Appendix A
Communication Concepts

This appendix introduces the general concepts of computer communication.

Computer communication allows you to send and receive information to and from other computers or to use one of the many timesharing services that are now available. Communication can occur either via a cable that is directly connected to another computer or via telephone lines.

Data representation in a computer. Within a computer, data are represented by pulses of electricity. The presence of a pulse is equivalent to a one and the absence of a pulse to a zero. Thus all system functions occur in terms of zeros and ones. These are combined in numeric codes to represent the letters of the alphabet, digits, and special characters. Each zero or one is referred to as a bit. A combination of bits representing one character or digit is called a byte. Bits and bytes are the basic elements of communication.

Data transportation between computers. Before a message can be sent across telephone lines, it must be changed into a form with characteristics that match the human voice. A modem (also called a data set) does this. The word modem is an acronym for *mod*ulator/*dem*odulator. This describes what the equipment does: it translates digital pulses into voice-compatible modulation for transmission. On the receiving end, the modem converts the signal back to a digital pulse.

Communication rates. The unit of measure for communication rates is termed the *baud rate,* which means the number of bits

transferred per second. The most common baud rates for communication are 300 and 1200. These rates translate into roughly 30 and 120 characters per second, respectively. If you own a 1200-baud modem, you can communicate at either rate. Data transfer between two computers via a cable (instead of a telephone line) can be performed at rates up to 9600 baud. For either method of communication, however, you must make sure that both sender and receiver are set to the same baud rate.

Pulse vs. touch-tone dialing. There are two ways your telephone can dial: pulse or touch-tone. Pulse dialing is the older of the two methods; it is used by rotary telephone units. It converts each digit dialed into that number of pulses or clicks. Touch-tone dialing is the method used by touch-tone telephones. It converts each digit pressed into a particular tone.

Even though you may own a touch-tone telephone, your local telephone company may still be using the pulse-tone method. If you are not sure which type of dialing method you have, contact the business office of your local telephone company. MTERM has two command menu options, I (Modem switch) and N (Auto-dial telephone), that refer to dialing types.

More about modems. A modem is referred to as a peripheral device. One type uses an acoustic coupler which looks like a pair of earphones sitting on top of the modem. You place the telephone receiver in it during transmission. Another type, referred to as a direct-connect modem, plugs directly into the wall outlet.

Line quality. Everyone knows what it is like to be unable to talk due to static and noise on a telephone line. This noise causes problems for data communication as well. If you are using a 1200-baud modem or live in a rural area, you might have problems with line quality. If you do, contact the business office of your local telephone company for help.

Communication lines. There are two types of communication lines:

- Dial-up lines. When transmission is required infrequently and when a transmission speed no greater than 2400 baud is required, a *dial-up* line is usually used. A dial-up line requires

that you place a telephone call before communication can begin. When data transmission is finished, you hang up the phone. This saves money because you only pay for the time the line is actually connected. Most communication today uses dial-up lines.

- Dedicated lines. If transmission takes place long distance and if the line is connected for long periods of time, it is more cost effective to lease a *dedicated* line. This type of line remains connected at all times.

Communication types. There are two forms of communication available using MTERM, either direct communication between two computers or communication that uses a network.

- QX-10 to another microcomputer. This form of communication can be operator-attended or unattended. In either case, data files can be *uploaded*, that is, transmitted from your QX-10 to another microcomputer, or *downloaded*, received by your system from another computer.

- Communication networks. This form of communication involves a group of computers and/or terminals that communicate with each other. The number of communication services available for microcomputers is growing daily. Some networks serve special interest groups, while others serve only their paid subscribers.

Some networks have been set up as public news services to provide local and international news, weather information, sports information, financial market prices, and local agricultural market prices. These systems usually do not accept messages because they are for the dissemination of information only.

Other services allow users to send and receive confidential messages that can be accessed and read only by the addressee.

Two of the most popular such networks are THE SOURCE and CompuServe; these services require you to subscribe and charge you for the time you are connected to the service.

These services are available over national electronic networks.

To find out what services are available in your community, check with your Epson dealer.

Buffers. A buffer is an area of space in the computer's memory that is used to store information. When you load MTERM, for example, you are storing it in a particular buffer which is larger than the program alone. You will use the remainder of the buffer to store incoming or outgoing information. As you use MTERM, you will become aware of what is in its buffer at any particular moment.

Appendix B
Technical Notes

This appendix gives technical information about MTERM. The average user of MTERM does not need this information, but experienced programmers may want information at this level of detail.

MTERM.DAT File Layout

MTERM.DAT is a file that contains the default feature settings. Additional files with alternate settings should use the same format. The defaults are loaded when the MTERM program is initialized at the beginning of communication. An alternate file can be used by typing its name immediately after pressing **MAIL**.

Unless you *always* specify an alternate filename, you must have a file named MTERM.DAT. If you press **MAIL** and **RETURN** when no MTERM.DAT file exists, MTERM cannot function and immediately exits to the CP/M-80 A > prompt.

MTERM.DAT has 12 sections. The first section contains a number of characters that either serve as flags or contain stored values pertinent to the system. Items marked with an asterisk (*) are for internal system use only. Do not alter information reserved for system use.

Here are descriptions of each MTERM.DAT section and of each character within the section.

Section 1

Byte 0: Status flags

Bit	*Meaning (when set)*
0	Output to printer on
1	Half duplex
2	LF suppression on
3	CR suppression on
4	Buffer open
5	Redraw display
*6	Transmit buffer on
7	Uppercase only enabled

Byte 1: Status flags

Bit	*Meaning (when set)*
0	Prompted output on
1	Send MacroKey
2	Device flag
*3	Automatic transmit on
*4	Hang up phone after automatic transmit
5	XON/XOFF flag
6	MacroKey flag
7	Reserved

Byte 2: Modem type

Value	*Modem type*
00	One
01	Two
02	Three
03	Four

Bytes 3 & 4: True break length. These two characters contain the length of the true break to be transmitted when **STOP** is pressed.

Byte 5: Display width. This character stores the display width in binary. Display width can be anywhere from 1 to 255 characters. The largest value you can set is 99. You can increase the display width past that value by altering this character.

Bytes 6 & 7: Character delay count for automatic log-on. These characters store the running delay counter, which controls pauses when MTERM transmits the automatic log-on message. They are for internal system use only.

Bytes 8: Buffer transmission delay counter. This character serves the same purpose as characters 6 and 7 above, with one additional feature; it keeps track of the character delay count when MTERM is transmitting from the buffer. This character is for internal system use only.

Byte 9: Buffer transmission delay value. MTERM uses this character to store the number of units of delay that you specify when you begin transmitting. MTERM ignores any subsequent modification that you make while transmission is in progress.

Bytes 10, 11 & 12: Transmit time. These characters contain the time MTERM will automatically dial the phone and transmit from the buffer. Byte 10 stores hours, byte 11 stores minutes, and byte 12 stores seconds.

Byte 13: Phone number selected for automatic transmit. This character controls the number (if any) you have selected to be dialed at the auto transmit time. If the byte is set to 00, no phone number has been selected, and the buffer transmits immediately. Otherwise the byte contains a value between 1 and 0 (01H and 0AH) to indicate which phone number (A - J) will be dialed.

Bytes 14 & 15: RS-232 base port address. These two bytes contain the starting port address for the RS-232 hardware.

Byte 16: 8259 init code. This byte contains the value that initializes the 8259 chip, which controls I/O interfacing. This value should remain unchanged.

Byte 17: CP/M-80 interrupt slot for RS-232C driver. This byte controls what slot is used for the RS-232 driver. It is configured for card 0.

Bytes 18 through 21: Reserved for future use.

Byte 22: Baud rate code. This byte contains a value (O - F) that determines which of 15 baud rates MTERM will use. This is not the actual baud rate, but rather an indicator to a baud rate.

Byte 23: Line-control register code. This byte contains the code used to configure the line-control register, which controls all areas of data transmission (word length, parity, etc.) except the baud rate.

Byte 24: Modem-control register code (normal). This byte contains the code used to configure the modem-control register when it is in normal (no carrier) state.

Byte 25: Modem-control register code (with carrier). This byte contains the code used to configure the modem-control register when the modem is in on-line state and the carrier has been established. For some modems, this is different from the standard code.

Byte 26: Dial code (on hook). This byte contains the value used to configure the modem-control register for an on-hook state during a pulse-dialing operation.

Byte 27: Dial code (off hook). This byte contains the value used to configure the modem-control register for an off-hook state during a pulse-dialing operation.

Bytes 28 through Section 3, character 155: Automatic log-on message. These 640 bytes contain the 10 MacroKeys that you set from the command menu. The 64th character of each MacroKey is reserved for the carriage return. You can edit the MacroKeys here.

Section 3

Bytes 156 through Section 5, Byte 63: Phone numbers. These 420 bytes are used for the dialing-menu table. Each entry is stored name first, followed by the number (exactly as it is displayed). All data are in ASCII.

Section 5

Bytes 64 to 255: Bell figure. These 192 bytes are used to store the user-defined figure displayed on the CRT when an ASCII BEL character is received. The 192 bytes represent a 24-character by 8-line figure.

Section 6

Bytes 0 through 255: KT (keyboard translation) table. Each byte (00 - FFH) indicates a translation value. For example, to manually set the keyboard table to translate a 1FH into a 00, locate byte 1F in this section and change it to a 00.

Section 7

Bytes 0 through 255: Display translation table. Each byte (00 - FFH) indicates a translation value.

Section 8

Bytes 0 through 255: Printer translation table. Each byte (00 - FFH) indicates a translation value.

Section 9

Bytes 0 through 255: BI (buffer-input) translation table.

Section 10

Bytes 0 through 255: BO (buffer-output) translation table.

Section 11

Bytes 0 through 255: RI (RS-232 input) translation table.

Section 12

Bytes 0 through 255: RO (RS-232 output) translation table.

Extended Keyboard Codes

MTERM accepts extended keyboard codes but only recognizes them with the high bits set. This moves them into the non-ASCII range.

Currently, **CTRL** plus the numeric keypad's numbers 1 through 0 are the MacroKeys; **STOP** transmits a true break; and **MENU** toggles back and forth between terminal and command modes.

STOP and **UNDO** terminate most sub-options.

Special Control Codes

MTERM responds to certain control codes that affect cursor movement or in some other way cause MTERM to take a specific action. These actions fall into two categories: escape commands and single-code commands.

With the exception of the form feed (ASCII 12) code, none of these functions may be filtered out or changed. They will always be in effect.

Escape Commands

Sending 26 produces the clear-screen command, also known as **CTRL/Z** . The screen clears and the cursor moves to the home position in the upper-left corner of the screen.

Sending 27 and 61 starts a cursor-control sequence. When the escape code (ASCII 27) is followed immediately by the equal sign (ASCII 61), the cursor moves directly to any specified spot on the screen. Once this command is received, the next two characters received will specify, respectively, the Y (vertical position) and X (horizontal position) coordinates for the cursor. They should be specified in that order.

MTERM takes the ASCII value of these characters, subtracts 32 from them, and moves the cursor to the indicated location. Note that if either coordinate specifies an improper location, the cursor will move to the last valid position in the specified direction and stop there.

Single-code commands

Sending 07 causes the beeper to sound. When MTERM receives an ASCII 07 (CTRL/G), it sounds the internal beeper and prints a 192-character user-definable bell character on the center of the screen. By default, this appears as a large **CTRL/G**.

Sending 08 causes a backspace. When MTERM receives a **CTRL/H**, it backspaces the cursor one position on the screen. The character backspaced over is deleted. Backspace does not cause the cursor to move up to the previous line if it is received as the first character of a line.

Sending 09 produces a tab. When MTERM receives **CTRL/I**, it advances the cursor to the next horizontal tab position on the screen.

Sending 10 produces a line feed. When MTERM receives **CTRL/J**, it moves the cursor down the screen one line. It does not cause the cursor to move back to the beginning of the line.

Sending 13 causes a carriage return. When MTERM receives **CTRL/M**, it moves the cursor down the screen one line and homes the cursor to the beginning of the new line.

Sending 17 turns on the transmit (XMIT) feature. When MTERM has paused from **CTRL/S** and receives **CTRL/Q**, it begins transmitting again. This command is effective only when used in conjunction with **CTRL/S**. It does not cause transmission to start remotely unless an XOFF command is in effect. **CTRL/Q** is also known as DC1 (device control 1.)

Sending 18 serves as an open-buffer command. When MTERM receives a **CTRL/R**, it opens its input buffer. This command is used by most communication services to open a remote buffer before downloading a file. If the service does send this code prior to the download, it will not be necesary for you to do it first.

If you return to the command menu after MTERM receives this command, you will notice that the status display for the buffer switch shows as Open. You can close it by using the B command.

When MTERM receives this code, it displays the message ***Buffer open***. Be careful not to press anything except **MENU** on the keyboard when downloading a file inasmuch as the resulting extra characters will also be stored in the buffer. Pressing **MENU** will send you to the command menu and terminate downloading. **CTRL/R** is also known as DC2 (device control 2).

Sending 19 turns the transmit feature off (XOFF). When MTERM receives a **CTRL/S**, whether from your QX-10 keyboard or a remote computer, it pauses in its output and waits for a signal. This code also pauses other systems when they are transmitting. **CTRL/S** is also known as DC3 (device control 3).

Sending 20 serves as a close-buffer command. When MTERM receives a **CTRL/T**, it closes its buffer. As with **CTRL/R**, most communication services send this command automatically when downloading is complete, so normally you do not have to close the buffer by using the B command.

If you return to the command menu after this command is received, you will notice that the status display for the buffer switch shows as closed. You can open it by using the B command.

When MTERM receives this code, it displays the message *** Buffer closed ***. You can then save the buffer to a disk file at your leisure.

CTRL/T is also known as DC4 (device control 4).

Translation Tables

The translation tables provide information that MTERM needs to communicate with special-purpose systems. MTERM includes the following translation tables:

Keyboard (KI)
Display (DO)
Printer (PR)
Buffer input (BI)
Buffer output (BO)
RS-232C input (RI)
RS-232C output (RO)

These translation tables allow you to:

- Alter any key on the keyboard to generate any value you wish when pressed.

- Change any code being sent to the display to any other code you wish. This can be used as an effective video filter to prevent unwanted codes from destroying your video display.

- Change any code being output to the printer into any other code you wish. This can be used as a highly effective printer filter, getting rid of codes that would otherwise cause your printer to behave erratically.

Using the translation tables, you can decode special characters that mean something to other systems into the corresponding codes for your system. This allows you complete flexibility to translate between ASCII and any other code.

The QX-10 uses the ASCII character set. The ASCII code is the most widely used code in the world, but it is not the only one. For example, other computers use the EBCDIC code. EBCDIC stands for Extended Binary Coded Decimal Interchange Code.

Because of the differences in these codes, if you wanted to communicate with a computer that used EBCDIC, you would need to translate everything coming into the RS-232C port from EBCDIC to ASCII. This could be done via the RS-232C input

translation table (RI). You would also need to translate everything going out of the RS-232C port from ASCII to EBCDIC. This would be done via the RS-232C output translation table (RO).

The vast majority of your communication will be done with systems that use ASCII, so it is highly unlikely that you will ever have to translate an entire code. You will be using the tables more to adjust for special hardware than anything else.

Undoubtedly the most common use will be to configure your output for the particular printer you use. The only thing that is standard about printers is that each one uses control codes—and that any printer may use these codes in a different manner from the others. For this reason, certain control codes that are sent to your terminal to manipulate the video display could cause problems on your printer.

For example, a control code that is displayed on your screen as the tilde (7EH) might be interpreted by your printer as an instruction to begin underlining or use boldface. If your printer is not capable of producing underlining or boldface, almost anything could happen.

To avoid such problems, you can block these codes or translate them into something useful by means of the Printer translation table (PR). For example, whenever the display screen is cleared, you could issue a top-of-form instruction to the printer.

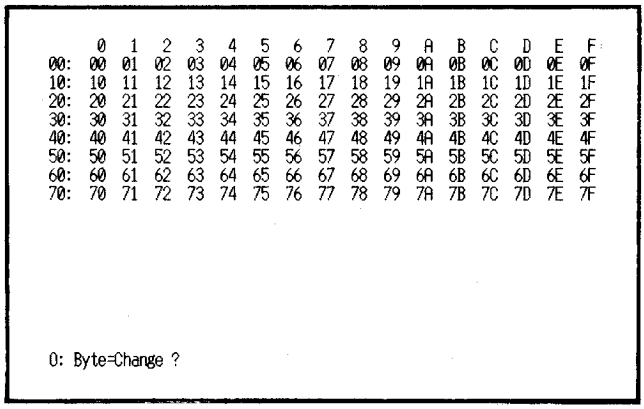
Change/examine tables

You modify the translation tables by using the Change/examine tables feature (command O) from the MTERM command menu. When you select command O, the prompt:

```
O: Which table (TE,KI,DO,PR,BI,BO,RI,RO) ?
```

displays. The TE (telephone) table is described in Chapter 3.

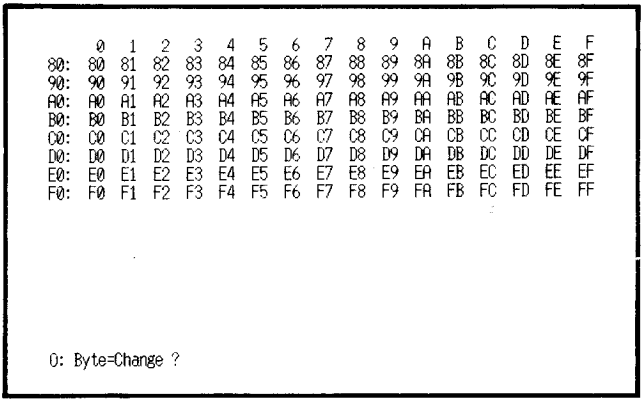
The translation tables all look the same. When you select the table you wish to view, the screen clears and displays:



This first screen shows the first 128 bytes of the table. Notice that there are 8 rows (00 - 70H) of 15 columns (0 - F) each. This allows you to alter bytes 00 - 7F in this first portion.

To find the current value of any byte, first find the row it is in, then look over to the proper column. For example, byte 1EH is in the row beginning with 10 and is underneath the letter E.

To see the next 128 bytes, press **RETURN**. MTERM displays:



This second screen shows the last 128 bytes of the table. Notice that you still have 8 rows (now 80 - F0) with 15 columns (still 0 - F). You look up bytes in the same manner. Pressing **RETURN** switches you back and forth between the first and second portions of the table.

In these examples, all bytes are set to their original values; none of them are translated. If this were the Keyboard table, all keys would send their standard values. If it were the Display or Printer table, all values woudl be transmitted to their respective devices without alteration.

To change a byte, display the portion of the table (first 128 bytes or last 128 bytes) where it is displayed. Then, at the prompt:

```
O: Byte=Change ?
```

type in first the byte you wish to alter, then an equal sign (=), followed by the new value you wish. Then press **RETURN**. The value will change on the screen.

To make it easy for you to tell at a glance which values have been altered, any byte that has been translated into another value is displayed on the table with the ¦ character on each side (for example, ¦**1E**¦). When the byte is restored to its original value, the display returns to normal.

For example, an Epson FX-80 printer is set to expanded mode (double-width print) when it receives an ASCII 0EH (14 decimal). Because this is normally undesirable when doing communications work, it can be prevented by changing the 0EH value to 00.

To change this value, select the Change/examine tables feature from the MTERM menu by pressing **O**. Then press **P** to select the Printer table. Respond to the Byte=Change prompt as follows:

```
O: Byte=Change ? 0E= ¦00¦ RETURN
```

You can see the immediate effect on the table itself; the byte displayed at location 0EH is displayed with the ! character and the value of the byte is changed to 00. (The display is !00!.) Whenever MTERM sends a 0EH to the printer, this table intercepts that value and sends a 00 (null character) instead. This filters out the undesired code.

Another example of changing a translation table involves your QX-10 display screen. Some systems will send codes to clear your screen other than the code recognized by the QX-10.

For example, a system might send a combination of two codes, an ASCII 28 and an ASCII 31. These codes would mean nothing to your QX-10. However, by using the translation tables to change all ASCII 28s into ASCII 26s, you can solve the problem.

To change this value, select the Change/examine tables feature by pressing **O**. Then press **D** to select the Display table. Respond to the Byte=Change prompt as follows:

```
O: Byte=Change ? 1C = 1A RETURN
```

This change filters the ASCII 1CH (28 decimal), causing MTERM to clear your screen every time it receives this character.

Keyboard translation table

The Keyboard translation table (KI) allows you to redefine any keyboard-generated value into any other value your computer is capable of sending. Simply indicate the current value of the key and what you wish this to become. Then, whenever you press that key, the new value will be sent.

Display translation table

The Display table (DO) allows you to translate any display output to match your system's functions. You can use this feature when control codes sent from the host computer do not conform to the proper sequence for your system. These codes are sometimes even assigned values outside the ASCII character set. An example might be the codes used to clear the screen.

This table can also be used as an advance filter to remove any display control codes that would have an undesirable effect on your system.

Printer translation table

The Printer translation table (PR) provides you with two main features. First, the Printer table allows you to use a printer that was configured for some code other than ASCII. For example, a printer that was designed to work with another computer might use EBCDIC. MTERM allows you to translate all of your ASCII output to the corresponding EBCDIC codes.

The second feature is more commonly used. This allows you to custom configure MTERM for your particular printer. For example, you can filter out codes that will cause double-width print, or codes that will cause boldface or underlined print, or any other control codes that might be sent by chance to the printer.

If the service you are using sends you a special character and your printer uses that character to perform a totally different function than the display, you can use the Printer translation table to change the value of the character.

For example, CompuServe might send a left bracket, which is ASCII code 5BH. If your printer uses the code 8AH for the left bracket, simply go to the Printer translation table and change the code at position 5BH to an 8AH. This would translate the CompuServe code 5BH into the 8AH required by your printer to produce a left bracket.

Buffer input (BI), buffer output (BO), RS-232C input(RI), and RS-232C output (RO) translation tables

These four tables perform the same functions except that the BI and BO tables control the MTERM buffer and the RI and RO tables control the RS-232C port. Also, BI and RI function on the receive side as data is input to your QX-10, and BO and RO function on the transmit side as data is output from your QX-10.

When MTERM loads data into the buffer, whether the data is loaded from disk or received from a remote computer, it checks the Buffer input table for any translations. When data is removed from the buffer, whether the data is stored on disk or transmitted to another terminal, MTERM checks the Buffer output table for any translations.

When MTERM receives data through the RS-232C port, it checks the RS-232C input table for translations. And when data is output through the RS-232C port, MTERM performs any translations specified in the RS-232C output table.

Any codes that are modified in these tables are translated as soon as they are received or transmitted. For example, when a code is sent from the host and received by MTERM, it is stored in the MTERM buffer. At this point, MTERM checks to see if the code is modified in the Buffer input table. If it is not, then MTERM checks to see if it needs to be translated for the display or the printer. Thus translations made in the BI, BO, RI, and RO tables are made before those in the Display and Printer tables.

The advantage of using the BI, BO, RI, and RO translation tables as opposed to the Display and Keyboard tables is this: if you have a code that is being sent as a prompt to transmit the buffer and you cannot type this code from the keyboard, you can still generate the code by translating some other code that is generated on the keyboard.

Glossary of Communication Terms

A

ACK

See Acknowledgement.

Acknowledgement

A signal sent from one system to another acknowledging that transmission can begin. See also ENQ, Handshaking, NAK.

Acoustic coupler

A form of low-speed modem that sends and receives data using a conventional telephone handset. This handset does not require a permanent connectin to the line. It is frequently used with portable terminals.

Answer mode

The condition of being ready to recive incoming data.

ASCII

The American Standard Code for Information Interchange (pronounced *ask-ee*), a standard adopted by the American National Standards Institute. This is a 7-bit code, plus parity. The set contains 128 unique character assignments.

Asynchronous

Lacking regular time relationship. As applied to computer communications, asynchronous refers to the method of transmitting data. It is not necessary for a steady stream of data to be transmitted at regular time intervals between each group of bits.

Auto answer

A feature that provides for the automatic answering of an incoming call.

Auto linefeed

A linefeed that is automatically appended to a carriage return, placing the cursor at the beginning of the next line. This prevents the overtyping of existing characters. CP/M-80 uses automatic linefeed.

B

Backup

An identical copy of a diskette, made as insurance against loss of important data.

Baud or baud rate

Transmission speed meaured in terms of the number of frequency changes per second. The most common baud rates are 110, 300, and 1200, with some equipment working at 4800 and 9600 bits per second (bps). A typical pattern for data transmission is 1 start bit, 1 stop bit, and 7 data bits to make up a word. Therefore, at 300 baud approximately 33 characters are transmitted per second.

BBS

Bulletin Board System. A program operating on a host computer, providing a base for Electronic Mail and public information exchange.

Binary

A numbering system based on twos instead of tens. This system uses only the digits zero and one.

Bit

A binary digit, the smallest piece of information recognized by a computer. The two digits are zero and one, which represent states of OFF and ON, or LOW and HIGH.

Block

A group of characters or bytes transmitted as a unit.

Boot

To get a program ready to run from a cold start of the machine. Startup.

Bps

Bits per second. The baud rate is calculated in the number of bits a system transmits each second. Thus, 1200 baud usually indicates 1200 bits are being transmitted each second.

Buffer

A temporary storage area in memory to hold data, usually for input or output. The MTERM buffer can be loaded from a disk file, then routed to the screen or the printer. The data in the buffer can also be transmitted to another computer.

Byte

A 7-bit character.

C

Carriage return or CR

A signal that sends either the cursor back to the beginning of the line or the print head to the beginning of the line it is printing. In CP/M-80, the signal is coupled with a linefeed (LF) to prevent overtyping the existing text.

Carrier

A signal composed of mixed frequency and amplitude. The modem modulates the tones according to signals sent from the computer to the telephone, and it demodulates incoming signals from the telephone to the computer.

Channel

A path for transmission between two or more points. It is also referred to as a circuit, line, link, or facility.

Character set

All characters that are capable of being transmitted, received,

or displayed on the screen. In addition to the printable characters, MTERM recognizes non-printable control characters.

Connect time

Total time spent on host system, measured from sign-on to sign-off, or log-on to log-off. Timesharing systems calculate charges based on connect time.

Cps

Characters per second.

Code conversion

Changing codes from binary to ASCII hexadecimal or from ASCII to binary.

Control character, control code

A character or code generated by pressing an alphanumeric or function key while holding down **CTRL**. This results in a different code being generated than usually associated with the key. In MTERM, using **CTRL** with one of the numeric keypad keys produces a MacroKey function. For example, you can specify MacroKey 2 by pressing **CTRL** and the numeric keypad **2** simultaneously.

CP/M-80

Control Program for Microprocessors. CP/M-80 is a trademark of Digital Research.

CRT

Cathode ray tube, the video screen used to display the characters being transmitted and received.

Cursor

A visual indicator used on a terminal screen to identify the next entry position.

D

Data bit

A bit that represents part of the actual character being transmitted. Parity, start, and stop bits are added before and after the data bits.

Data communication

The technology of transmitting data between computers, often with one host computer and numerous remote terminals and frequently between two microcomputers. Signals are transferred by hard-wire connections, telephone lines, or satellites.

Data set

This is the logical name for a data file, used to eliminate reference to a physical storage device. It is also another name for a modem.

Default

A value supplied by the system in the absence of operator-supplied values.

Demodualtor

The function of a modem which converts incoming signals composed of mixed frequencies and amplitudes to the original digital signals sent by the other computer.

Duplex

Simultaneous transmission in both directions; synonymous with full duplex. See also Full duplex, Half duplex.

E

EBCDIC

IBM's Extended Binary Coded Decimal Interchange Code (pronounced *Ebsidik*). This character set uses eight-bit codes rather than the seven-bit codes of ASCII. EBCDIC can provide for 256 characters.

Electronic Mail

Messages from one computer to another, stored temporarily as files in a third computer. Electronic mail is rapidly becoming popular as an alternative to postal services.

ENQ

Enquiry. A signal sent to the transmitting computer to indicate that the receiver is ready to receive another sector of data. The ACK (acknowledge) signal is the response sent by the transmitting computer. This series of signals is known as *handshaking* and is a method of preventing one computer from getting ahead of the other.

EOF

End of file. A signal indicating that the file being transmitted has reached the end. The receiving computer then sends the transmitting station an acknowledgment that it has received the end of the file. In CP/M-80, the end-of-file is **CTRL/Z**.

F

File

A group of character codes. Data or text files are ACSII codes representing printable characters, grouped by words, sentences, etc., making them useful for communication between people. Program or binary files are used by computers to execute a set of instructions. All types are also known as data files.

Filename

The identifying series of up to eight characters (with an optional extension of three characters) given to a block of data.

Format

1) The process by which a blank diskette is prepared to receive data. Fomatting sets up the tracks and sectors to be used.

2) The layout of information on the screen. Screen formatting can be altered easily from the MTERM Command menu.

3) The grouping of bits that make up a signal, which is transmitted via a modem. Normal formats are five-, six-, seven-, and eight-bit words.

Full duplex

A communication mode that permits transmission of data in two directions simultaneously over the same line. In full duplex mode, characters are sent to the remote computer, which echos back to the sender what was received; this echo is then displayed on the screen.

H

Half duplex

A communication mode that can carry transmission of data in either direction but limits it to one direction at any given moment. In half duplex, all characters entered at the keyboard are sent directly to the display as well as to the modem.

Handshaking

A system of signals sent and received between two computers, assuring that they are working in harmony. See ACK, ENQ, and NAK.

Hardcopy

Paper printout of a file or transmission. With MTERM, you are able to produce hardcopy of a single message or the entire session.

Hardware

Mechanical, electrical, electronic, or magnetic devices. The physical components of a system, as opposed to the software instructions given to the system. See Software.

Hexadecimal or hex

Base 16 number system. Because there are no single digits greater than nine, hexadecimal uses the letters A through F as symbols for the numbers 10 through 15.

I

Intelligent terminal

A combination of hardware and software that permits sending and receiving data while connected to a remote computer. Capabilities include editing and verification, remote data entry, and user-programmable functions. MTERM and your QX-10 combine to form a highly intelligent terminal.

Interface

A device that provides a means of connecting the computer to some outside device, thus allowing communication between the computer and another device. Examples are the RS-232C interface and the cassette interface.

L

Leased line

A channel or circuit furnished to a subscriber for exclusive use. Also referred to as a private line.

Linefeed or LF

A signal that moves the cursor down one line on the screen or advances the paper one line on a printer. Most systems use **CTRL/J** for this function.

Log on, log off

Making connection with the host computer is known as logging on, and the time at which the host recognizes the remote terminal is known as the log-on time. Terminating connection with the host computer is known as logging off. Often, charges are based on the actual connect time between log-on and log-off.

M

MacroKey

A user-defined key capable of storing up to 64 keystrokes. After being programmed, MacroKeys are invoked by pressing **CTRL** and any of the keys on the numeric keypad.

Menu

A table from which the operator can select choices or options.

Mode

The MTERM program has two separate modes, terminal and command. In terminal mode, the actual communicating is carried on. In command mode various functions are performed, such as loading the buffer or changing baud rates.

Modem

A device that converts between the binary digital codes of business machines and the analog signals that can be transmitted over telephone lines. Modem is a contraction of the terms modulator and demodulator. Modems are also known as data sets, subscriber sets, and line adaptors.

N

NAK

Negative acknowledgement. A signal sent to the remote computer indicating that the transmitting computer is not yet ready to transmit. This is the opposite of ENQ. See also ADK, ENQ, Handshaking.

Null

A dummy signal used to fill time or space.

O

Originate mode

One of two pairs of frequencies used for sending and receiving data communications. In the originate mode, the modem transmits and receives a certain set of frequencies. These frequencies are reversed in the answer mode.

P

Parameter

A limit or boundary that is required by the system. Each paramater has a form to which you must adhere when you enter such information as baud rates, etc.

Parity

A constant state or equal value. For example, character bit patterns are forced into parity (total number of one bits that are odd or even) by adding a one or zero bit as appropriate. Parity checking is one of the oldest checking techniques.

Parity bit

A bit transmitted at the end of a character and used by the receiving computer to assist in determining whether or not an accurate transmission was received.

Peripheral

Any device connected to the computer and dependent on the computer for its operational instructions, such as a printer or a modem.

R

RAM or random access memory

Microprocessor memory chips capable of storing data during processing. RAM contains both read and write capability.

ROM or read only memory

A special memory that can be read from but not written into.

Real time

Processing without delay. The term is used by data processing personnel to describe the processing of transactions as they occur.

Record

Logically complete information group.

RS-232C

A standard for interfacing terminals and modems which specifies the connections carrying data signals, control signals, time duration, voltage levels, and current. This standard makes possible data communications almost anywhere in the world.

S

Sector

A physical section of a disk, containing 128 bits.

Serial interface

The connection which permits data transmission serially (sequentially), carrying one bit at a time, as opposed to a parallel interface, which transmits groups of bits simultaneously over several lines.

Software

Programs, codes, routines, or other information written for use with a computer; the information which instructs the computer to do something. See also Hardware.

Spooling

The capability of storing data in a special buffer to allow the printer to operate at its own speed. Because most printers operate at a slower rate than data are received, this feature allows you to continue working without waiting for the printer to catch up or losing data.

Start bit

The bit transmitted immediately prior to the group of data bits to assist in synchronization and verification of the data. Most systems utilize zeros for start bits.

Stop bit

The bit transmitted immediately after the group of data bits to assist in synchronization and verification of the data.

T

Telex

The general name for an international network of teleprinter subscriber services.

Terminal

Any device connected to a communication line (usually a computer) to send and/or receive signals from a remote computer. A dumb terminal is capable of sending and receiving, but has no programmable capabilities, and no ability to store data. See also Intelligent terminal.

Timesharing

Sharing the processing time of a single computer facility among multiple, simultaneous users.

Top of form

The top edge of a printed page.

Transfer of files

Transmission of text or data files from one computer to another. With MTERM this is easily accomplished by loading the desired file into the buffer, then using the T command.

U

Unattended operation

Transmission and/or reception without an operator.