually check for the other computer's response. It waits until the other computer pauses for two seconds. Whenever there is a two second pause, then AUTOTERM compares the characters you've placed between QUOTATION MARKS against the characters received from the other computer. If a match is found, then AUTOTERM executes the next keystroke sequence. If no match is found, then AUTOTERM waits for the next pause of two seconds and then checks again for a match, etc.

You can override the pause time by specifying a time delay at the end of the sequence. A time delay is indicated by the use of WEDGE NOTATION. See table 8.3, p. 77. Here are a few examples of time delay notation:

<05S> Five seconds
<13S> Thirteen seconds
<.3S> Three tenths of a second

If a TRAILING HYPHEN is present, then the time delay is placed after the TRAILING HYPHEN and before the QUOTATION MARK.

processed. There is at least twenty seconds of forced pause time.

7.4. :ALT: DENOTES ALTERNATE RESPONSES

You may wish to take different actions depending on the response received from the other computer. This can be done by inserting the five characters ":ALT:" on the line ahead of each alternative response from the other computer. The last alternative must be preceded by a line containing the nine characters ":LASTALT:".

7.4.1. ALTERNATE ACTION DURING SIGN-ON

Some computer services expect you to type a certain character or sequence of characters to start the session. Usually you type the character(s) and wait for a response from the other computer. If you don't get a response then you type the character(s) again. This kind of repeated trial can be included in your computer dialogue. We will look at two ways of handling such sign-on sequences.

EXAMPLE 7.3a. COMPUSERVE SIGN-ON WITH REPEATED TRIALS

Refer to example 7.1c, p. 57. We will modify this sign-on sequence to include repeated typing of the CONTROL-C, just in case our first CONTROL-C is missed by the other computer. We are faced with two alternate possibilities after the CONTROL-C has been typed. If the other computer asks for our USER I.D., then we want to proceed with the sign-on. However, if there is no response, then we want to type the CONTROL-C again. The following KSM DEFINITION handles these alternative conditions.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

*COMPUSERVE SIGN-ON WITH RE-TRY     Remark.
1:==:
:AA:                                Marker for GOTO.
[#]                                 Send a CONTROL-C.
:ALT:                               First alternative.
"-ID:-"                             Asking for USER ID.
98765,321<ENT>                      Enter my ID.
"-[RD]:-"                           Asking for PASSWORD.
ABCDEF<ENT>                         Enter my PASSWORD.
:GOTOND:                            All done.
:LASTALT:                           Second alternative.
"<03S>"                             No response in 3 seconds.
:GOTOAA:                            Do it again!
:ND:
```

After the CONTROL-C has been sent and we are waiting for a response from the other computer, we have two alternatives. If the other computer responds with the appropriate phrase, then the sign-on dialogue is performed. If the other computer makes

no response whatsoever and three seconds go by, then we go back to the point "AA". This means that the CONTROL-C is sent again and we wait for another three seconds, etc. Notice that CONTROL-C is displayed as [#], a NUMBER SIGN in reverse, and is obtained by typing <CLEAR-C>.

The above KSM DEFINITION has a major drawback. It never gives up! If the other computer never responds, then the CONTROL-C will be repeated over and over again. We can devise a system which will give up after several tries.

EXAMPLE 7.3b. COMPUSERVE SIGN-ON WITH THREE TRIALS ONLY

The following KSM DEFINITION contains three sets of alternatives. Each set of alternatives consists of waiting for either the COMPUSERVE prompt for USER I.D. or a time period of three seconds with no response. If the USER I.D. prompt is received, then the sign-on is continued. If three seconds pass, then we send CONTROL-C and do the next set of alternatives.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

*COMPUSERVE SIGN-ON WITH 3 TRIES    Remark.
1:==:
[#]                                 1st try.  Send CONTROL-C.
:ALT:                               First alternative.
"-ID:-"                             Asking for USER ID.
:GOTOAA:                            Go continue sign-on.
:LASTALT:                           Second alternative.
"<03S>"                             No response in 3 seconds.
[#]                                 2nd try.  Send CONTROL-C.
:ALT:                               First alternative.
"-ID:-"                             Asking for USER ID.
:GOTOAA:                            Go continue sign-on.
:LASTALT:                           Second alternative.
"<03S>"                             No response in 3 seconds.
[#]                                 3rd try.  Send CONTROL-C.
:ALT:                               First alternative.
"-ID:-"                             Asking for USER ID.
:GOTOAA:                            Go continue sign-on.
:LASTALT:                           Second alternative.
"<03S>"                             No response in 3 seconds.
:GOTOND:                            Give up!
:AA:                                Marker for GOTO.
98765,321<ENT>                      Enter my ID.
"-[RD]:-"                           Asking for PASSWORD.
ABCDEF<ENT>                         Enter my PASSWORD.
:ND:
```

Notice that the keystroke action following each "LASTALT" leads into the next set of alternatives. Our next example uses the keystroke action of each "ALT" to lead into the remaining alternatives.

EXAMPLE 7.4. DOW JONES SIGN-ON, WITH ERROR BRANCH

Look at example 7.2a, p. 57. At each stage, AUTOTERM will wait for the expected response. If an expected response is not received, then the KSM will never be ended. You may wish to take some special action when such a "communications breakdown" occurs. For example, if you are using an intelligent modem, you may wish to hang-up the phone and print a message on the screen. Or you may wish to re-dial the computer. The following revised KSM DEFINITION contains "ALTERNATIVES" which catch such an error in the sign-on dialogue. If an error occurs, then the KSM for the key <Q> is executed. Example 6.11, p. 54, shows a sample KSM for "Q".

```
KEYSTROKE MULTIPLIER DEFINITION    EXPLANATION

*DOW JONES SIGN-ON + ERROR CHECK   Remark.
1:==:
:ALT:
"-[IER]-"                          Prompt from TYMENET.
A                                  My answer.
:ALT:
"-[IN]:-"                          Next TYMENET prompt.
DOW1;;                             My answer.
:ALT:
"-?<ENT>-"                         Next prompt.
DJNS<ENT>                          My response.
:ALT:
"- [3]-"                           Prompt from DOW JONES.
ABCDEF<ENT>                        My response.
"- [3]-"                           Prompt from DOW JONES.
:GOTOND:                           All done!
:LASTALT:
"-<05S>"                           None of the above.
<SCL>Q                             Take action for error cond.
:ND:
```

At the start of the dialogue there are five alternatives. If the other computer does not send "-ier-", then something has gone wrong. In such a case, the other computer will stop sending characters and the last alternative will occur. This will result in the execution of the KSM for the key "Q". On the other hand, if the other computer does send "-ier-", then your machine will respond with the terminal identifier "A" and proceed to a set of four alternatives. If the other computer fails to send "-in:-", then it will probably stop sending characters and the last alternative will occur. This will trigger the execution of the KSM for the key "Q". If the other computer sends "-in:-", then your machine will respond with "DOW1;;" to continue the sign-on process. Eventually, the sign-on dialogue will be completed. If not, then the last alternative will take effect, and the KSM for the key "Q" will be executed.

7.4.2. ALTERNATE ACTION WHILE OBTAINING INFORMATION

A computer service usually sends its longer amounts of information in segments, such as, one screen-full at a time. At the end of each segment, you are expected to indicate that you desire to receive the next segment. These actions can be handled by the use of :ALT:. Let's look at an example.

EXAMPLE 7.5. GETTING A DOW JONES NEWS ARTICLE

After signing-on to the DOW JONES NEWS, you can obtain the latest news on any company by typing a PERIOD and the company's code and a CARRIAGE RETURN. You will receive a section of news followed by an ASCII RS, a DC1, three SPACES, and a DC3. You must type a CARRIAGE RETURN to receive the next section of news. When there are no more sections of news, the computer will respond with "NO PAGE", a CARRIAGE RETURN, a LINEFEED, an ASCII RS, a DC1, three SPACES, and a DC3. The following KSM asks for the latest news on TANDY CORP. and sends a CARRIAGE RETURN after each section of news is received until "NO PAGE" is received. The code for TANDY CORP is ".TAN".

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******GET TANDY NEWS                 Remark.
3:==:
.TAN                                Send code for TANDY.
:AA:
<ENT>                               Send a CARRIAGE RETURN.
:ALT:
"-NO PAGE<ENT>>1  3<.1S>"           No more news.
:GOTOND:                            All done!
:LASTALT:
"- 3<.5S>"                          Not yet at end.
:GOTOAA:                            Try for more news.
:ND:
```

Notice that the :AA: is placed between the .TAN and the <ENT>. After receiving a section of news, we jump to :AA:. The keystroke sequence which follows :AA: consists of a CARRIAGE RETURN and then two alternatives. "NO PAGE<ENT>>1 3" means that there is no more news and we stop after a .1 second pause. Any other response involving "- 3" and a .5 second pause is assumed to mean that there may be more news. Notice that a response satisfying the first alternative, would also satisfy the second alternative. However, we will act on the .1 second pause before we act on the .5 second pause.

EXAMPLE 7.6. GETTING DOW JONES NEWS ON SEVERAL COMPANIES

The previous example can be easily modified to obtain several news articles. By removing the ".TAN", we obtain a KSM which obtains the news for whatever company code had been typed. This can be coupled with a KSM which types the code of each

company. The two KSM DEFINITIONS are given below. The first one types the company codes and calls upon the second one, which types the CARRIAGE RETURNS until the end of the article is reached.

```
KEYSTROKE MULTIPLIER DEFINITIONS    EXPLANATION

*GET TANDY, EXXON, & IBM NEWS       Remark.
4:==:
.TAN                                Send code for TANDY.
<SCL>3                              Get the news.
.XON                                Send code for EXXON.
<SCL>3                              Get the news.
.IBM                                Send code for IBM.
<SCL>3                              Get the news.
:ND:
******GET A NEWS ARTICLE            Remark.
3:==:
:AA:
<ENT>                               Send a CARRIAGE RETURN.
:ALT:
"-NO PAGE<ENT>>1   3<.1S>"          No more news.
:GOTOND:                            All done!
:LASTALT:
"- 3<.5S>"                          Any other response.
:GOTOAA:                            Try for more news.
:ND:
```

Notice that the first KSM does not type the CARRIAGE RETURNS. It merely types each company code and then calls the other KSM, which types the CARRIAGE RETURN and continues to do so until there is no more news on that company.

EXAMPLE 7.7. A COMPLETE RUN ON DOW JONES NEWS

We will create a KSM which performs an entire session on the DOW JONES NEWS SERVICE. We will call upon other KSMS. Example 7.2, p. 57, is used to sign-on. Example 7.6, p. 63, is used to obtain some news during the session. We will include a KSM for the key "9" which will sign-off. The sign-off proceedure consists of typing "DISC" and a CARRIAGE RETURN. Two KSM DEFINITIONS are presented below. The first executes the KSMS for signing-on, getting the news, signing-off, and then saves the entire session onto disk under the filename NEWS. The second is the KSM for signing-off.

KEYSTROKE MULTIPLIER DEFINITIONS	EXPLANATION
*SIGN-ON, GET NEWS, & SIGN-OFF	Remark.
5:==:	
<SCL>1	Sign-on.
<SCL>4	Get the news.
<SCL>9	Sign-off.
<SCL>SC<SUA><ENT><SDA><ENT> NEWS<ENT>	Save it all to disk under the name of NEWS.
:ND:	
*******SIGN-OFF FROM DOW JONES	Remark.
9:==:	
DISC<ENT>	Send disconnect signal.
:ND:	

The <SCL>1 action is given in example 7.2, p. 57. Notice that example 7.2 calls upon <SCL>Q if an error arises during sign-on. This will result in an attempt to sign-off. The <SCL>4 action is given in example 7.6, p. 63.

EXAMPLE 7.8. GETTING MESSAGES FROM A BULLETIN BOARD

Here is the screen display of a session on the DALTRUG BULLETIN BOARD, which uses FORUM-80 (TM). The responses of the user are underlined. The user dials the BULLETIN BOARD and waits for its prompting.

SCREEN DISPLAY	EXPLANATION
.	DOTS denote additional output from the bulletin board.
.	
.	
YOUR FIRST NAME: '%JOHN	All user responses include a CARRIAGE RETURN.
YOUR LAST NAME: ''JONES	
YOUR CITY AND STATE: ''DALLAS, TX	
NAME: JOHN JONES FROM: DALLAS, TX	
IS THIS CORRECT, JOHN ? ''Y	
CHECKING USER FILES	
.	
.	
.	
(HIT 'S' TO SKIP BULLETINS) (HIT 'P' TO PAUSE)	

SCREEN DISPLAY	EXPLANATION
==========	Bulletins, if present,
.	begin with row of EQUALS.
.	User types "S" to skip
.	the bulletins.
HIT C/R TO CONTINUE [;!!!]	
02:00 COMMAND: [''R]	User types "R" to
	retreive messages.
02:12 (R) SUBCMD: [''S]	User types "S" to search
	for certain messages.
MESSAGE FILE SEARCH	
(MSGS 108 TO 269)	
START SEARCH AT MSG: [''1]	"1" has same effect as "108".
F = FROM S = SUBJECT	
T = TO C = CATEGORY	
SEARCH WHICH FIELD ? [''S]	User is searching for
	all messages related
ENTER SEARCH STRING: [''COLOR]	to COLOR COMPUTER.
START: MSG 108	
FIELD: SUBJECT	
STRING: COLOR	
IS THIS CORRECT, JOHN ? [''Y]	
.	Messages are printed
.	here.
.	
4:20 (R) SUBCMD: [''T]	User types "T" to
	terminate session.
WOULD YOU LIKE TO LEAVE	
CONFIDENTIAL COMMENTS	
TO THE SYSTEM OPERATOR (Y/N) ?	
[''N]	
.	Bulletin board signs off.
.	
.	

Notice that the BULLETIN BOARD sends some ASCII CONTROL CHARACTERS when requesting a response from the user. These characters appear on the screen as [7] , [%] , [;] , and [!] , which represent the BEL, ENQ, ESC, and SOH, respectively, as can be seen from table 8.1, p. 76.

The following KSM DEFINITION handles the entire session from sign-on to sign-off.

KEYSTROKE MULTIPLIER DEFINITION	EXPLANATION
*GET BULLETIN BOARD MESSAGES	Remark.
1:==:	
"-[/]-"	Receive prompt.
JOHN<ENT>	Send First Name.
"-[/]-"	Receive prompt.
JONES<ENT>	Send Last Name.
"-[/]-"	Receive prompt.
DALLAS, TX<ENT>	Send City & State.
"-[/]-"	Receive prompt.
Y<ENT>	Send "Y" for "YES".
:AA:	Look for several
:ALT:	possible responses.
"-==<OOS>"	If EQUALS, then
S	send "S" to "SKIP",
<.1S>	wait one-tenth second,
:GOTOAA:	& go look again.
:ALT:	
"-[!!]-"	Receive prompt for CR.
<ENT>	Send CARRIAGE RETURN.
:LASTALT:	
"-[/]-"	Receive prompt for COMMAND.
R<ENT>	Send "R" for "RETREIVE".
"-[/]-"	Receive prompt for SUBCMD.
S<ENT>	Send "S" for "SEARCH".
"-[/]-"	Receive prompt for START.
1<ENT>	Send "1" for "EARLIEST".
"-[/]-"	Receive prompt for FIELD.
S<ENT>	Send "S" for "SUBJECT".
"-[/]-"	Receive prompt for STRING.
COLOR<ENT>	Send "COLOR".
"-[/]-"	Receive prompt for CORRECT?.
Y<ENT>	Send "Y" for "YES".
"-[/]-"	Receive prompt for SUBCMD.
T<ENT>	Send "T" for "TERMINATE".
"-[/]-"	Receive prompt for COMMENTS?.
N<ENT>	Send "N" for "NO".
:ND:	

Notice the series of alternatives which handles the use of "S" to skip the bulletins. We look for either "==" with no pause, "[!!]" with a two second pause, or "[/]" with a two second pause. If the "==" is received, then we type "S", wait one-tenth second, and go back to waiting for any of the three alternatives. In this way, we are covering the possibility of our "S" not being heard. If the "[!!]" is received, then we know that the bulletin board is asking for a CARRIAGE RETURN, and we send it. If the "[/]" is received, then we know that the bulletin board is asking for our COMMAND, and we begin retrieving the desired messages.

This KSM could include an automatic SAVE-TO-DISK, as was done in example 7.7, p. 64.

7.5. AUTOMATIC DIALING

Intelligent modems are becoming increasingly available at modest cost. Such modems are able to dial the telephone and take a variety of actions depending on the results. They can also answer the telephone and let your computer converse with the callling computer. Many intelligent modems are controlled by the character sequences which are sent over the serial port of your computer. These intelligent modems can be controlled by KSMS.

In this section we will look at two specific intelligent modems. We will present KSMS for automatic dialing with the RADIO SHACK MODEM II and the D. C. HAYES SMARTMODEM.

EXAMPLE 7.9. AUTOMATIC DIALING WITH THE RADIO SHACK MODEM II

The following steps are required to acheive automatic dialing via the RADIO SHACK MODEM II:

1. Put the modem into program mode by sending an ASTERISK. The ASTERISK is echoed when the modem is ready for instruction.
2. Send the letter "D" followed by the sequence of digits which you want to dial. You should pause at least .033 second after each digit.
3. Send the character "X" to execute the dialing.
4. Wait for an appropriate time period for the dialing to be completed.

If we are going to check for receipt of the echoed ASTERISK in step 1, then we must have the DUPLEX set to HALF. We will assume that FULL DUPLEX is appropriate for the computer service being accessed. So our KSM DEFINITION begins with settig DUPLEX to HALF and ends wth re-setting DUPLEX to FULL. The command characters must be sent to the modem with PARITY set to SPACE. We will assume that the computer service requires PARITY to be EVEN. The KSM DEFINITION includes seting PARITY to SPACE at the beginning and re-setting PARITY to EVEN at the end.

The following KSM DEFINITION performs the above four steps.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******DIAL VIA MODEM II              Remark.
6:==:
<SCL>UDUP<ENT>HALF<ENT>             Set DUPLEX to HALF.
<SCL>UPAR<ENT>SPACE<ENT>            Set PARITY to SPACE.
:AA:
*                                   Send ASTERISK.
:ALT:
"*"                                 ASTERISK received.
:GOTOBB:                            Go finish dialing.
:LASTALT:
"-<03S>"                            "*" not received.
:GOTOAA:                            Go try again!
:BB:
D<.1S>3<.1S>2<.1S>1<.1S>            Send D 3 2 1.
7<.1S>6<.1S>5<.1S>4<.1S>            Send 7 6 5 4.
X                                   Send EXECUTE command.
<05S><.5S>                          Wait 5.5 seconds.
<SCL>UPAR<ENT>EVEN<ENT>             Set PARITY to EVEN.
<SCL>UDUP<ENT>FULL<ENT>             Set DUPLEX to FULL.
:ND:
```

Notice that we are waiting three seconds for the ASTERISK to be returned. We are not paying any attention to the echoing of the remainder of the command information. We are merely waiting one-tenth second between each character of command information. After sending the execute command, we wait for five and one-half seconds. This delay period depends upon the dialing mode and phone line response. The most appropriate delay would be determined by experimentation.

EXAMPLE 7.10. AUTOMATIC DIALING WITH THE D. C. HAYES SMARTMODEM

The following steps are required to acheive automatic dialing via the D. C. HAYES SMARTMODEM:

1. Get the modem's attention by sending the two characters "AT" and a CARRIAGE RETURN. The modem should respond with the digit "0" and a CARRIAGE RETURN.

2. Send the dial command by sending the character "D", and the digits of the phone number to be dialed, and a CARRIAGE RETURN. The modem will respond with the digit "1" and a CARRIAGE RETURN after the connection has been made. It can take as long as thirty seconds to respond with "1" or some other digit.

We are assuming that the modem has been set to NON-VERBOSE mode, NO-ECHO of commands, and the proper DUPLEX, etc.

The following KSM performs the above two steps.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******DIAL VIA SMARTMODEM            Remark.
6:==:
:AA:
AT<ENT>                             Send "AT".
:ALT:
"0<ENT><.1S>"                       "AT" acknowledged.
:GOTOBB:                            Go finish dialing.
:LASTALT:
"-<03S>"                            "AT" disregarded.
:GOTOAA:                            Go try again!
:BB:
D3217654<ENT>                       Send dial instruction.
:ALT:
"1<ENT>"                            Connection made.
:GOTOND:                            All done!
:LASTALT:
"-<60S>"                            Connection not made.
:GOTOAA:                            Go try again!
:ND:
```

Notice that we are waiting three seconds for the "AT" to be recognized. After giving the dial command, we are waiting sixty seconds for evidence that the connection has been made.

EXAMPLE 7.11. A COMPLETE DOW JONES SESSION INCLUDING DIALING

Example 7.7, p. 64, can easily be extended to include dialing the phone. Let's assume that 321-7654 is the correct phone number for accessing the DOW JONES NEWS service. We merely insert the autodialing at the start. The modem will automatically hang up when the CARRIER SIGNAL is lost at the end of the session.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******CALL & GET NEWS               Remark.
5:==:
<SCL>6                              Dial the phone.
<SCL>1                              Sign-on.
<SCL>4                              Get the news.
<SCL>9                              Sign-off.
<SCL>SC<SUA><ENT><SDA><ENT>         Save it all to disk
NEWS<ENT>                           under the name of NEWS.
:ND:
```

The actions for <SCL>6, 1, 4, & 9 are given in examples 7.9(10), pp. 68-69, 7.2, p. 57, 7.6, p. 63, and 7.7, p. 64, respectively.

7.6. AUTOMATIC ANSWERING

Many modems are able to monitor the telephone line and answer the phone. When a carrier signal is detected, the modem

connects the calling computer or terminal to your computer. At this point, your computer can send and/or receive information.

KSMS can handle some tasks involved with automatic answering. We will first discuss auto-answering with the RADIO SHACK MODEM II and the D. C. HAYES SMARTMODEM. Then we will look at an example of a "MESSAGE TAKER".

EXAMPLE 7.12. AUTO-ANSWERING WITH THE RADIO SHACK MODEM II

The RADIO SHACK MODEM II monitors the phone line when placed in automatic operation. When the phone rings, the modem answers and tries to make a connection via carrier tones. When a successful connection has been made, then all received data is transmitted to your computer and data sent by your computer is transmitted to the calling computer or terminal. We will assume that the calling computer or terminal transmits a CARRIAGE RETURN after connection has been made. Our KSM merely waits (perhaps for hours) until a CARRIAGE RETURN is received. It then executes the KSM for the key <8>, which is presented in example 7.14, p. 72.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******ANSWER VIA MODEM II            Remark.
2:==:
:AA:                                Start waiting.
:ALT:
"-<ENT><.1S>"                       Receive CARRIAGE RETURN.
<SCL>8                              Take message.
<10S>                               Delay for 10 seconds.
:GOTOAA:                            Go do more waiting.
:LASTALT:
"-<90S>"                            No action for 90 seconds.
:GOTOAA:                            Go do more waiting.
:ND:
```

Notice that we are waiting to receive a CARRIAGE RETURN. If it is received, then we take the desired action, pause for ten seconds, and go back to waiting. If it is not received and ninety seconds go by with no activity, then we merely go back and start waiting again. The action for <SCL>8 is presented in example 7.14, p. 72.

EXAMPLE 7.13. AUTO-ANSWERING WITH THE D. C. HAYES SMARTMODEM

The D. C. HAYES SMARTMODEM differs somewhat from the RADIO SHACK MODEM II in the way phone answering is handled. The HAYES modem can be programmed to answer after a designated number of rings have occurred. When a connection has been made, the HAYES modem sends the digit "1" and a CARRIAGE RETURN to your computer. Our KSM DEFINITION is very similar to that of example 7.12. We assume that the calling computer or terminal sends a CARRIAGE RETURN when the connection has been made. The only difference is that we look for the "1" and CARRIAGE RETURN sent

by the HAYES modem and then look for the CARRIAGE RETURN sent by the caller.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******ANSWER VIA SMARTMODEM          Remark.
2:==:
:AA:                                Start waiting for modem.
:ALT:
"1<ENT><00S>"                       Receive "1" & CR from modem.
:GOTOBB:                            Go wait for CR from caller.
:LASTALT:
"-<90S>"                            No action for 90 seconds.
:GOTOAA:                            Go do more waiting.
:BB:                                Start waiting for caller.
:ALT:
"-<ENT><.1S>"                       Receive CR from caller.
<SCL>8                              Take message.
<10S>                               Delay for 10 seconds.
:GOTOAA:                            Go do more waiting.
:LASTALT:
"-<30S>"                            No CR for 30 seconds.
:GOTOAA:                            Go do more waiting.
:ND:
```

We have two sets of alternatives. The first occurs while waiting for the modem to send a "1" and a CARRIAGE RETURN. If this occurs, we move on to the second part at :BB:. If this does not occur within ninety seconds, we go back to :AA: and start waiting again. Notice that we do not look for a time delay after the "1" and CARRIAGE RETURN. This is specified by the <00S> which overrides the default value of two seconds.

EXAMPLE 7.14. A MESSAGE TAKER

Here is a simple way to have your computer collect messages sent by another computer or terminal. Your computer will say:

> PLEASE ENTER YOUR MESSAGE.
> TYPE A CARRIAGE RETURN AT THE
> END OF EACH LINE. WAIT FOR A
> "% " BEFORE STARTING A NEW
> LINE. END THE MESSAGE BY
> TYPING TWO QUICK CARRIAGE
> RETURNS OR BY TYPING NOTHING
> FOR ONE MINUTE.

Next it will type a "% " at the start of the next line. Then it will wait for the caller to send a line of the message. After receiving the CARRIAGE RETURN at the end of the line, it will type a "% " at the start of the next line, etc. When two CARRIAGE RETURNS are received in sequence, it will say:

> THANK YOU. PLEASE HANG UP.

We assume that the phone has been answered and a successful connection has been made. After taking the message, we save it to a disk file called MSGS and then delete it from memory. This last step eliminates the possibility of running out of memory space.

```
KEYSTROKE MULTIPLIER DEFINITION     EXPLANATION

******TAKE A MESSAGE                 Remark.
8:==:
<LFD>
PLEASE ENTER YOUR MESSAGE.          Send
<ENT><LFD>                          instruction
TYPE A CARRIAGE RETURN AT THE       message.
<ENT><LFD>
END OF EACH LINE.  WAIT FOR A
<ENT><LFD>
"% " BEFORE STARTING A NEW
<ENT><LFD>
LINE.  END THE MESSAGE BY
<ENT><LFD>
TYPING TWO QUICK CARRIAGE
<ENT><LFD>
RETURNS OR BY TYPING NOTHING
<ENT><LFD>
FOR ONE MINUTE."
<ENT><LFD>
:AA:
<LFD>%                              Send "% " as prompt.
:ALT:
"-<ENT><03S>"                       Receive CARRIAGE RETURN.
:GOTOAA:                            Go get next line.
:ALT:
"-<ENT><ENT>"                       Receive two CR's.
:GOTOBB:                            Go say thanks.
:LASTALT:
"-<60S>"                            No action for one minute.
:BB:                                End of message.
<LFD>                               Say
THANK YOU.  PLEASE HANG UP.         thank you!
<ENT><LFD>
<SCL>SC<SUA><ENT><SDA><ENT>         Add the message to
MSGS<ENT>                           the MSGS file.
<SCL>D<SUA><ENT><SDA><ENT>          Delete it from memory.
:ND:
```

We are assuming that a CARRIAGE RETURN followed by a three second pause means the end of a line. If the caller types two CARRIAGE RETURNS with a three second pause between them, then we will treat the first one as the end of a line. That is why we say "QUICK" in our instruction message.

7.7. SELF-TEST OF DIALOGUE ACTION

KEYSTROKE MULTIPLIER SELF-TEST enables you to test a COMPUTER DIALOGUE without actually calling the other computer. You play the role of the other computer by typing at your computer's keyboard.

To SELF-TEST any KSM do the following:

1. Enter SEND/RECEIVING operation, but do not connect with the other computer.

2. Type <SHIFT-CLEAR>. The top of the screen will show:

```
TYPE COMMAND OR <?>    NNNNN M L
```

3. Now type <K>, for KEYSTROKE MULTIPLIER SELF-TEST. The top of the screen will show:

```
TYPE COMMAND TO TEST   NNNNN M L
```

4. At this point, type the key for the KSM which you wish to test.

AUTOTERM will execute the KSM in SELF-TEST MODE. Whenever the KSM is waiting for a response from the other computer, then you can type at the keyboard and AUTOTERM will treat your typed characters as though they had been received from the other computer. You can use the <CLEAR> key as a CONTROL KEY. You can generate any of the 128 ASCII characters as indicated in table 8.1, pp. 76-77, except for the ESCAPE, which involves the <BREAK> key. Any use of <BREAK> will cancel the KSM and end the SELF-TEST as well.

During SELF-TEST MODE, you can act like the other computer only when a KSM is waiting for a response from the other computer. You cannot use the keyboard while the KSM is typing or delaying. If you do, you will receive a BOP, and your keystroke will be ignored, except for the <BREAK> key.

The SELF-TEST MODE is carried over into any other KSMS which are executed as a result of the KSM being tested. The SELF-TEST MODE ends when all KSM action ceases.

8. TECHNICAL SUMMARY

8.1. SCREEN DISPLAY

OPERATING INFORMATION ON 1ST LINE - The top line displays, from left to right, the OPERATING MODE, the NUMBER OF CHARACTERS AVAILABLE FOR ADDITIONAL STORAGE OF TEXT, an "M" if SAVE-TO-MEMORY is ON, and an "L" or "U" for keyboard being LOWER CASE or UPPER-CASE-ONLY.

PROMPTING ON 1ST & 2ND LINES - The top one or two lines display prompts for various user operations.

TEXT DISPLAY ON 2ND & SUBSEQUENT LINES - Table 8.1 shows how each ASCII character is displayed. Some of the unusual characters have peculiar displays. A CARRIAGE RETURN or LINEFEED causes the start of a new line. These characters are displayed via a slowly flashing cursor. The cursor flashes slowly in the appropriate form when positioned over a CARRIAGE RETURN or LINEFEED.

FLASHING CURSOR - The flashing cursor is usually positioned after the last text character. However, it can be positioned over any of the text characters, except those trailing blanks which are not actually part of the text.

8.2. KEYBOARD ACTION

KEY BEEP & BEEBOP - A BEEP is sounded by any key except the <SHIFT> or <CLEAR>. If the pressed key is not appropriate for the current operating mode, then a low pitched BOP will follow the BEEP. This BEEBOP indicates that the pressed key is being ignored.

CARRIAGE RETURN - <ENTER> acts as a CARRIAGE RETURN.

CONTROL KEY - <CLEAR> acts as an ASCII control key. Table 8.1 shows which characters require use of the <CLEAR> key.

COMMAND KEY - <SHIFT-CLEAR> tells the system that the next key pressed will be a single key command.

CANCEL KEY - <BREAK> tells the system to cancel the current command or operating mode. Accumulated text is not lost.

LINE BREAK - <SHIFT-BREAK> generates a LINE BREAK when the operating mode is SEND/RECEIVING.

ARROW KEYS - The four ARROW keys move the cursor through the text and also cause scrolling. The <SHIFT-UP-ARROW> (or <SHIFT-DOWN-ARROW>) moves the cursor to the start (or end) of the text. The <LEFT-ARROW> generates a BACKSPACE during SEND/RECEIVING operation.

TABLE 8.1.
ASCII CHARACTER ENTRY & DISPLAY

ASCII	CONTROL CHARACTER	HEX VALUE	KEYBOARD	TV DISPLAY
NUL	Null	00	CLEAR @	
SOH	Start of Heading	01	CLEAR A	!
STX	Start of Text	02	CLEAR B	"
ETX	End of Text	03	CLEAR C	#
EOT	End of Transmission	04	CLEAR D	$
ENQ	Enquiry	05	CLEAR E	%
ACK	Acknowledge	06	CLEAR F	&
BEL	Bell	07	CLEAR G	'
BS	Backspace	08	CLEAR H	(
HT	Horizontal Tabulation	09	CLEAR I	)
LF	Line Feed	0A	CLEAR J	*
VT	Vertical Tabulation	0B	CLEAR K	+
FF	Form Feed	0C	CLEAR L	,
CR	Carriage Return	0D	ENTER or CLEAR M	-
SO	Shift Out	0E	CLEAR N	.
SI	Shift In	0F	CLEAR O	/
DLE	Data Link Escape	10	CLEAR P	0
DC1	Device Control 1 (XON)	11	CLEAR Q	1
DC2	Device Control 2	12	CLEAR R	2
DC3	Device Control 3 (XOFF)	13	CLEAR S	3
DC4	Device Control 4	14	CLEAR T	4
NAK	Negative Acknowledge	15	CLEAR U	5
SYN	Synchronous Idle	16	CLEAR V	6
ETB	End of Transmiss'n Block	17	CLEAR W	7
CAN	Cancel	18	CLEAR X	8
EM	End of Medium	19	CLEAR Y	9
SUB	Substitute	1A	CLEAR Z	:
ESC	Escape	1B	CLEAR BREAK	;
FS	File Separator	1C	CLEAR 4	<
GS	Group Separator	1D	CLEAR 5	=
RS	Record Separator	1E	CLEAR 6	>
US	Unit Separator	1F	CLEAR 7	?
DEL	Delete or Delay	7F	CLEAR 0	←

ASCII CHAR	HEX VALUE	KEY-BOARD	TV DISPLAY	ASCII CHAR	HEX VALUE	KEY-BOARD	TV DISPLAY
SP	20	SPACE-BAR		!	21	SHIFT 1	!
"	22	SHIFT 2	"	#	23	SHIFT 3	#
$	24	SHIFT 4	$	%	25	SHIFT 5	%
&	26	SHIFT 6	&	'	27	SHIFT 7	'
(	28	SHIFT 8	(	)	29	SHIFT 9	)
*	2A	SHIFT :	*	+	2B	SHIFT ;	+
,	2C	,	,	-	2D	-	-
.	2E	.	.	/	2F	/	/

(Continued on next page)

TABLE 8.1.
ASCII CHARACTER ENTRY & DISPLAY
(Continued from previous page)

ASCII CHAR	HEX VALUE	KEY-BOARD	TV DISPLAY	ASCII CHAR	HEX VALUE	KEY-BOARD	TV DISPLAY
0	30	0	0	1	31	1	1
2	32	2	2	3	33	3	3
4	34	4	4	5	35	5	5
6	36	6	6	7	37	7	7
8	38	8	8	9	39	9	9
:	3A	:	:	;	3B	;	;
<	3C	SHIFT ,	<	=	3D	SHIFT -	=
>	3E	SHIFT .	>	?	3F	SHIFT /	?
@	40	@	@	A	41	SHIFT A	A
B	42	SHIFT B	B	C	43	SHIFT C	C
D	44	SHIFT D	D	E	45	SHIFT E	E
F	46	SHIFT F	F	G	47	SHIFT G	G
H	48	SHIFT H	H	I	49	SHIFT I	I
J	4A	SHIFT J	J	K	4B	SHIFT K	K
L	4C	SHIFT L	L	M	4D	SHIFT M	M
N	4E	SHIFT N	N	O	4F	SHIFT O	O
P	50	SHIFT P	P	Q	51	SHIFT Q	Q
R	52	SHIFT R	R	S	53	SHIFT S	S
T	54	SHIFT T	T	U	55	SHIFT U	U
V	56	SHIFT V	V	W	57	SHIFT W	W
X	58	SHIFT X	X	Y	59	SHIFT Y	Y
Z	5A	SHIFT Z	Z	[	5B	CLEAR 8	[
\	5C	CLEAR /	\	]	5D	CLEAR 9	]
^	5E	CLEAR 3	↑	_	5F	CLEAR -	←
`	60	CLEAR ;	@	a	61	A	A
b	62	B	B	c	63	C	C
d	64	D	D	e	65	E	E
f	66	F	F	g	67	G	G
h	68	H	H	i	69	I	I
j	6A	J	J	k	6B	K	K
l	6C	L	L	m	6D	M	M
n	6E	N	N	o	6F	O	O
p	70	P	P	q	71	Q	Q
r	72	R	R	s	73	S	S
t	74	T	T	u	75	U	U
v	76	V	V	w	77	W	W
x	78	X	X	y	79	Y	Y
z	7A	Z	Z	{	7B	CLEAR ,	[
|	7C	CLEAR 1	\	}	7D	CLEAR .	]
~	7E	CLEAR 2	↑	DEL	7F	CLEAR 0	←

1. CLEAR key acts like a CONTROL KEY. Hold down the CLEAR key while typing the other key.
2. ☐ indicates that display has a reverse background. If ULD = 4 thru 8, then reverse background is used only for CONTROL CHARACTERS.

SHIFT LOCK ON/OFF TOGGLE - <SHIFT-ZERO> causes the keyboard to alternate between LOWER CASE and UPPER CASE ONLY.

MEMORY ON/OFF TOGGLE - Typing <CLEAR-UP-ARROW> during SEND/RECEIVING causes SAVE-TO-MEMORY to alternate between ON and OFF.

RESET - When "all else fails", press the RESET button at the right rear of the computer to return to the MAIN MENU. Accumulated text will not be lost.

RETURN TO BASIC - When the MAIN MENU is beign displayed, type <SHIFT-BREAK> to return to BASIC via a cold start.

8.3. USER OPTIONS

Table 8.2 lists the various operating parameters which can be altered via the "U" command.

8.4. SYSTEM OPERATION

MAIN MENU - At start-up the MAIN MENU offers three operating modes: TEXT PROCESSING, INTELIGENT TERMINAL, or KEYSTROKE MULTIPLIER DEFN. While in any of these modes, typing <BREAK> once or twice triggers a return to the MAIN MENU, without loss of the text accumulated in memory.

TEXT PROCESSING - AUTOTERM acts like a simplified word processor. All system commands are available except for TRANSMIT BLOCK & KSM SELF-TEST. This mode is used for the creation of new text &/or modification of text previously accumulated in memory during INTELLIGENT TERMINAL operation.

INTELLIGENT TERMINAL - AUTOTERM acts like a high powered computer terminal. The SERIAL I/O PORT is the connection to the other computer. All system commands are available except for PRINT BLOCK, which can be used only if the printer is connected somewhere other than the SERIAL I/O PORT.

KEYSTROKE MULTIPLIER DEFN - AUTOTERM displays the accumulated KSM DEFINITIONS, with SWW=N and ULD=1. All system commands are available except for TRANSMIT BLOCK & KSM SELF-TEST. This mode is used for creation &/or modification of KSM DEFINITIONS. Accumulated text is maintained in memory.

SYSTEM COMMANDS - Type <SHIFT-CLEAR> to initiate any system command. Then type the command's single letter identifier. Prompting appears on the top one or two lines of the screen. Type <BREAK> to cancel the command at any time. The available commands are:

TABLE 8.2.
USER OPTIONS

These operating parameters may be set via the "U" command.

CODE	DESCRIPTION	RANGE OF VALUES	INITIAL VALUE
GENERAL:			
BEP	Keyboard BEEP	Y or N	Y
BOP	Keyboard BOP	Y or N	Y
SWW	Screen Word Wrap	Y or N	Y
ULD	Upper/Lower Case Display	1 thru 8	1
COMMUNICATIONS:			
CBD	Communications Baud Rate	110*, 150, 300, 600, or 1200	300
DUP	Communications Duplex	HALF or FULL	FULL
IGD	Ignore the DEL character	Y or N	Y
IGL	Ignore the LINEFEED received after a CARRIAGE RETURN is sent or received	Y or N	Y
IGN	Ignore the NULL character	Y or N	Y
PAR	Communications Parity	OFF, EVEN, ODD, MARK, or SPACE	EVEN
SCR	Scrolling while Receiving	Y or N	Y
SLF	Send LINEFEED after CARRIAGE RETURN during BLOCK TRANSMISSION	Y or N	N
STP	Stop Bits	1 thru 125	1
XNF	ON/OFF CHARACTERS during BLOCK TRANSMISSION	0000 thru FFFF	0000
PRINTING:			
BMG	Bottom Margin (lines)	0 thru 125	6
CNP	Continuous Printing	Y or N	N
LMG	Left Margin (spaces)	0 thru 125	6
LSP	Line Spacing (lines)	1 thru 125	1
OCR	Omit "artificial" CARRIAGE RETURNS	Y or N	N
PGL	Page Length (lines)	0 thru 255	66
PGP	Page Pause	Y or N	N
PGW	Page Width (spaces)	1 thru 255	80
PLF	Print LINEFEED after CARRIAGE RETURN	Y or N	N
PWW	Printer Word Wrap	Y or N	Y
RMG	Right Margin (spaces)	0 thru 125	6
TMG	Top Margin (lines)	0 thru 125	6

* Operation at 110 BAUD does not tolerate simultaneous sending while receiving. This is tolerated at the other BAUD RATES.

C = CHECKSUM A BLOCK OF TEXT OR DATA

D = DELETE A BLOCK/FILE OF TEXT OR DATA

E = EDIT TEXT DURING INTELLIGENT TERMINAL OPERATION

F = FIND A STRING OF CHARACTERS

G = GET A DISK DIRECTORY

H = HEX CODE ENTRY

K = KEYSTROKE MULTIPLIER SELF-TEST

L = LOAD A BLOCK OF TEXT OR DATA FROM DISK

M = MAINTAIN A DISK COPY OF THE SESSION

P = PRINT A BLOCK OF TEXT

S = SAVE A BLOCK OF TEXT OR DATA TO DISK

T = TRANSMIT A BLOCK/FILE OF TEXT OR DATA

U = USER OPTIONS - DISPLAY/CHANGE

Many of the commands involve a BLOCK (section) of text or binary data. A BLOCK'S START and STOP points can be specified by embedding :START1:, :STOP1:, . . . :START9:, :STOP9: in the text or by positioning the cursor in response to AUTOTERM's prompts.

8.5. KEYSTROKE MULTIPLIERS (KSM)

A KEYSTROKE MULTIPLIER DEFINITION associates a sequence of keystrokes with a single key. The sequence is triggered by typing <SHIFT-CLEAR> and then the key. Typing <BREAK> halts AUTOTERM's execution of the sequence.

The KEYSTROKE MULTIPLIER DEFN mode is used for displaying & editing the KSM DEFINITIONS. The following format rules apply:

1) The first line consists of five characters; namely, the character being defined, followed by :==:.

2) CARRIAGE RETURNS and LINEFEEDS are not considered as part of the sequence, unless represented by WEDGE NOTATION. That is, the sequence should include <ENT> in place of <ENTER>, and <LFD> in place of <CLEAR-J>. The <ENTER> and <CLEAR-J> may be used for formatting the sequence on the screen.

3) The WEDGE NOTATION (see table 8.3) must be used in place of <ENTER>, <CLEAR-J>, <BREAK>, <SHIFT-BREAK>, <SHIFT-CLEAR>, and all ARROW keys.

4) The sequence may include TIME DELAYS. These are designated by use of WEDGE NOTATION as shown in table 8.3.

5) A line may contain a "TWO CHARACTER LABEL" bounded by COLONS, such as :AA:. Nothing else may appear on such a LABEL line. LABELS are not considered as part of the sequence. They are used with GOTO commands.

6) A line may contain :ALT: or :LASTALT:. Nothing else may appear on such a line. These lines are used for designation of "ALTERNATES" in COMPUTER DIALOGUES. See section 8.6.

7) A line may contain a GOTO command, such as :GOTOAA:. Nothing else may appear on such GOTO lines. When the line is reached during execution of the sequence, then the sequence is continued at the designated label.

8) A line may contain :CLEAR:. Nothing else may appear on the line. When the line is reached during execution, then all pending KSMS are cancelled.

9) The last line consists of the four characters :ND:.

10) Remarks may be placed on separate lines ahead of the first line and/or after the last line of each definition. Remark lines require no special designator.

TABLE 8.3.
WEDGE NOTATION

KEYSTROKE DESCRIPTION	WEDGE NOTATION
CARRIAGE RETURN or ENTER	<ENT>
LINEFEED or CLEAR-J	<LFD>
BREAK	<BRK>
SHIFT-BREAK	<SBR>
SHIFT-CLEAR	<SCL>
UP-ARROW	<UAR>
SHIFT-UP-ARROW	<SUA>
DOWN-ARROW	<DAR>
SHIFT-DOWN-ARROW	<SDA>
LEFT-ARROW	<LAR>
SHIFT-LEFT-ARROW	<SLA>
RIGHT-ARROW	<RAR>
SHIFT-RIGHT-ARROW	<SRA>
CLEAR-UP-ARROW	<CUA>
Time Delay of .N seconds	<.NS>
Time Delay of NN seconds	<NNS>

8.6. COMPUTER DIALOGUES

A COMPUTER DIALOGUE is a KSM which includes RESPONSES from the other computer during SEND/RECEIVING. The first portion of the sequence of keystrokes is executed. Then AUTOTERM waits for the proper RESPONSE from the other computer. When this RESPONSE is received, then the next portion of the sequence of keystrokes

is executed, etc. The keystroke sequence may include ALTERNATIVE ACTIONS based upon several possible RESPONSES from the other computer.

OTHER COMPUTER'S RESPONSE - The RESPONSE by the other computer is placed on a separate line and between QUOTES. The following rules apply:

1) At most twenty-four response characters may appear in the set. The following keystrokes may not be included: <BREAK>, <SHIFT-BREAK>, <CLEAR>, <SHIFT-CLEAR>, and all ARROW keys.

2) CARRIAGE RETURN and LINEFEED must be represented as <ENT> and <LFD>, respectively.

3) The HYPHEN may be used as the first character, that is, only immediately after the first QUOTE MARK. When used in this manner, it indicates that the specified response could be preceded by any number of other characters. Most responses involve use of the HYPHEN in this way.

4) The HYPHEN may be used as the last character, that is, immediately before the ending QUOTE MARK or the ending time delay. When used in this manner, a HYPHEN indicates that the specified response may be followed by as many as three additional characters.

5) The sequence may include a time delay at the end only, that is, just before the ending QUOTE MARK. The delay should be expressed in WEDGE NOTATION, as shown in table 8.3. The time delay is used by AUTOTERM to recognize when the other computer has finished sending its response. If a time delay is not specified at the end of the sequence, then AUTOTERM uses two seconds.

ALTERNATE ACTIONS - When AUTOTERM encounters an :ALT: during execution of a KSM, then it does the following:

1) It searches for all successive occurrences of :ALT: until it finds a :LASTALT:. The associated RESPONSES are considered to be the ALTERNATIVE RESPONSES. At most thirteen alternatives may occur in a single series.

2) It looks for receipt of any one of the ALTERNATIVE RESPONSES.

3) As soon as any of the looked-for RESPONSES is received, then AUTOTERM continues execution of the KSM at the point following that RESPONSE.

RECOGNITION OF A RESPONSE - AUTOTERM looks for a PAUSE in the stream of data being received from the other computer. The PAUSE must be as long as the delay specified at the end of the particular RESPONSE, or two seconds, if no delay is specified. When such a PAUSE occurs, then AUTOTERM tries to match the received data to the SPECIFIED RESPONSE. If a match is found, then AUTOTERM continues execution of the KSM at the point following the QUOTE MARK at the end of the RESPONSE specification. If no match is found, then AUTOTERM waits for more data to be received and then looks again for the required PAUSE, before trying again to match that particular RESPONSE.

PXE Computing

11 Vicksburg Lane
Richardson, TX 75080
(214) 699-7273

AUTOTERM
DISK VERSION 5
OPERATIONAL RESTRICTIONS

1. DISCONNECT THE JOYSTICKS - They may interfere with keyboard action, such as the arrow keys repeating when held down.

2. TELEWRITER FILES SHOULD END WITH A CARRIAGE RETURN - A file which does not end with a CARRIAGE RETURN will cause a read error when loading into TELEWRITER. When saving data which is to be read into TELEWRITER (ASCII FORMAT) be careful to have a CARRIAGE RETURN at the end.

3. MID-LINE CARRIAGE RETURNS MAKE BASIC PROGRAMS UNLOADABLE - A bulletin board may insert CARRIAGE RETURNS in the middle of long lines when downloading a BASIC program. You should delete these extra CARRIAGE RETURNS before saving the program to disk. Otherwise, it will cause a DS ERROR when it is loaded into your computer as a BASIC program. The 51 or 64 character screen width can help you spot these unwanted CARRIAGE RETURNS.

4. PRINTER PERFORATION SKIP UPSETS PAGING AND MARGINS - If your printer has the PERFORATION SKIP turned on, then the line count will conflict with the normal page length of 66 lines. In such a case, set TMG = 0 and BMG = 0.

5. FORMAT JDOS DISKS WITH " DSKINI0,R " - This formats the disk so that it is compatible with the RADIO SHACK operating system. Other JDOS formats result in files being incompatible between JDOS and AUTOTERM.

If you uncover an unlisted restriction or bug, please describe it below and cut along the dashed line and send it to PXE Computing. Thank you.

DEAR COLOR COMPUTER OWNER,

AUTOTERM HAS OCCUPIED ALL OF MY SPARE TIME FOR WELL OVER TWO YEARS. JIM WHITAKER SPENT MANY, MANY HOURS ON THE DISK VERSION. WE ARE AMAZED (& EMBARRASSED) AT HOW MUCH WORK HAS GONE INTO THIS PROGRAM.

BOTH OF US HAVE ENJOYED THE WORK. WE SINCERELY HOPE THAT YOU ENJOY THE RESULTS!

Chet Zornes

11/23/83

AUTOTERM!

...AT A GLANCE

KEYBOARD BEHAVIOR

RESET — Return to MAIN MENU
BREAK — Cancel command or mode
ARROW KEYS — Move CURSOR
SHIFT UP ARROW — Jump to START
SHIFT DOWN ARROW — Jump to END
SHIFT LEFT ARROW — Delete character
SHIFT RIGHT ARROW — Insert SPACE
CLEAR UP ARROW — Memory ON/OFF
SHIFT ZERO — ALL-CAPS ON/OFF
SHIFT CLEAR — Initiate command or KSM
SHIFT BREAK — Send LINE BREAK

CONTROL KEY	KEYBOARD	TV
CONTROL-C	CLEAR C	
BACKSPACE	CLEAR H	
LINEFEED	CLEAR J	
FORMFEED	CLEAR L	
DC1, XON	CLEAR Q	
DC3, XOFF	CLEAR S	
ESCAPE	CLEAR-BREAK	

SYSTEM COMMANDS

BLOCK/FILE OPERATIONS
C = Checksum L = Load D = Delete
T = Transmit S = Save P = Print
Use CURSOR or :START1:d:STOP1:,etc.

E = Edit while on line
F = Find a string
* is wild card
G = Get disk directory

H = Enter as two Hex digits
K = KSM Self-Test
M = Maintain session file
U = Display/change User options

BLOCK TRANSMISSION

TO CHECK TRANSMISSION SPEED
Upper left of TV changes at 100 chars.

TO SEND TEXT ONE LINE AT A TIME USE
the XNF option: e.g. XNF: = 3A0D for COLON/CR

BINARY DATA

LOADING/SAVING: Use as Header
□ BB or □ ML
S = hhhh
T = hhhh
□ is Hex FF

SENDING/RECEIVING: Use
PAR = OFF IGD = N IGL = N IGN = N

USER OPTIONS

GENERAL

BEP Key beep Y/N — BOP Error bop Y/N — SWW Word wrap Y/N — ULD U/L Display 1 to 8

COMMUNICATIONS

CBD	Baud rate	110,150,300 600,1200
DUP	Duplex	HALF, FULL
IGD	Ignore DEL	Y/N
IGL	Ignore LFD	Y/N
IGN	Ignore NUL	Y/N
PAR	Parity	OFF, EVEN, ODD, MARK, SPACE
SCR	Scrolling	Y/N
STP	Stop bits	1 to 125
SLF	Send LFD	Y/N
XNF	Trans ON/OFF	0000 to FFFF

PRINTING

BMG	Bottom margin	0 to 125
CNP	Continuous	Y/N
LMG	Left margin	0 to 125
LSP	Line spacing	1 to 125
OCR	Omit CR's	Y/N
PGL	Page length	0 to 255
PGP	Page pause	Y/N
PGW	Page width	1 to 255
PLF	Print LFD	Y/N
PWW	Word wrap	Y/N
RMG	Right margin	0 to 125
TMG	Top margin	0 to 125

KEYSTROKE MULTIPLIERS

USE OF COLONS

x:==:	Start definition
:xx:	Label
:GOTOxx:	Jump to xx
:CLEAR:	Cancel pending KSMS
:ALT:	Alternative, but not last
:LASTALT:	Last alternative in set
:ND:	End definition

OTHER COMPUTER'S RESPONSE

"xxx" "-xxx" "xxx-" "-xxx-" "-"
"<nnS>" "xxx<nnS>" "-<nnS>"
"-xxx<nnS>" "xxx-<nnS>" "-xxx-<nnS>"

WEDGE NOTATION

<ENT> <LFD> <BRK> <SBR> <SCL>
<UAR> <SUA> <DAR> <SDA> <LAR> <SLA>
<RAR> <SRA> <CUA> <.nS> <nnS>

SAFELY
SAFELY

PXE Computing
11 Vicksburg Lane • Richardson, TX 75080 • 214/699-7273