ATL-008
Technical User's Manual

DECEMBER 1983

This equipment complies with the requirements in Part 15 of FCC Rules for a Class A computing device. Operation of this equipment in a residential area may cause unacceptable interference to radio and TV reception requiring the operator to take whatever steps are necessary to correct the interference.

DOCUMENT TM0283-T1-1 REVISED 2/84 VERSION XB

BEEHIVET.M.

WORLDWIDE HEADQUARTERS
SALT LAKE CITY
UTAH

4910 Amelia Earhart Drive • Box 25008 • Salt Lake City, Utah • Phone (801) 355-6000 • TWX 910-925-5271

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

LIST OF FIGURES

LIST OF TABLES

Trademark Notice

DEC is a trademark of Digital Equipment Corporation. VT100 and VT52 may be considered trademarks of Digital Equipment Corporation.

Motorola is a trademark of Motorola Incorporated.

Beehive and ICM are trademarks of Beehive International.

SECTION 1. INTRODUCTION

1.1 ATL-008 DOCUMENTATION

The complete documentation for the ATL-008 terminal consists of three separate manuals. Order these manuals from the Field Service Department of Beehive International. See the title page for the address.

1.1.1 Technical User's Manual

This, the manual you are currently reading, describes functions and features of the ATL-008 for a technically trained user. It also describes the modes of operation of the terminal and provides information for the terminal operator.

1.1.2 Maintenance Manual

This manual describes problem diagnosis and terminal repair procedures for service personnel. It is used in conjunction with the Illustrated Parts Breakdown (see below).

1.1.3 Illustrated Parts Breakdown

This manual lists parts of the terminal in order of disassembly. It provides stock numbers and other information necessary to order replacement parts.

1.2 SCOPE OF TECHNICAL USER'S MANUAL

The Technical User's Manual is designed primarily to be used by programmers and installers. It is written for the technically trained person. It is not a tutorial for the uninitiated user.

This manual is organized as follows:

Section 1 is this introduction. It describes the documentation available for the ATL-008, provides a summary of the Technical User's Manual, explains the notation used in the manual, and describes the important characteristics of the terminal.

Section 2 is the installation section. Inspection, unpacking, site selection, connection of power and communications cables, and turning the terminal on, are all covered in this section.

Section 3 describes the Terminal Configuration Manager - TCM (TM). TCM is a software program which allows the operator to configure the terminal

---

TM TCM is a trademark of Beehive International.

completely by using keyboard selections. The ATL-008 has no configuration switches. Configuration parameters are stored in electronically erasable ROM (EEROM or EEROM), which allows them to be saved even when terminal power is off for a period of time.

Section 4 is the operation section. Modes of operation, keyboard usage, the display test, and simple troubleshooting procedures are described.

Section 5 describes the softkeys. These keys perform various predefined and user-defined functions.

Programmers should enjoy Section 6. This part of the manual lists the various escape sequences and control codes used to make the terminal perform functions specified by ANSI X3.64.

Section 7 lists the escape sequences and control codes used when the terminal is emulating a DEC(TM) VT100(TM), or VT52(TM).

Section 8 gives information and instructions concerning bootloading. This section is intended for people familar with the ATL-008 and C programming.

Appendix A is a standard ASCII chart.

Appendix B lists the keyboard layouts and code charts used by the ATL-008 when in the various country character sets.

Appendix C shows the characters generated by the ATL-008 character generator.

Appendix D is a TCM memory map. This allows a host to reconfigure the ATL-008.

Appendix E presents a summary of terminal command sequences.

1.3 CODES AND CONVENTIONS

Fairly complex code sequences must be sent to the terminal to perform certain tasks. When several keys must be pressed in a given sequence, the names of the keys will be surrounded by a box (▭) or rectangle to separate them from the surrounding text. For example, to cause text to be displayed blinking, the terminal must receive an ESCape code, followed by a left bracket code [, followed by the ASCII code for the number 5 , followed by the ASCII code for the lowercase letter m . This is presented as follows:

ESC [5 m

The spaces in the box are for clarity only and are not to be entered. If a space is a necessary part of the command sequence, the word "SPACE" will appear in the command string. For example, the following performs a scroll right function.

ESC [SPACE A

TM DEC and VT are trademarks of Digital Equipment Corporation. VT52 and VT100 may be considered trademarks of Digital Equipment Corporation.

Some functions require that multiple keys be pressed simultaneously. For example, the scroll right function mentioned above is usually entered from the keyboard by pressing and holding down the CTRL (Control) key, then pressing the → (Cursor Right) key. This is indicated by underlining the names of these keys like this:

CTRL→

1.4 BRIEF TERMINAL DESCRIPTION

The ATL-008 is an intelligent terminal based upon the Motorola MC68008 microprocessor. The MC68008 is similar to the MC68000, but is designed to work with an 8-bit data bus. The terminal also contains 32K of RAM, expandable to 128K.

COMPATIBILITY -- The ATL-008 terminal is based on the ANSI standard X3.64. It operates in two main modes: ANSI standard and VT100 emulation. The operating mode is selected in TCM. In ANSI mode, the terminal responds to all applicable ANSI commands, plus certain Beehive-defined commands. In VT100 emulation mode, the terminal accepts those commands recognized by a DEC VT100. VT52 operation is available as a submode of VT100 operation.

TCM -- Terminal configuration is accomplished via the Terminal Configuration Manager, TCM. TCM allows the operator to configure the terminal completely by using keyboard selections. The ATL-008 has no configuration switches. Configuration parameters are stored in electronically erasable ROM (EEROM), which allows them to be saved even when terminal power is off for a period of time.

SOFTKEYS -- The eight large keys at the top of the ATL-008 keyboard are called soft function keys. These keys are referred to as softkeys because they perform different functions depending upon the operating mode selected:

- In TCM, the softkeys are used to specify terminal configuration parameters.
- In data entry mode, the softkeys select functions such as character insert and delete, set visual and logical attributes, erase screen areas, and printer operations.
- When used with the SHIFT and/or CONTROL keys, the softkeys work as programmed function keys sending escape sequences to the host.
- The operator can specially configure the softkeys to execute any desired sequence without losing any of the predefined functions of the softkeys.

The softkeys have an empty rectangle in place of an engraved label. The labels for the softkeys are found on the bottom line of the screen. As the function of a softkey changes, so does its label. This allows the operator to use the softkeys without memorizing their functions or referring to a manual.

DISPLAY -- The ATL-008 is designed for ease of use. The screen is a 14-inch (35.56 cm) diagonal measure with a P31 (green) phosphor. It displays crisp, easy-to-read characters, making prolonged viewing less tiresome. Either 80 or 132 columns of data can be displayed. This can be selected by the host computer or from the keyboard through TCM.

The screen displays 27 lines. The first 24 are used for data. Line 25 is a separator line. Lines 26 and 27 are the terminal status line and the softkey label line, respectively.

NONGLARE SCREEN -- The ATL-008 has a nonglare nylon mesh filter over the screen.

STATUS LINE -- There are 27 lines on the display. The first 24 are used for display of data. Line 25 is a separator line which accents the bottom two lines. Line 26 is the status line, which indicates the current state of various terminal parameters. There are two status lines which may be displayed: the normal status line and an ANSI status line. In addition, the terminal selftest report or a host message can be displayed on this line.

SCREEN SAVER -- If no characters are received from the keyboard or either of the ports for a specified length of time, the display is turned off to prevent damage to the CRT. Pressing any key restores the display. While the display is off, the cursor moves slowly across the screen to indicate that the terminal is still powered.

The time the terminal waits for input before screen saver turns it off is selectable in TCM. Screen saver can be turned off in TCM.

CHARACTER SET -- The ATL-008 character generator displays 256 unique symbols using a 9 x 13 cell with a 7 x 9 matrix.

NORMAL/REVERSE DISPLAY -- Either light characters on a dark background or dark characters on a light background can be selected in TCM.

PAGES & WINDOWS -- Up to twelve separate pages of data may be defined in terminal memory. Each page is accessed by a window which may be displayed on the screen. A user may move data within a page or between pages. Page and window size are selected in TCM.

POINTERS -- There are three separate active pointers provided by the ATL-008: 1) the cursor, 2) the main data comm pointer, and 3) the aux data comm pointer. Each specifies page number, column number, and row number. The cursor indicates where the next keyboard function will take place. The main data comm pointer indicates where the next main port I/O function will take place. The aux data comm pointer indicates where the next aux port I/O function will take place.

VIDEO ATTRIBUTES -- Three types of video attributes are available on the ATL-008: page attributes, line attributes, and character attributes. Page attributes, which are effective over the entire page, consist of: 80 or 132 characters per line, dark or light background, half or normal brightness. Line attributes specify an action for one line. They allow the host or operator to specify double-wide or double-high/double-wide characters. Character attributes are specified for each character position. Available character attributes include alternate intensity, blank or security video, blink, reverse video, and underline.

BRIGHTNESS AND CONTRAST -- The screen contrast setting is specified by a TCM selection. There are 16 distinct levels of contrast available. The circuitry is designed so that the minimum setting is still visible and the maximum setting is not bright enough to damage the screen.

EDITING FEATURES -- Basic editing can be supported by the host computer in conversational mode or locally by the terminal in block mode.

GRAPHICS -- In graphics mode, 14 user-selectable symbols may be chosen from the character set and inserted on the screen using the softkeys. The graphics key symbols are defined in TCM. Boxes and similar shapes may be drawn using the arrow keys while holding the shift key. The terminal will automatically insert intersections and corners where two lines meet.

COMMUNICATIONS -- The ATL-008 has a main port and an auxiliary port. The main port is generally used to connect the terminal to a host computer. The terminal may be ordered with an RS232C interface only (one main port connector) or with a choice of either RS232C and RS422 or RS232C and VT100 compatible 20mA current loop (two connectors). Selecting between the two interfaces is done in TCM. The auxiliary port is used to connect a serial printer to the terminal. It has an RS232C interface only. Both ports support communication rates from 50 to 19,200 baud in a variety of protocols.

BOOTLOADING -- A programmer can modify the function of the terminal by writing additional code and loading it into terminal RAM. Such code can be used to provide additional user-defined tests to monitor the keyboard for special input. Bootloaded code must be loaded in Motorola S-record format.

ALARMS -- The keyboard generates four types of tones:

- warble
- ring
- beeps (600 Hz and 1200 Hz)
- click

The warble and ring are alternating 600 Hz and 1200 Hz tones. The warble sounds to indicate error conditions in the operation of the terminal (i.e., illegal keyboard actions). The ring is a simulated telephone ring.

The beep, or bell, can generate 600 and 1200 Hz tones. The 600 Hz tone sounds when the keyboard buffer is full or when the terminal is online and the transmitter buffer is full. The 1200 Hz bell is sounded when:

- The cursor moves through a TCM-designated column.
- The line monitor mode is turned off.
- A received parity error is detected.

The click tone (if enabled through TCM) sounds each time a key is pressed.

The volume level of all tones can be set to either high or low through TCM.

SELFTEST -- When the terminal is powered on, the selftest is run. Either the operator or the host may request the terminal to run the selftest by sending an escape sequence. If initiated from the keyboard, the selftest results are presented to the operator on the screen. If initiated by the host, the selftest results are sent to the host through the main port.

ELECTRICAL REQUIREMENTS -- The ATL-008 is adaptable to both U.S. and international power requirements. It can be set at the factory for either 115 or 230 volts and either 50 or 60 Hz.

PHYSICAL DESCRIPTION -- The display portion of the terminal can tilt and swivel for the operator's convenience. The keyboard meets the DIN specification of being 30mm high at the home row. Its retractable cord makes it usable anywhere within three feet of the front center of the terminal. The terminal weighs 25 pounds (11.34 kg) and the keyboard weighs 4.5 pounds (2.04 kg). It takes up an area on the desk of 14.0 inches (35.56 cm) by 15.7 inches (39.88 cm) The keyboard is 20.5 inches (52.07 cm) wide, 7.5 inches (19.05 cm) deep, and 1.5 inches (3.81 cm) high. The terminal itself is 13.4 inches (34.04 cm) high.

OPTIONS -- The following options are available on the ATL-008:

Power: 115VAC 60Hz or 230VAC 50Hz
With or without pedestal
Keyboards: USASCII, United Kingdom, German, Danish, Norwegian, Swedish, French, Canadian French, French lower case, Spanish, or Finnish
Main Port: RS232C only, RS232C and current loop, RS232C and RS422
RAM: 32K or 128K bytes
EEROM: 512 or 2K bytes

MANUFACTURABILITY AND MAINTAINABILITY -- The ATL-008 has a mean-time-between-failure (MTBF) of approximately 5000 hours and a mean-time-to-repair (MTTR) of 15 minutes at the subassembly level. The terminal is designed for fast and inexpensive assembly and disassembly.

SECTION 2. INSTALLATION

NOTE: Installation of the ATL-008 requires a certain amount of technical knowledge and skill. In particular, the installer must possess certain information about the host computer to which the terminal is being connected. Improper installation will not only result in a terminal which does not function properly, but may also cause equipment damage or personal injury. In addition, installing the terminal improperly may void the warranty. Follow the instructions given in this manual. Beehive International assumes no responsibility for conditions arising from improper installation.

2.1 UNPACKING AND IDENTIFICATION

The first thing you'll want to do is check for any signs of shipping damage. The shipping container has been specially designed to protect its contents and prevent damage under normal shipping conditions. Mishandling should be evident upon inspection of the shipping container. If damage is found during visual inspection, take care not to destroy the evidence. If necessary, document the damage with photographs and contact the transport carrier as soon as possible.

The model number is indicated on the outside of the box. Be sure that this is the same number as the model number on the backpanel of the terminal. Save all boxes and packing materials. You will need them if you ever return the terminal for repairs.

Unpack the terminal, referring to the following instructions and Figure 2-1.

1. Place the cardboard box containing the ATL-008 terminal lengthwise on the floor. The handles in the sides of the box will be up.

2. Open the end of the box with the model number printed on it.

3. Remove the four corner cushions.

4. Slide out the inner cardboard box.

5. Open the front end of this inner box.

6. Remove the plastic bag containing the manual and the strain relief clamp.

7. Remove the smaller rectangular box containing the keyboard.

8. Remove the locating tube from the terminal's bezel.

9. Remove the power cord and the keyboard connecting cord from the side folding pads.

10. Slide out the terminal and remove the anti-static shroud.

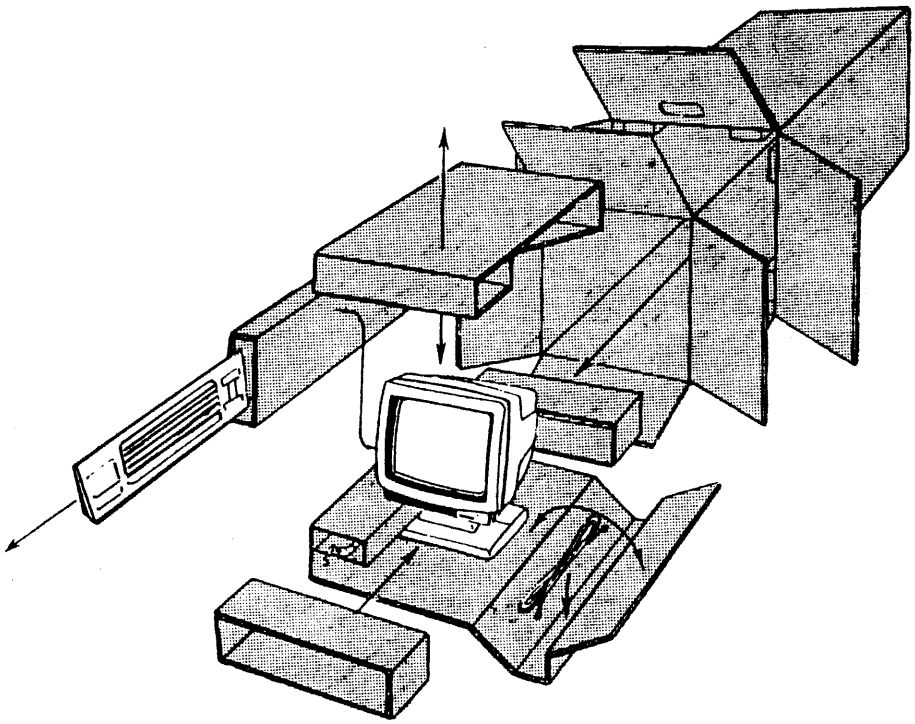


Figure 2-1. Unpacking the ATL-008 Terminal

The model number, part number, serial number, voltage and current requirements, and frequency/power classifications are located on a plate on the back of the terminal. See Figures 2-2 and 2-3. Be sure that the voltage, frequency, and power consumption conform to your application before connecting the ATL-008 to power.

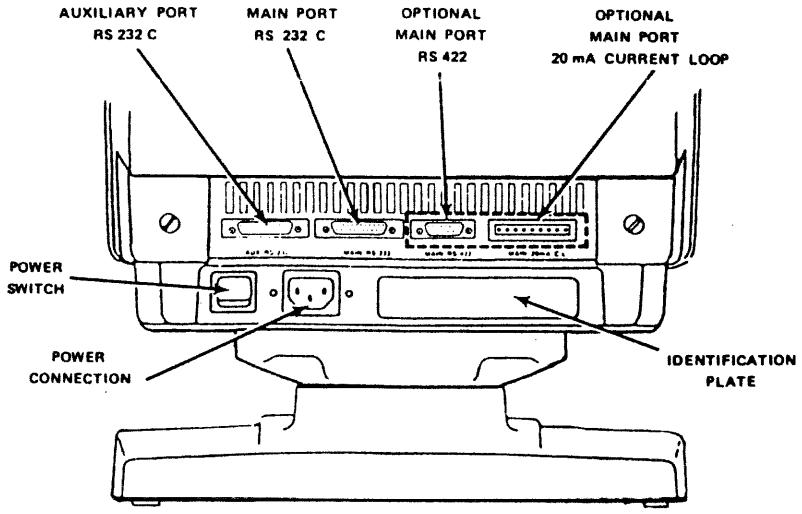


Figure 2-2. ATL-008 Rear Panel

You can make sure you have the correct model with the correct options by looking at the identification number on the plate (see Figure 2-3). Figure 2-4 shows what each character means on the identification number. For example, a terminal with an identification number of 1015-1000-0100 has a USASCII keyboard, a pedestal, a 14" P31 phosphor 22.5 kHz monitor, is configured for 115 volt operation, has RS232C and current loop connectors on the main port, and has 32K of RAM and 512 bytes of EEROM.

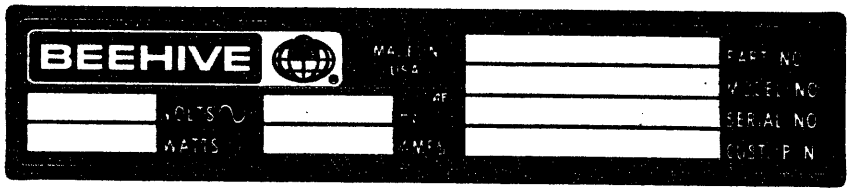


Figure 2-3. Backpanel ID Plate

2.1.1 Air Flow Warning and Keyboard Attachment

After removing the ATL-008 from its container, place it on a level surface where the power and I/O cables are not in your way. Make sure the cables won't be pulled if the terminal is moved. Connect the detachable keyboard to the output jack on the left side of the terminal (See Figure 2-5).

ATL terminals are cooled by convection. This means that air enters the terminal through the bottom, side and rear vents, cools the circuitry, and leaves through the top vents. There is no fan. The ATL-008 requires free airflow. Don't place the ATL-008 on any surface that might block cooling air from the back, sides, bottom, or top of the terminal. Keep at least 3 1/2" clearance all around the terminal. Care must be taken not to cover the vents in the top, back, sides and bottom of the terminal. Even a sheet of paper inadvertently placed on top of the terminal will impede airflow and may result in overheating and failure.

In addition, great care must be taken to avoid spilling liquids into the terminal or keyboard. Common substances such as coffee, soft drinks, cigarette ash and dust can cause problems with proper heat dissipation if they are spilled onto the circuit components. In extreme cases, such substances can cause short circuits resulting in erratic operation or damage to the terminal.

Preventing contamination is much easier than correcting it. Keep food, drink and smoking materials away from the terminal and keyboard. Wipe off any spilled substances with a soft dry cloth.

In areas where airborne dust or other contaminants are unavoidable, it is a good idea to cover the terminal when it is not in use. Be sure to remove the cover each time the terminal is turned on or overheating can occur. Remember there is no fan and the cooling vents must be unobstructed.

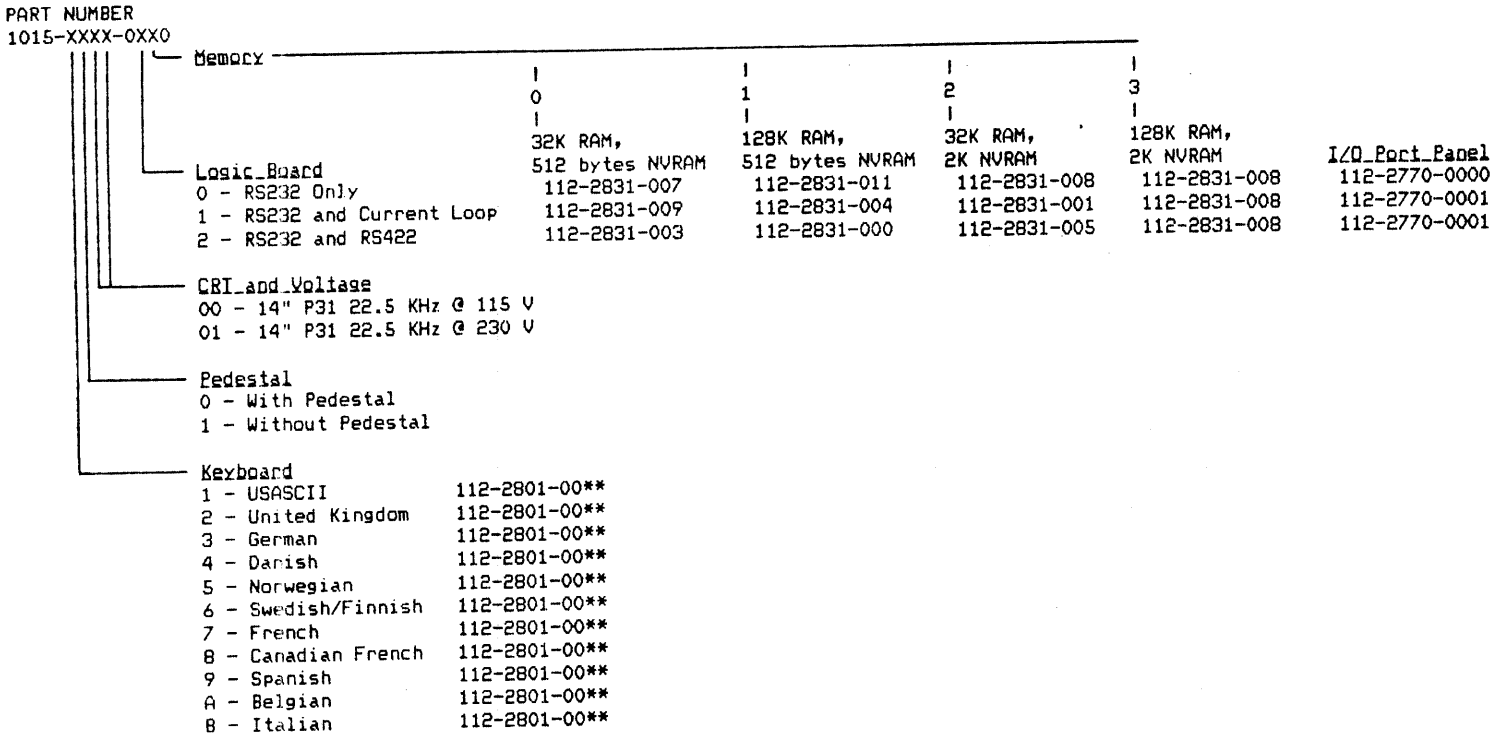


Figure 2-4 ATL-00B Terminal Feature Identification Chart

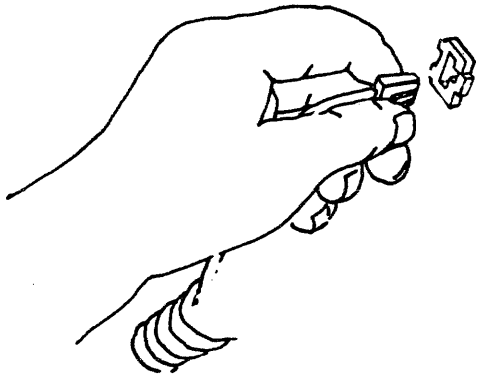


Figure 2-5 Connecting the Keyboard

2.1.2 AC Power Connection

Before connecting the ATL-008 to to power, verify that the line voltage matches the specification on the identification plate on the rear of the terminal. Whenever you turn on the ATL-008, make sure the power outlet is properly grounded and supplies the correct operating voltage and frequency.

The ATL-008 is shipped with a three-pronged power cord. If you use an adaptor, ground the pigtail. DO NOT REMOVE the grounding pin on the AC plug since a good ground is essential for proper operation of data processing equipment. Not having your terminal properly grounded may void the warranty.

If it becomes necessary to move the terminal, disconnect it from the power source.

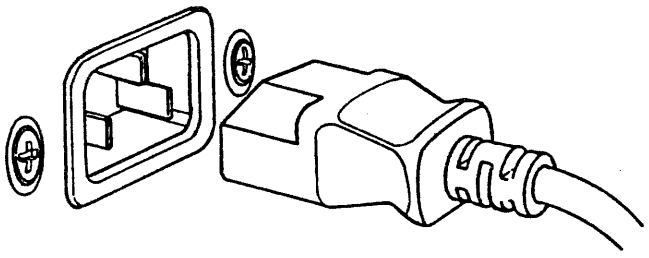


Figure 2-6 Connecting the Power Cord

To prevent the power cord from being accidentally pulled from the power plug, attach the strain relief clamp as shown in Figure 2-7. Do this with the power cord disconnected from the AC line.

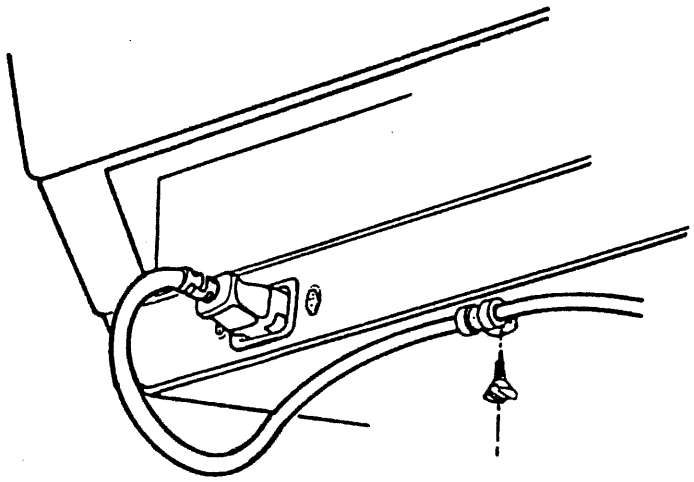


Figure 2-7 Attaching the Strain Relief Clamp

2.1.3 Terminal Dimensions

Monitor Enclosure (including Tilt/Swivel Pedestal)

Height	13.4 in (34.04 cm)
Width	14.0 in (35.56 cm)
Depth	15.7 in (39.88 cm)

Keyboard Enclosure

Height	1.5 in (3.81 cm)
Width	20.5 in (52.07 cm)
Depth	7.5 in (19.05 cm)

Desk Top Depth (footprint) 23.2 in (58.93 cm)

Tilt

Forward Tilt	5°
Backward Tilt	15°
Total Range	20°

Swivel 360°

Weight

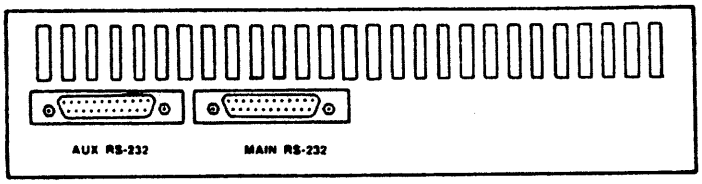
Monitor Enclosure	25.0 lb (11.34 kg)
Keyboard Enclosure	4.5 lb (2.04 kg)

2.2 INTERFACING
(Connecting Your Terminal to Other Devices)

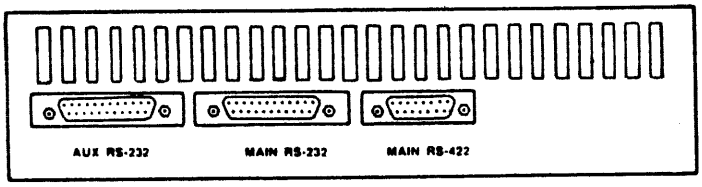
The ATL-008 has two input/output ports, called the main and the auxiliary ports. The auxiliary port has a single connector and allows EIA RS232C electrical interface. The main port has one or (optionally) two connectors and allows RS232C and (optionally) RS422 or 20mA current loop. The connectors are located at the rear of the terminal (see Figure 2-8). Although the main port may allow two different interfaces, only one interface can be selected at any one time. All outputs on both connectors are active at all times. Only the inputs from the selected connector are active at any one time. Switching between the optional interfaces is accomplished through TCM (see the paragraph called Main Port Hardware Interface in Section 3).

The connectors used for the main port include a 25-pin D-subminiature connector (RS232C), a 15-pin D-subminiature connector (RS422), or an 8-pin AMP MATE-N-LOK connector (20mA current loop) depending on the interface ordered.

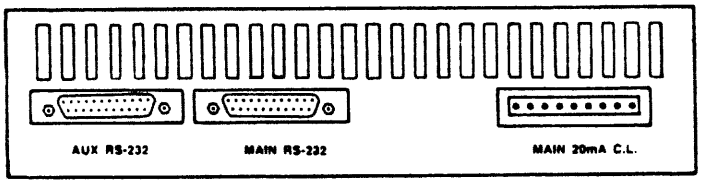
The auxiliary port uses a 25-pin D-subminiature connector.



Main Port RS232C Only



Main Port RS232C and RS422



Main Port RS232C and 20mA current loop

Figure 2-8 ATL-008
Main and Aux Port Connector Configurations

2.2.1 Main Port

The main port allows asynchronous communications. All standard baud rates from 50 to 19,200 baud are software-selectable through TCM. See the paragraph called Main Port Hardware Interface in Section 3.

Parity, stop bits, and auto echo selection are controlled through TCM. See the paragraphs called Main Port Hardware Interface and Main Port Protocol in Section 3. All configuration parameters are stored in non-volatile memory.

Modem Considerations - Thresholds are set so that all modem signals fail-safe off. Auto enables on CTS and DCD can be enabled under TCM. This allows a modem connection or three-wire connection with no false indications.

Error Detection - Async parity errors display a substitute character (DEL, 7FHEX), and actuate the warble tone. Any errors are logged in the status line message area.

RS232C Connector Pinouts - A 25-pin, D-subminiature connector is used for RS232C. Signals and pin assignments are listed in Table 2-1.

Table 2-1
ATL-008 Main Port Pin Assignments
25-Pin RS232C Connector

PIN #	NAME	DIRECTION AT TERMINAL	DEFAULT/ FAIL SAFE STATE
1	AA Protective Ground		
2	BA Transmitted Data	OUT	
3	BB Received Data	IN	OFF
4	CA Request to Send	OUT	
5	CB Clear to Send	IN	OFF
6	CC Data Set Ready	IN	OFF
7	AB Signal Ground		
8	CF Received Line Signal Detector	IN	OFF
9	Reserved		
10	Reserved		
11	NC		
12	CI Speed Mode Indication	IN	OFF
13	NC		
14	NC		
15	DB Transmit Sig. Element Timing	IN	OFF
16	NC		
17	DD Received Sig. Element Timing	IN	OFF
18	CN Analog Loop	OUT	
19	NC		
20	CD Data Terminal Ready	OUT	
21	RL Remote Digital Loop	OUT	
22	CE Ring Indicator	IN	OFF
23	CH Speed Select-Originate		
24	NC		
25	TM Test Mode	IN	OFF

NOTE: Pin 1 may be connected to Pin 7 with a strap.

This interface has the following specifications:

- Output voltage range.....+12 V to -12 V
- Output impedance.....300 ohms minimum
- Input resistance.....4 kilohms
- Driver circuits current....10 mA (sink)
- Signals will accommodate a 212A-type modem.

Main Port RS422 - An additional connector is supplied with the ATL-008 when the RS422 option is ordered (see Figure 2-8b). The RS422 interface has the following features:

- Conforms to the standard used by the Public Data Networks as outlined in ISO 4903-1980 and RS-449 appendix.
- Possible cable lengths of up to 4000 feet (1.2km)
- Baud rates of up to 1 Mbps

A 15-pin, D-subminiature connector is used for RS422. Signals and pin assignments are listed in the following table. Definitions for these signals are given in CCITT Standard X.24.

Table 2-2
ATL-008 Main Port Pin Assignments
15 Pin RS422 Connector

PIN #	NAME	DIRECTION AT TERMINAL	RS232C EQUIVALENT
1	Shield		1 AA Protective Ground
2	T(A') Transmit	OUT	2 BA Transmitted Data
3	C(A') Control	OUT	4 CA Request to Send
4	R(A) Receive	IN	3 BB Received Data
5	I(A) Indication	IN	8 CF Received Line Sig. Det.
6	S(A) Signal Element Timing	IN	15,17 DB, DD Trans & Rec. Signal Element Timing
7	NC		
8	G Signal Ground		7 AB Signal Ground
9	T(B') Transmit	OUT	
10	C(B') Control	OUT	
11	R(B) Receive	IN	
12	I(B) Indication	IN	
13	S(B) Signal Element Timing	IN	
14	NC		
15	Reserved		

NOTE: Pin 1 may be connected to chassis ground or to signal ground through straps, or left unconnected. The default state is chassis ground.

RS422 Cabling - If you provide your own cables for the transmit and receive lines, use 24 AWG copper, twisted pair, shielded cables.

In doing your own cabling, you will have to ensure that there is a path between the terminal and the host. There are two ways to wire this path, as mentioned below. Regardless of the wiring method, the first step is to connect the shielding of the two cables (transmit and receive) at both ends.

Method 1 - When the terminal chassis ground (pin 1 of the main port) and the chassis ground of the host CPU have little or no potential difference, tie both ends of the shielding to their chassis grounds.

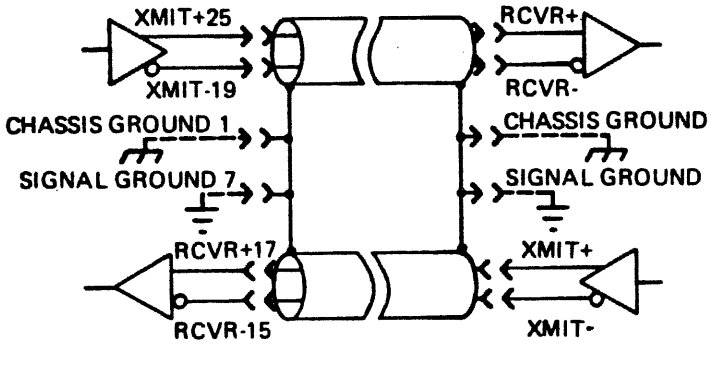


Figure 2-9 Wiring the Terminal/Host Path

Method 2

When the potential difference is greater than 4 V RMS, test the chassis grounds with an oscilloscope to see which is the quieter. The shielding of the quieter end should be connected to its respective chassis ground and the shielding of the noisier end should be connected to its respective signal ground. In the equipment of the noisier end, the signal and chassis grounds must be separated and the circuitry must be able to withstand the resulting potential difference.

Main Port 20 mA Current Loop - an 8-pin AMP MATE-N-LOK connector is used when the 20 mA current loop option is ordered. This connector is compatible with VT-100 and VT-52 type equipment connectors. Signals and pin assignments are listed in Table 2-3.

Table 2-3
ATL-008 Main Port Pin Assignments
8-Pin 20 mA Current Loop Connector

PIN #	NAME
1	Test Negative
2	Transmit -
3	Receive -
4	NC
5	Transmit +
6	NC
7	Receive +
8	Protective Ground

Field Service accessible switches mounted on the P.C. board allow for passive or active current loop on transmit and/or receive. The circuits are compatible with those used by DEC. The current loop converts the TTL logic signals into current signal patterns at one end of the loop, transmits the pattern, reconverts the pattern to TTL logic signals at the other end, and delivers it to the computer or terminal.

This interface has the following features:

- Possible cable lengths up to 2000 ft (609.6m)
- If less than 9600 bps, lengths up to 6000 ft (1.8km)

Common interfacing schemes using current loop are shown in Figure 2-10.

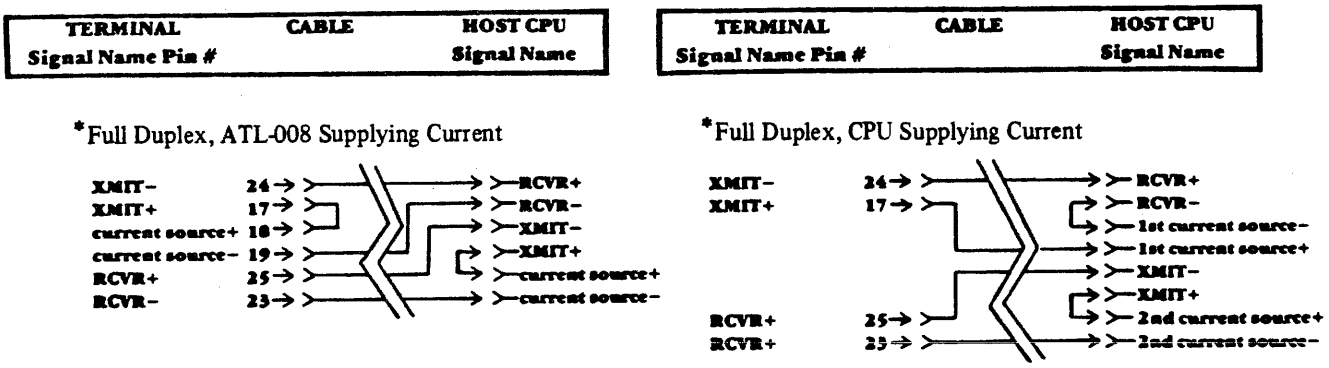


Figure 2-10 Current Loop Cabling

2.2.2 Auxiliary Port

The auxiliary port uses a 25-pin female connector for use mainly as a printer port and is RS232-compatible (see Figure 2-8). The aux port is also capable of asynchronous and isochronous transmission. Ten hardware flow control techniques are available:

Pin 11/19 = Busy	ETX/ACK
Pin 20 = Printer Ready	XON/XOFF
Pin 11/19 = Ready	ACK/NAK
Pin 4 = Request to Send	TTY
Pin 11/19 = Reverse Channel (Centronics)	None

All auxiliary port electrical interface components are commonly available RS232C standard components. Pins 11 and 19 are tied together in the interface and use the same RS232C receiver. Pins 15 and 17 come from the same clock but separate RS232C drivers are used for each output.

The aux port is capable of asynchronous and isochronous transmission. Additional flow control techniques, all standard baud rates, as well as other standard options (all saved in non-volatile memory) are available through TCM (see the paragraphs in Section 3 titled Aux Port Hardware Interface and Aux Port Protocol).

Connector Pinouts - All circuits on the 25-pin aux connector are RS232C-compatible. RS422/423 and current loop are not supported. Signals and pin assignments are listed in the Table 2-4

Table 2-4
ATL-008 Auxiliary Port Pin Assignments
25 Pin RS232C Connector

PIN #	NAME	DIRECTION AT TERMINAL	DEFAULT/ FAIL-SAFE STATE
1	Protective Ground		
2	AX TXD	IN	OFF
3	AX RXD	OUT	
4	AX RTS	IN	OFF
5	AX CTS	OUT	
6	AX DSR	OUT	
7	Signal Ground		
8	AX DCD	OUT	
9, 10	NC		
11	AX DEV RDY	IN	OFF
12 - 14	NC		
15	Signal Element Timing	OUT	
16	NC		
17	Signal Element Timing	OUT	
18	NC		
19	AX DEV RDY	IN	OFF
20	AX DTR	IN	OFF
21 -25	NC		

2.3 TERMINAL CONFIGURATION MANAGER (TCM)

The Terminal Configuration Manager allows the keyboard (or a host computer) to control brightness, audio alarm volume, installation parameters, and other parameters. TCM consists of menus used to select the values for these parameters. As in all menu-driven systems, some menus may lead to additional menus.

Default TCM parameter settings defined by BEEHIVE are stored in ROM. As these default settings are changed by the user, a copy, with the changes, is stored in EEROM to preserve the changes when the terminal is powered off. Each time the terminal is turned on, TCM is copied from EEROM into RAM.

TCM is divided into two parts. The larger part (and less frequently used) is called Installation TCM. It is usually used only when major changes are needed in the terminal configuration and may be protected with a password. The smaller part (and more frequently used) is called Operator TCM. It controls brightness, alarm volume, video parameters, keyboard parameters, and graphic key definitions. Operator TCM may be used while in Installation TCM, but Installation TCM may not be used from Operator TCM.

TCM is accessed by the CONFIGURE key. The CONFIGURE key has three uses:

1. Press it to display menus for Operator TCM menu, brightness, and alarm volume. These menus are enclosed in boxes.
2. Press it with the SHIFT key (SHIFT CONFIGURE) to display menus for Installation TCM, Operator TCM, brightness, and alarm volume. If Installation TCM has been password-protected, the ATL-008 will display an appropriate prompt indicating that a password must be typed. Entry of an incorrect password aborts TCM; the correct password results in the Installation TCM menu.
3. When pressed simultaneously with the CTRL key, a test pattern is displayed. This is why the legend on the front of the CONFIGURE key is TEST. This test pattern consists of all available characters and video attributes.

These screens will be discussed in detail in Section 3.

2.4 TURNING ON THE TERMINAL

Before the terminal is used, it must be properly installed and configured in accordance with Section 2 by qualified personnel.

Set the rear panel POWER ON/OFF switch to ON. (See Figure 2-11.) After a short warm-up period, the cursor and status line will appear on the screen. The cursor appears at the upper left-hand corner of the screen. The status line appears in the 26th line of the display, in normal intensity. The soft key descriptions appear in the 27th line, as shown in Figure 2-12 or Figure 2-13.

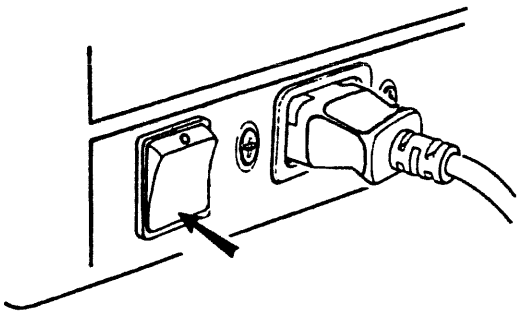


Figure 2-11. ATL-008 Power Switch

2.4.1 Terminal Selftest

On powering up, the terminal performs a display memory test and a terminal operation test. When the selftest is completed, if no errors are detected, the screen should resemble Figure 2-12.

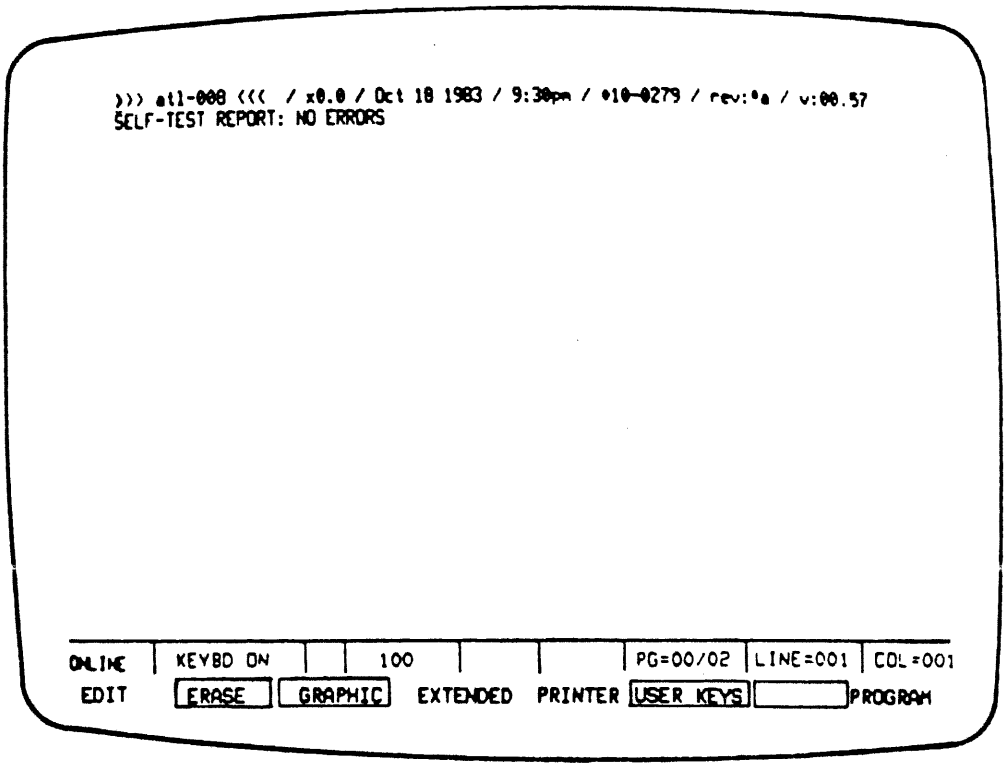


Figure 2-12
Selftest Report (No Errors)

In the event of a test failure, a message describing the error is displayed, as in Figure 2-13.

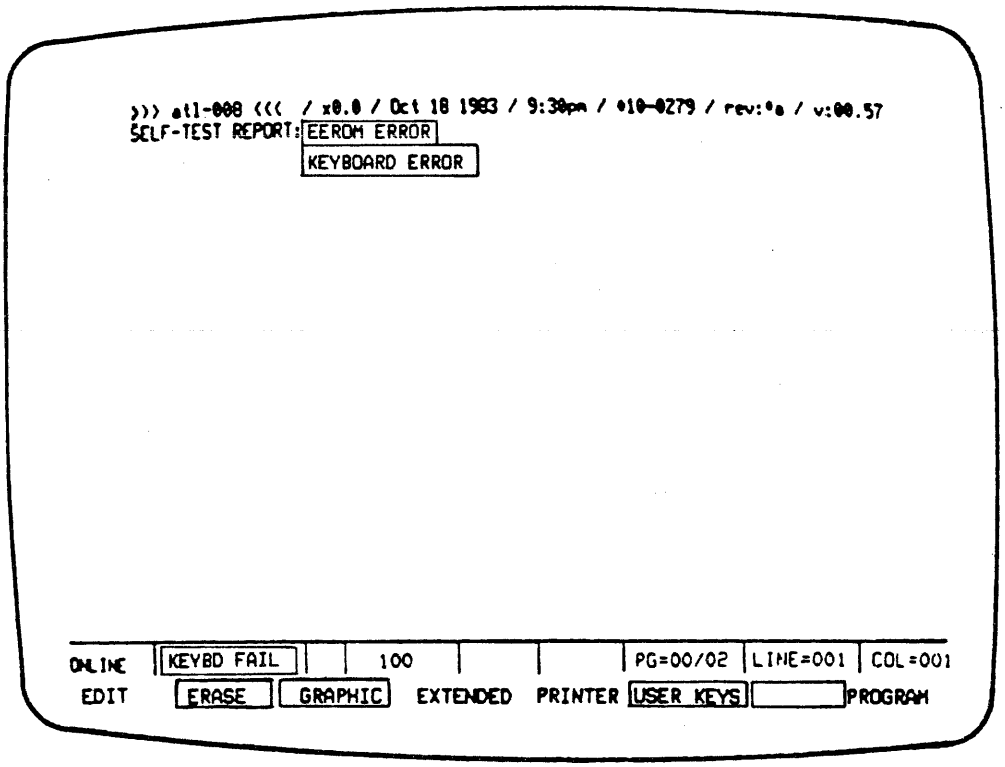


Figure 2-13
Selftest Report (With Errors)

NOTE: Selftest may also be initiated by the host computer or by the operator using the RIS or DCS escape sequences (see Sections 6 and 7).

2.5. DISPLAY TEST

When the CONFIGURE key is pressed simultaneously with the CTRL key, a test pattern is displayed. This is why the legend on the front of the CONFIGURE key is TEST. The next keystroke re-displays the normal screen. The test pattern consists of all available characters and video attributes (Figure 2-14).

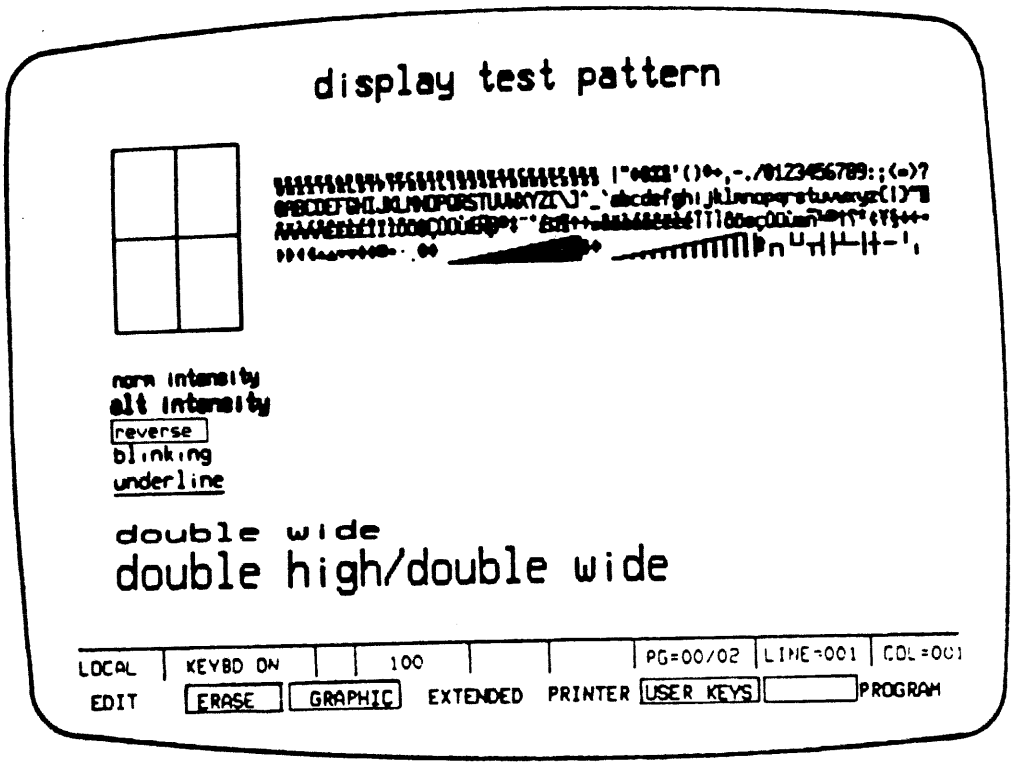


Figure 2-14
Display Test

SECTION 3. TERMINAL CONFIGURATION MANAGER (TCM)

The Terminal Configuration Manager allows the keyboard (or a host computer) to control brightness, audio alarm volume, installation parameters, and other parameters. TCM consists of menus used to select the values for these parameters. As in all menu-driven systems, some menus may lead to additional menus. TCM menus may also be displayed in a cyclic fashion using the NEXT and PREV MENU softkeys. This section will illustrate these menus with default settings and explain the various selections.

Default TCM parameter settings defined by BEEHIVE are stored in ROM. As these default settings are changed by the user, a copy, with the changes, is stored in EEROM to preserve the changes when the terminal is powered off. Each time the terminal is turned on, TCM is copied from EEROM into RAM. The RAM copy is what is actually used in setting up the terminal.

TCM is divided into two parts. The larger and less frequently used part is called Installation_TCM. It is usually used only when major changes are needed in the terminal configuration and may be protected with a password. The smaller and more frequently used part is called Operator_TCM. It controls brightness, alarm volume, video parameters, keyboard parameters, and graphic key definitions. Operator TCM may be used while in Installation TCM, but Installation TCM may not be used from Operator TCM.

TCM is accessed by the CONFIGURE key. The CONFIGURE key has three uses:

1. Press it to display menus for Operator TCM menu, brightness, and alarm volume. These menus are enclosed in boxes.
2. Press it with the SHIFT key (SHIFT CONFIGURE), to display menus for Installation TCM, Operator TCM, brightness, and alarm volume. If Installation TCM has been password-protected, the ATL-008 will display an appropriate prompt indicating that a password must be typed. Entry of an incorrect password aborts TCM; the correct password results in the Installation TCM menu.
3. Press it simultaneously with the CTRL key, to display a test pattern. This is why the legend on the front of the CONFIGURE key is TEST. This test pattern consists of all available characters and video attributes. Pressing TEST (CTRL CONFIGURE) again will eliminate the display test pattern.

These screens (except the test pattern) will be discussed in detail in this section.

3.1 USING TCM

Several keys are used to control TCM. The cursor right and left keys move the cursor from menu box to menu box while in TCM. The cursor up and down keys move the cursor vertically within each menu box except when in the brightness menu box on the first TCM screen. The softkeys change screens and control other TCM functions.

The large keys on the top row of the keyboard are called softkeys because their functions change depending on the current mode or operation. Each key has three legends indicating the normal, shifted, and control functions of the key. The shifted and control functions are on the top and front of the key respectively. The normal key legend on all of these keys is a rectangle. Each rectangle indicates that the normal key operation is given in a corresponding area on the bottom line of the screen. These areas are the softkey labels. The colors of the keys correspond to the light and dark labels.

TCM menus are displayed with current parameter settings shown in reverse video. TCM parameters are usually selected by positioning the cursor on the desired setting and pressing the IDENTIFY softkey. There are some exceptions to this selection process; they are noted where appropriate.

3.1.1 TCM Softkeys

There are two sets of softkeys used to control TCM. The first set, shown in Figure 3-1, is used to select TCM parameters. The second set, shown in Figure 3-2, is used to store these parameters.

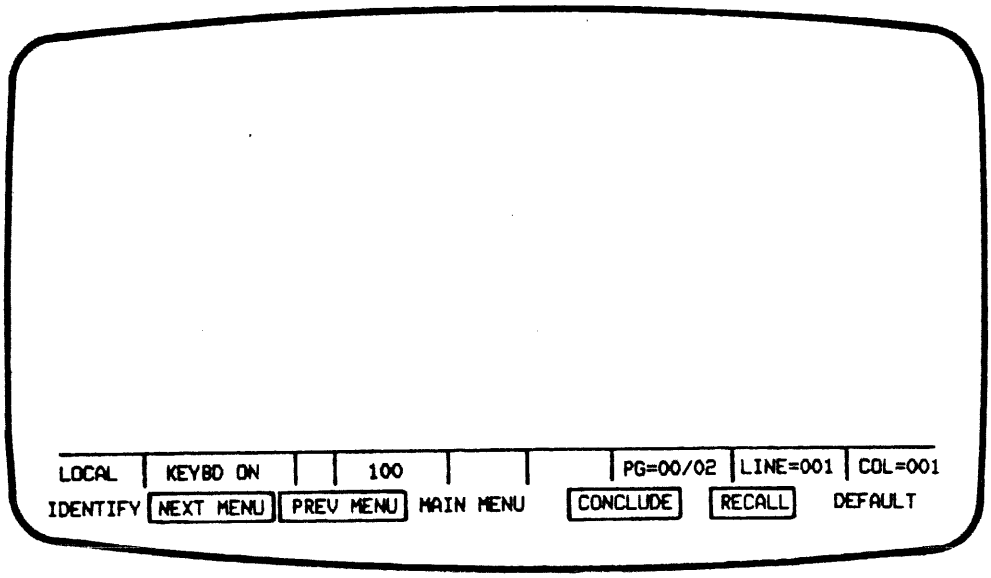
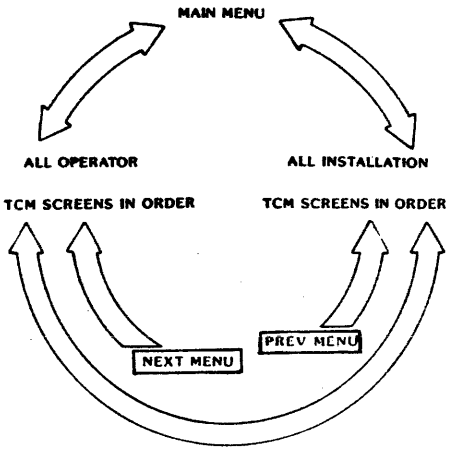


Figure 3-1 First Set of TCM Softkey Labels

IDENTIFY This softkey selects an item from a group of several options; it is not used when alphanumeric input is required. Use the cursor positioning keys to locate the cursor on the desired parameter setting and press this key. The new selection will be highlighted.

NEXT MENU This key displays the next TCM menu screen. If in Installation TCM, pressing this key repeatedly displays the various Installation TCM menu screens in the order they are listed in the main Installation TCM menu, followed by the screens for Operator TCM. If in Operator TCM, pressing it repeatedly displays the various Operator TCM menus in the order they are listed in the main Operator TCM menu.



PREV MENU This key performs a function similar to NEXT MENU, but in reverse. Instead of displaying the next menu, it displays the PREVious MENU screen. Experiment with NEXT and PREV MENU; they also work from the Main menu. Pressing either of them repeatedly will display every available TCM menu.

MAIN MENU This key displays the Main Menu for either type of TCM.

CONCLUDE After specifying the TCM parameters, press this key. It will display the second set of TCM softkeys that determine how TCM will be stored.

RECALL This key is sort of an un-do key. If you change a TCM parameter setting, RECALL will recall the parameter from the TCM EEROM (whatever value was saved the last time TCM was saved with the SAVE TCM softkey). The softkey label on the screen will blink to indicate that the EEROM value has been used. Pressing the key again stops the blinking and restores the previous value.

DEFAULT Like RECALL, this is also an un-do key. It recalls the Beehive default setting from ROM. It also blinks to indicate that the default setting has been used. Pressing the key again stops the blinking and restores the previous value.

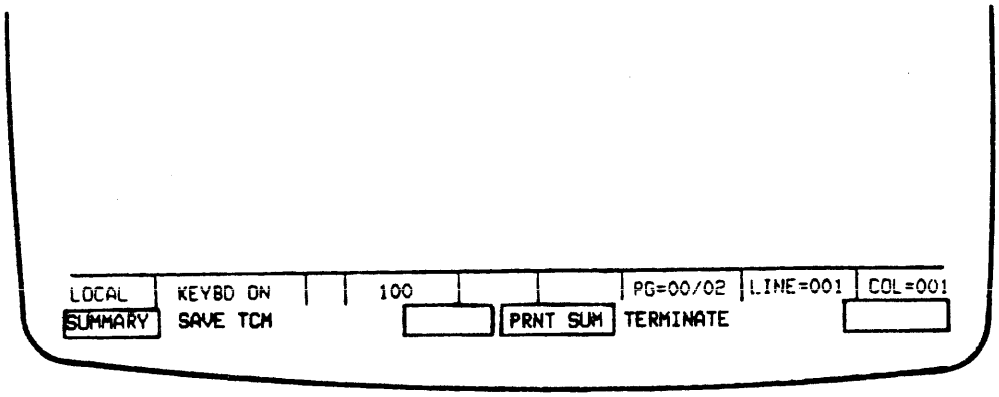


Figure 3-2 Second Set of TCM Softkey Labels
(From CONCLUDE softkey)

SUMMARY This softkey displays a sequence of summary screens showing the TCM parameter settings. A sample summary is illustrated in the section on TCM Summary Reporting. Summaries are provided for both Installation and Operator TCM.

SAVE TCM This softkey saves the TCM settings in EEROM so that they will be available after the terminal is powered off or reset.

PRNT SUM This key prints the summary if there is a printer properly attached to the ATL-008. A sample summary is illustrated in the section on TCM Summary Reporting.

TERMINATE This key ends the TCM modification session without saving any changes in EEROM. Any changes will function only until the terminal is powered off or reset because they are stored in RAM. When the terminal is reset the last SAVEd settings will be reloaded from EEROM.

3.2 INSTALLATION TCM MAIN MENU

Installation TCM provides control over a wide variety of parameters. The Installation TCM Main Menu is accessed by pressing SHIFT CONFIGURE. The Main Menu is shown in Figure 3-3.

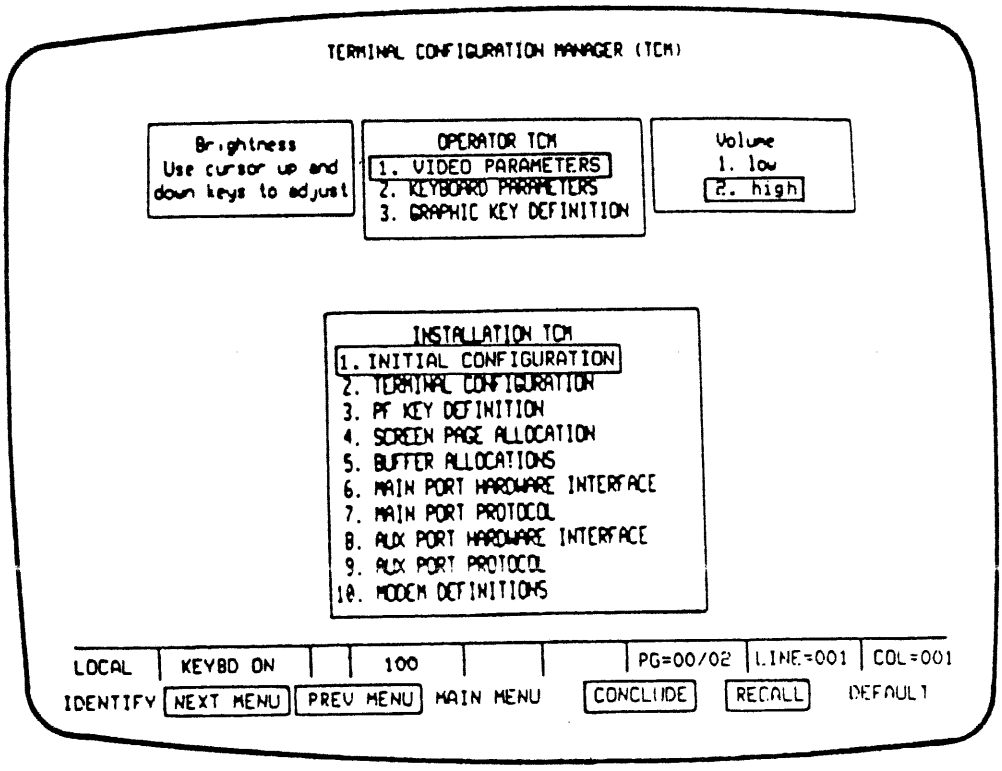


Figure 3-3 Installation TCM Main Menu Screen

This screen controls the following four parameters:

Brightness -- To change the brightness, position the cursor anywhere in the box. Press the cursor up key to increase the brightness, and the cursor down key to decrease the brightness. The audible alarm will sound when the brightest or faintest limit is reached. Normal or Full intensity is determined by the setting of the Display parameter in the Video Parameters menu.

Operator ICM -- This menu leads to three additional menus for control of Video Parameters, Keyboard Parameters, and Graphic Key Definition. This is how Operator TCM may be accessed through Installation TCM. See the section titled Operator ICM Main Menu for details.

Volume -- To select low or high volume level for the audible alarms and the keyclick, use the cursor positioning keys and place the cursor on either the low selection or the high selection. Press the IDENTIFY softkey. The selection you chose will be highlighted (displayed in reverse video). When you exit TCM, the new volume level takes effect.

Installation ICM -- This is a list of the ten menus which comprise Installation TCM. There are two ways to view these menus. The first is to position the cursor on the title of the desired menu and press the IDENTIFY softkey. The menu will be displayed. The second method is to repeatedly press the NEXT or PREV MENU softkey. This will display the Installation TCM menus in turn in the order they are listed in the Installation TCM menu. These menus will be described in the following pages. The screens will be shown with the Beehive defaults and text to explain the various selections.

3.2.1 Installation ICM Password

If Installation TCM has been assigned a password, a screen like Figure 3-4 will result. Type the password and press RETURN; the Installation TCM screen will be displayed.

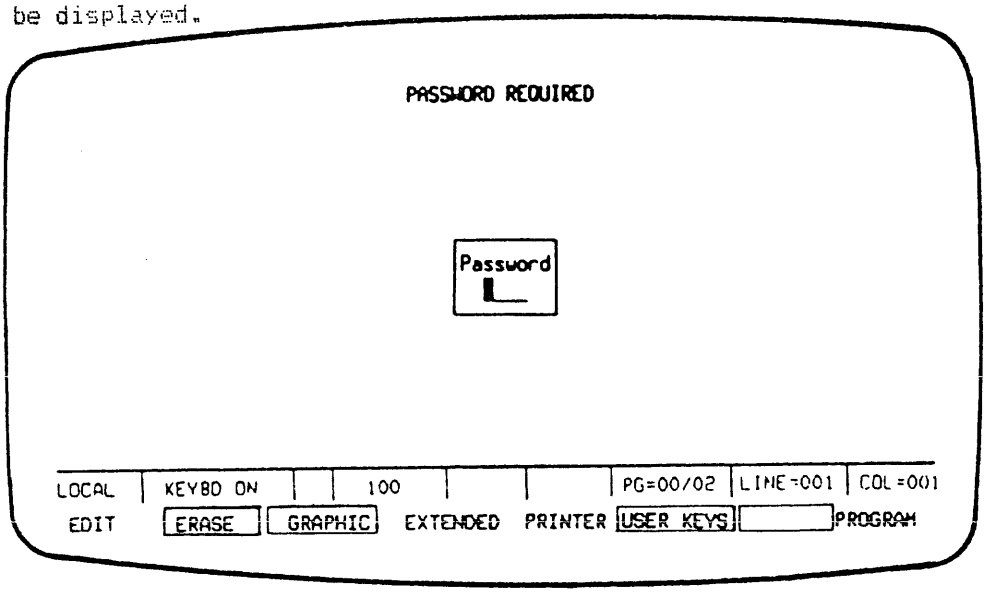


Figure 3-4 Password Screen

3.3 INSTALLATION TCM SUBMENUS

3.3.1 Initial Configuration

The Initial Configuration screen sets the operating Mode, the Installation TCM and Program access, the Line frequency, and the nationality of the character set. The screen is illustrated in Figure 3-5.

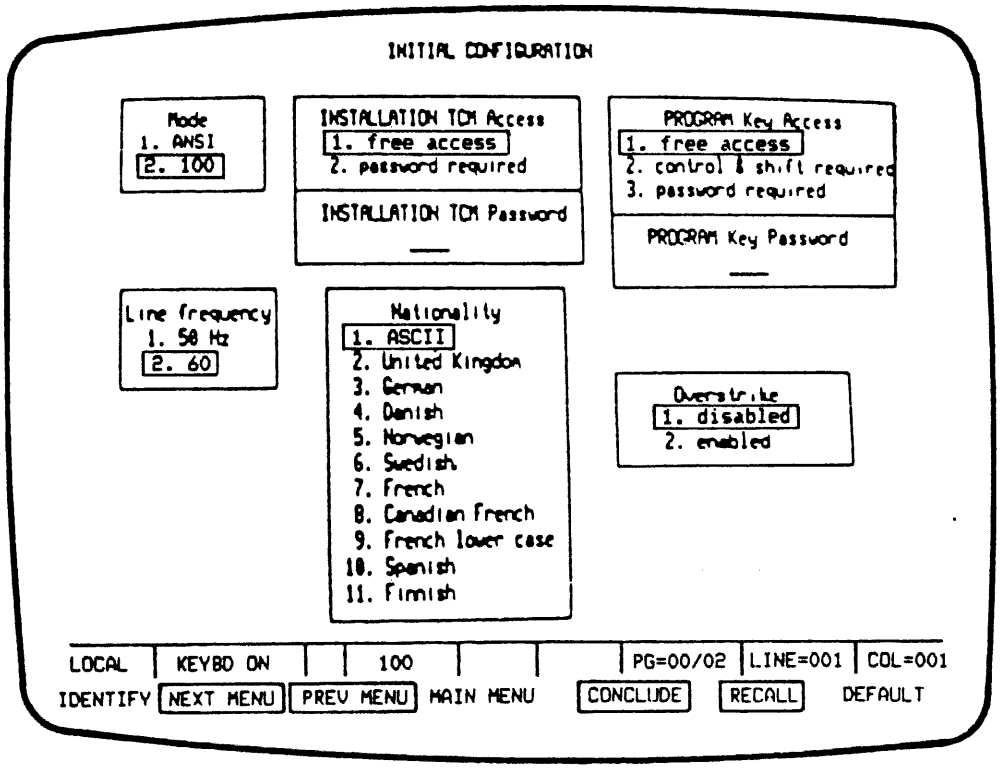


Figure 3-5 Initial Configuration Screen

This menu screen provides control over four parameters:

Mode -- The ATL-008 has two main operating modes - 100 and ANSI. 100 Mode is an emulation of a DEC VT100 (TM) terminal. ANSI Mode means that the terminal complies with applicable portions of ANSI Standard X3.64.

INSTALLATION TCM Access -- This menu provides either free access (no password required) or password required. If you select password required, reposition the cursor to the line in the box titled INSTALLATION TCM Password and type the four-character password to be used when accessing Installation TCM.

PROGRAM Key Access -- This menu provide three type of access to the PROGRAM softkey. Free access means that no special action is required to use the PROGRAM softkey. Control and shift required means that the CTRL and SHIFT keys must be pressed in conjunction with the PROGRAM softkey to use the PROGRAM softkey. Password required means that pressing the PROGRAM softkey results in a prompt indicating that a password is required. If you select Password required, reposition the cursor to the line in the box titled PROGRAM Key Password and type the desired password.

Line Frequency -- Place the cursor on the appropriate selection, 50 or 60 Hz, and press the IDENTIFY softkey.

Nationality -- This menu allows you to specify the nationality of the keyboard so that the screen characters match the characters on the keycaps of different national keyboards.

Overstrike -- If you are using a nationality that requires overstrike capability, use this box to enable or disable overstrike.

3.3.2 Terminal Configuration

The Terminal Configuration screen controls the display width, bell column, ANSI editing modes and status line, transmission mode, line wrap, scrolling, and tab stop settings. The screen is shown in Figure 3-6.

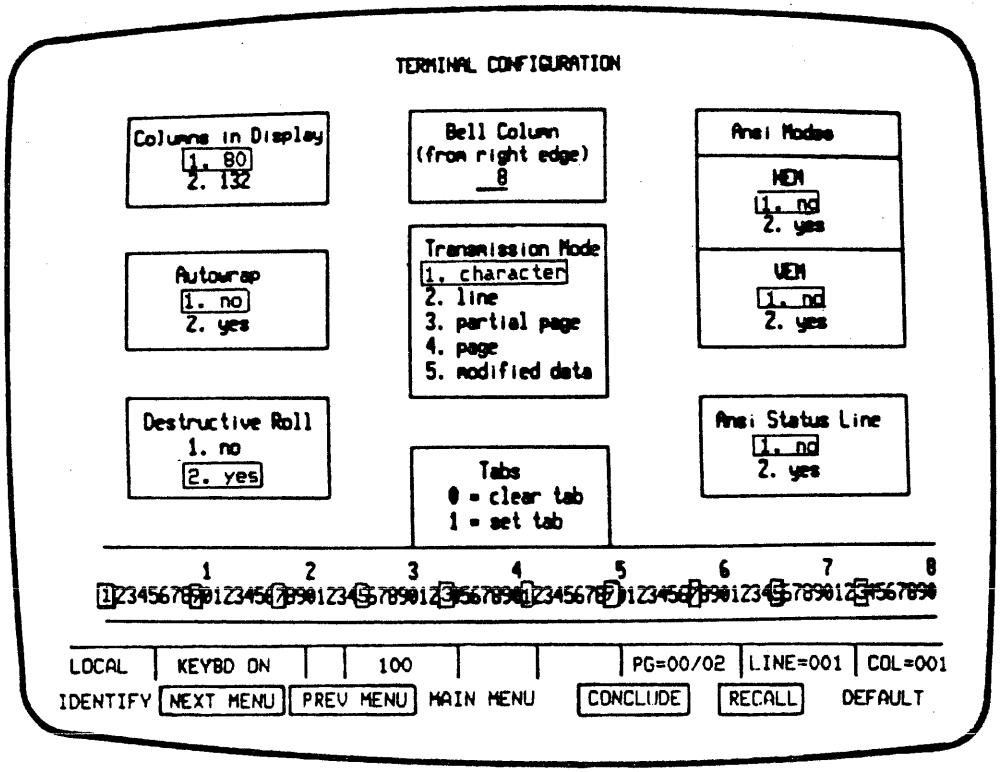


Figure 3-6 Terminal Configuration Screen

Columns_in_Display -- This menu permits selection of either an 80 or a 132 column display by placing the cursor on the appropriate selection and pressing the IDENTIFY softkey.

Bell_Column -- Enter the desired number of spaces from the right side of the screen. For example, if the ATL-008 is set to display 80 columns and you want the bell to ring when you pass column 72, enter a value of 8.

ANSI_Modes -- Select HEM (Horizontal Editing Mode) and VEM (Vertical Editing Mode) on or off by placing the cursor on the appropriate selection and pressing the IDENTIFY softkey.

Autowrap -- This parameter determines the action of the cursor when it reaches the last position on the line. The no selection means that the cursor will remain in the last position of the line. Subsequent characters overwrite the character in this position. The yes selection means that the cursor moves to the first position of the next line.

Transmission_Mode -- The manner in which screen data is selected for transmission depends on several things: SATM, MATM, TTM, GATM, FETM, the TCM space-suppression setting, the TCM transmission mode (this box), placement of SOE characters on the screen, and the key used to initiate the transmission. This section describes the interaction of the transmission mode, the keys used to initiate transmissions, FETM, the space-suppression setting, and SOE characters. See the section called ANSI MODE CONTROL FUNCTIONS for additional details.

There are five available transmission modes:

1. Character -- If this mode is selected, characters are sent to the host as they are typed at the terminal except for locally executed sequences. In this mode FETM and space suppression have no effect.
2. Line -- If this mode is selected, the current line is sent to the host when SEND PAGE or SEND LINE is pressed or when an DECXMIT command is used. The NEW LINE key (the key that occupies the position of a RETURN key) will also initiate a line transmission with different format. In this case the transmission is terminated by a CR or CRLF (depending on the configuration of the NEW LINE key -- see TCM Keyboard Parameters). Any end of line and end of block space-suppression and termination characters are not used.
3. Partial_Page -- If this mode is selected, the entire page is transmitted to the host when the SEND PAGE key is pressed or when an DECXMIT command is used. The terminal places an SOE in the last column of the current line and moves the cursor to the first position on the next line.
4. Page -- If this mode is selected, the entire page is transmitted to the host when the SEND PAGE key is pressed or when an DECXMIT command is used.
5. Modified_Data -- If this mode is selected, qualified areas that have been changed since the last transmission are transmitted to the host when the SEND PAGE key is pressed or when an DECXMIT command is used. HT characters are transmitted in place of unmodified qualified areas to act as place holders.

All of these except character are block transmission modes. In these modes, data entered at the terminal is generally processed locally and stored by the terminal for later transmission on command. There are some exceptions to this general rule as follows:

BREAK	Pressing the BREAK key sends a BREAK.
HOLD SCROLL	Pressing the HOLD SCROLL key once stops scrolling on the display; pressing it again allows scrolling to continue.
CTRL S or CTRL Q	Pressing these keys results in transmission of the corresponding code.
PF1 to PF16	Pressing these keys results in transmission of the previously programmed sequence.
BEESND	Any data bracketed by the BEESND sequence that makes data transmittable but not displayable is sent. This data may come from the keyboard or from a user-defined softkey.

In all block transmissions if FETM is set, data is sent without space suppression and without the addition of end of line characters (CR or CRLF). If FETM is reset and space suppression is used, trailing spaces are removed and end of line characters are added. If FETM is reset and space suppression is not used, trailing spaces are not removed and end of line characters are added. The TCM block termination character is independent of these items and is sent.

An SOE (Start Of Entry - C2H) character defines a mobile home position for transmission and printing operations. An SOE character may be placed on the screen by three methods: pressing the SOE key, sending the appropriate SS2 or SS3 sequence to the terminal, or performing Partial Page transmission.

All data to be transmitted must be bracketed between starting and ending positions (these starting and ending positions depend on the particular command, the terminal's mode settings, and the cursor position). If one or more SOE characters exist between the starting and ending positions, the SOE closest to the ending position is used as the home position. The SOE is not included in the transmission.

Destructive_Roll -- This parameter determines the action of a line feed when the cursor reaches the last line on the screen. The no selection means that the cursor will remain in the last line of the screen. Subsequent characters overwrite the characters in this line. The yes selection means that the data on the screen scrolls up and a new line is available. Data scrolling off of the top of the screen is lost. Compare with Autowrap.

Tabs -- The box labeled tabs contains instructions for setting and clearing tab stops. The line of digits immediately below this box is a ruler line that shows tab stops. Existing tab stops are shown in reverse video. These tab settings may be changed by positioning the cursor on the ruler line and following the instructions. To set a tab, position the cursor at the desired column of the ruler, and press 1 (one). The column number in the ruler line will be displayed in reverse video, indicating that a tab stop has been set at that column. To clear a tab, position the cursor at the desired column of the ruler, and press 0 (zero). The column number in the ruler line will be displayed in normal video,indicating there is no tab stop at that column.

ANSI Status Line -- If yes is selected and the terminal is in ANSI mode, pressing the CTRL key and the SELECT/STATUS keys simultaneously displays the ANSI status line as long as the keys are held down. The no selection disables the ANSI status Line.

3.3.3 PF Key Definition

PF keys (the SHIFTed and CTRL versions of the softkeys) generate a code sequence that has this general form:

Control Sequence	P	n	ST

where Control Sequence is an ANSI-compatible sequence indicating APC, OSC, PM, DCS, or SS3; P is the program function key parameter; n is a unique code corresponding to the specific PF key; and ST is a string terminator.

Figure 3-7 illustrates the PF key definition screen. It contains two boxes. The left box contains three columns labeled, from left to right, PF Key, Unique Code, and Control Sequence. The Control Sequence is the only thing that may be changed on this screen. The Control Sequence consists of a single digit between 1 and 5 inclusive. The meanings of the Control Sequence digits are given in the box on the right.

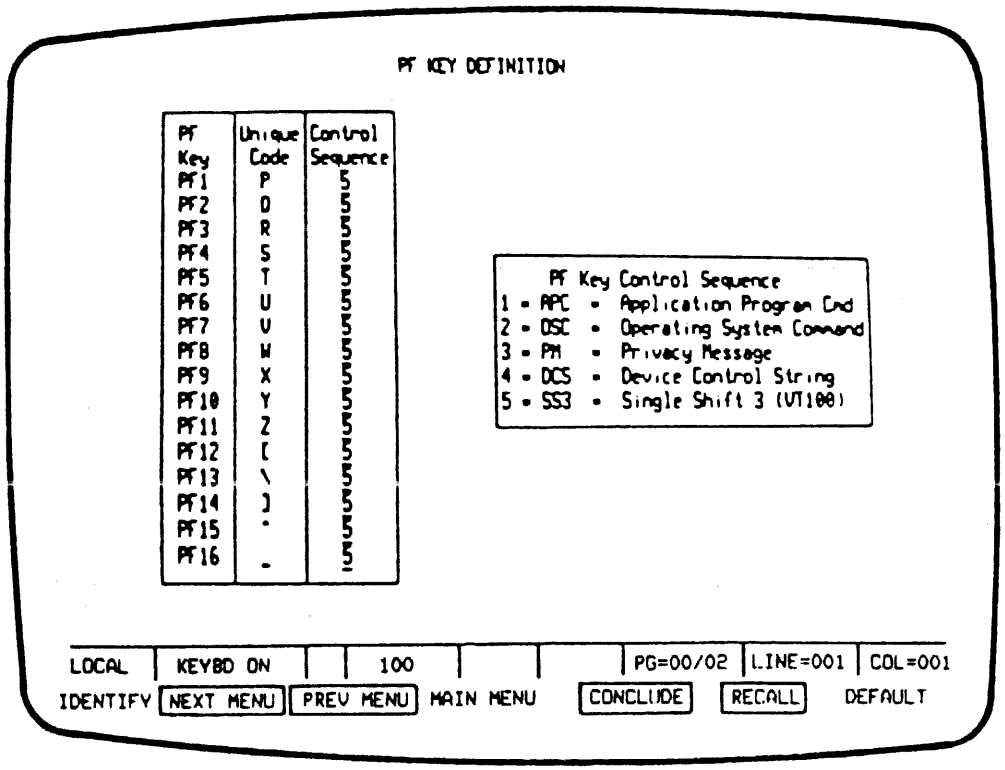


Figure 3-7 PF Key Definition

For example, if PF4 is to be defined as an ANSI-compatible Application Program Command (APC), the Control Sequence for PF4 should be set to a 1 (one). The PF4 key would then send the ANSI APC string:

APC	P	S	ST

where APC is [ESC _]. The ANSI-compatible APC sequence is:

APC	P	n	ST

or

ESC	_	P	n	ST

where n is the Unique Code given by the second column of this screen. In the above example, PF4 corresponds to S, therefore S replaces n in the string.

3.3.4 Screen Page Allocation

This screen has two boxes as shown in Figure 3-8. The smaller box displays the Total Available Space and the manner in which it is allocated. These values may not be changed directly. They are displayed for the user's information. This same information is also displayed on the Buffer Allocations screen. The actual values on your screen may vary from the illustration. The larger box allows definition of up to 12 screen pages and their corresponding screen windows. As you change the page definitions, take note of the changes in the smaller box. The first two pages have default values specified from the factory. These and the blank areas may be changed.

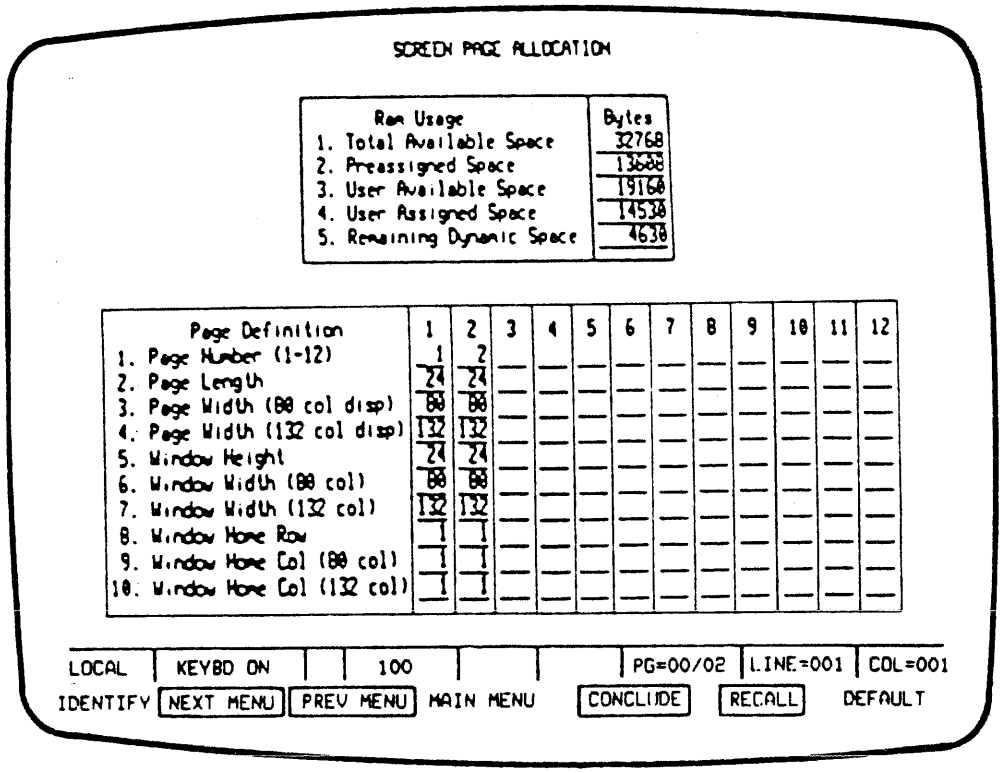


Figure 3-8 Screen Page Allocation

As changes are made in the page allocations, the contents of the smaller box are updated to inform the user of remaining free space.

The larger box allows specification of the page size (height and width), the window height, two different window widths (one for 80-column display and one for 132-column display), and the window home row and column (one for 80-column display and one for 132-column display).

The window size (height and width) must not be greater than the size of the corresponding page. Window size may not exceed the dimensions of the screen. Each page has one window. If the page is larger than the window, the page may be moved relative to its window by the SCROLL keys (CNTL and cursor control keys). The data moves in the direction given on the SCROLL keys. Two different widths are allowed to provide different arrangements for 80-column or 132-column display. For smooth scrolling to work, the page and window widths must be set equal to the display width -- either 80 or 132 columns. The home row and column of a window specify the location of the upper left corner of the window on the screen. Windows allow various pages to be displayed on the same screen at the same time. See Figure 3-9.

Pages, and their corresponding windows, have independent on-line or local status. The status line (immediately above the softkey labels) is the key to what is happening in each page. As the PAGE UP and DOWN keys (SHIFTed cursor UP and DOWN) move the cursor from page to page, the status line reflects any corresponding changes in status. It also gives the current page, line and column address of the cursor. When the cursor is moved from one page to another, a position marker resembling the cursor is left on the page. Overlapping windows are allowed (the default windows overlap entirely), but may not be displayed concurrently. As the cursor moves between overlapping windows, each of the windows is displayed by itself.

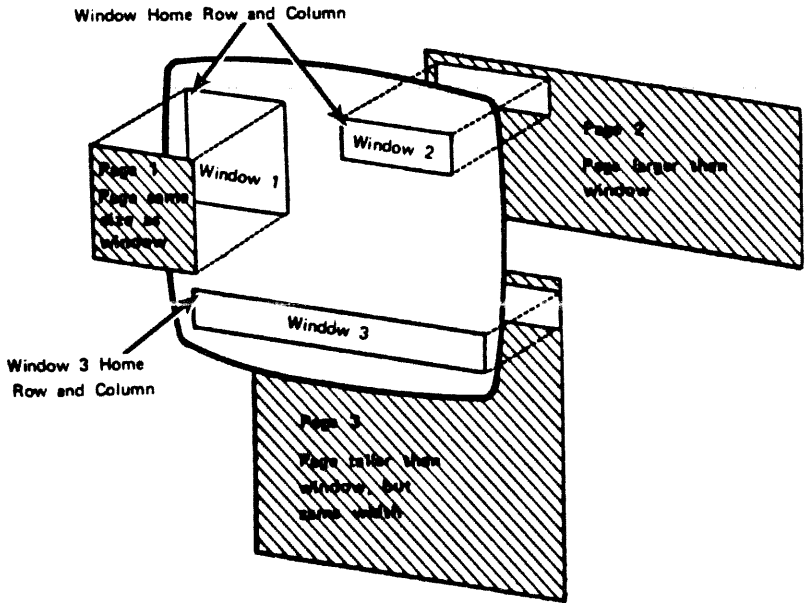


Figure 3-9 Pages and Windows

3.3.5 Buffer Allocations

This screen has three boxes as shown in Figure 3-10. The upper box displays the Total Available Space and the manner in which it is allocated. These values may not be changed directly. They are displayed for the user's information. This same information is also displayed on the Screen Page Allocations screen. As you change the remainder of this screen, watch the values in this box. The other two boxes allow specification of Buffer Space Usage and Special RAM usage. As changes are made in the Buffer and RAM allocations, the contents of the upper box are updated to inform the user of remaining free space.

Buffer Space Usage -- This allows specification of the number of 128-byte buffers to be allocated to the Main and Auxiliary Port receivers and transmitters. It also provides buffers for Qualified and Selected areas as defined by ANSI 3.64 for forms-type operations.

Special RAM Usage -- Space for user function keys and bootloading is allocated in this box. If you want to define a user key, you must first allocate at least one buffer using this menu. If you want to download a Boot Load string, you must first allocate enough space for it using this box. Both allocations provide contiguous space.

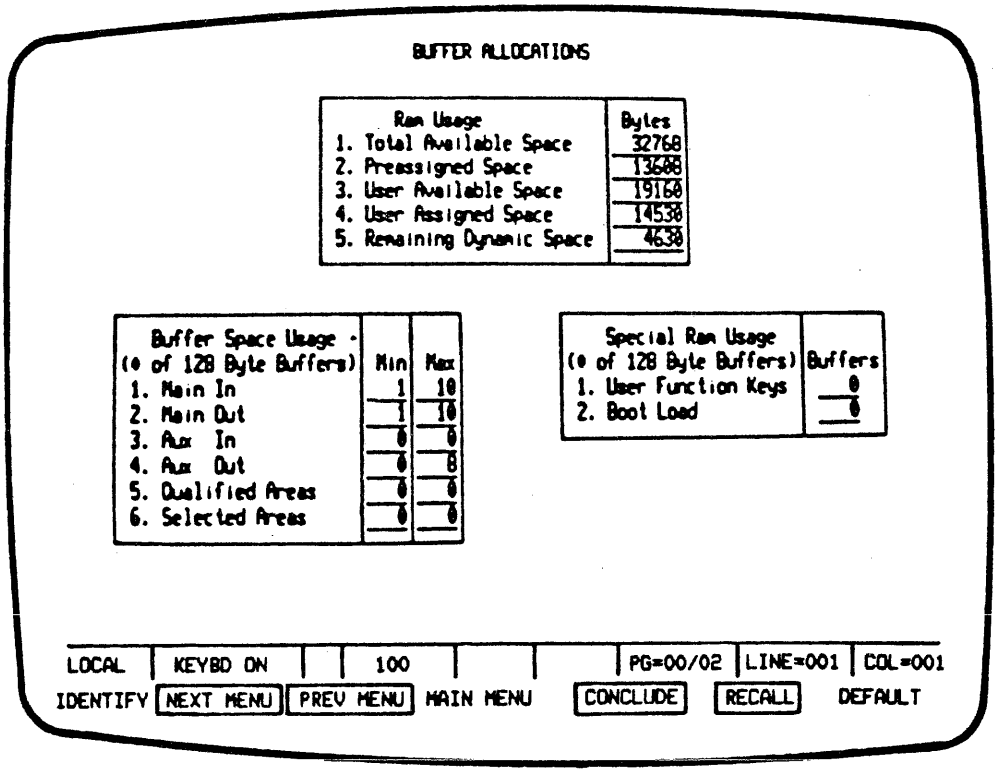


Figure 3-10 Buffer Allocations

3.3.6 Main Port Hardware Interface

The Main Port Hardware Interface screen allows selection of baud rate, data length, parity, stop bits, parity checking, and electrical interface as illustrated in Figure 3-11. These selections apply only to the Main Port. The Auxiliary Port has a separate menu.

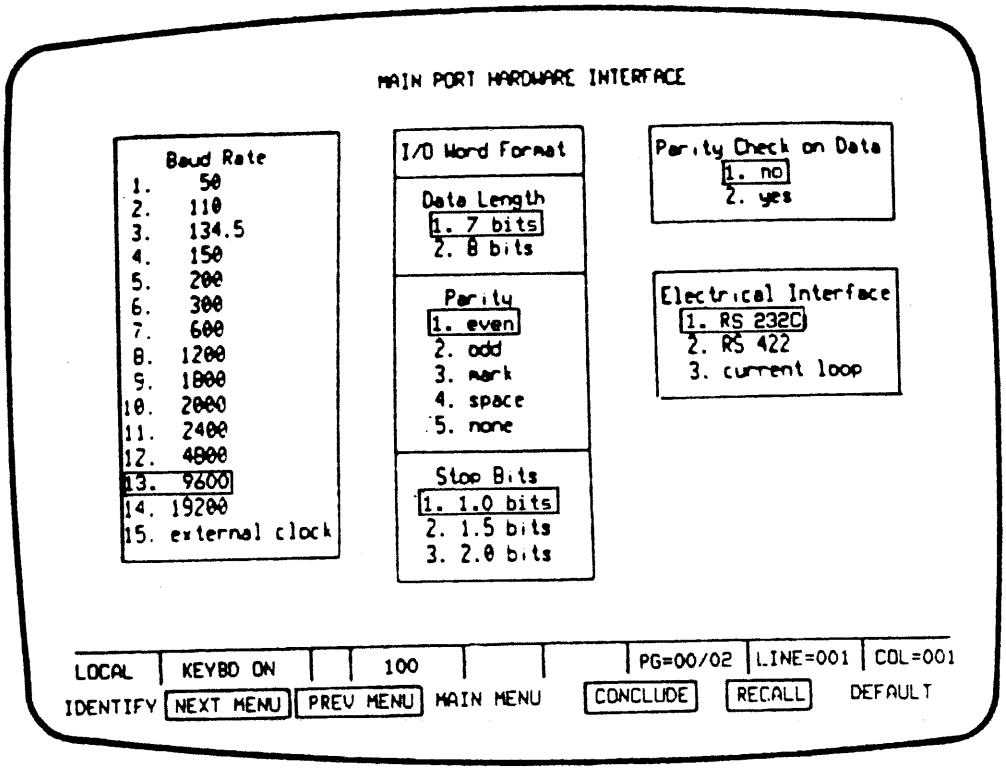


Figure 3-11 Main Port Hardware Interface

Baud Rate -- Standard baud rates from 50 to 19200 are available in this menu, as is a provision for an external clock.

I/O Word Format -- Standard data lengths, parity settings, and number of stop bits are specified with this menu.

Parity Check on Data -- Select yes to check parity on output; otherwise select no.

Electrical Interface -- ATL-008 terminals may be ordered with a Main Port that supports one of three combinations of interfaces: RS232C only, RS232C and RS422, or RS232C and 20ma Current Loop. Use this selection to specify the interface that you are using.

3.3.7 Main Port Protocol

The Main Port Protocol screen allows specification of communications parameters for the Main Port as illustrated in Figure 3-12.

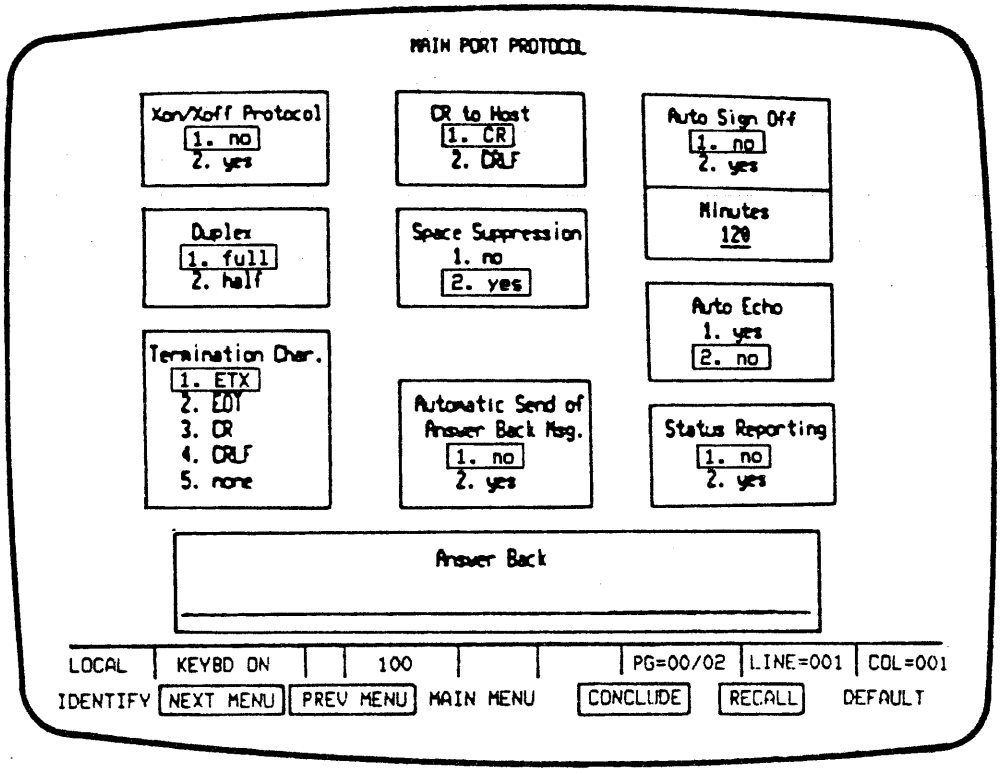


Figure 3-12 Main Port Protocol

Xon/Xoff Protocol -- If you wish to use Xon/Xoff, select yes; otherwise, select no.

Carriage Return to Host -- This specifies what type of carriage return is sent to the host for block send operations. If CR is selected, each carriage return is sent as a simple carriage return. If CRLF is selected, each carriage return is sent as a carriage return-line feed.

Auto Sign Off -- If yes is selected, the terminal will automatically sign off (pin 20 drops) from the host after a programmable time limit has passed. The time limit is programmed by entering the desired number of minutes on the line in the adjacent box. Auto sign off may be disabled by selecting no. Any action from the keyboard or ports resets the timer.

Duplex -- ATL-008 terminals support full and half duplex communications. Full duplex means that two-way simultaneous communications will occur. While using full duplex, the RTS pin will be asserted at all times. Half duplex means that two-way communications can occur, but only in one direction at any one time. While using half duplex, the RTS pin will be asserted before transmission of the character and dropped after transmission.

This selection interacts with the Modem Definition screen. If a CTS is required, then data cannot be sent on a half duplex channel unless the CTS pin is asserted.

Space Suppression -- If space suppression is enabled (yes), spaces at the end of the line are not transmitted in block send operations.

Auto Echo -- If yes is selected, all keyboard data will be processed and displayed by the terminal. If no is selected, keyboard data will not be processed locally and the host must echo keyboard data if it is to be displayed by the terminal.

Termination Character -- The following block termination characters are available:

ETX	End of text	03 Hex
EOT	End of transmission	04 Hex
CR	Carriage Return	0D Hex
CRLF	Carriage Return-Line feed	0D0A Hex
None	No termination character	

The selected character is appended to the end-of-block transmissions.

Automatic Send of Answer-Back Message -- The programmable answer-back message is enabled with this box. To program the message, position the cursor on the line in the box titled Answer Back and type the message.

Status Reporting -- The yes selection enables status reporting to the host. This means that at the start of any confidence test, the terminal will send a Device Status Request busy sequence [ESC [2 n] to the host indicating that the terminal is busy. When the test is concluded successfully, the ATL-008 sends the Device Status Request ready sequence [ESC [0 n], indicating that the terminal is again ready.

3.3.8 Aux Port Hardare Interface

The Auxiliary Port Hardware Interface screen allows selection of baud rate, data length, parity, stop bits, parity checking, and electrical interface as illustrated in Figure 3-13. These selections apply only to the Aux Port. The Main Port has a separate menu.

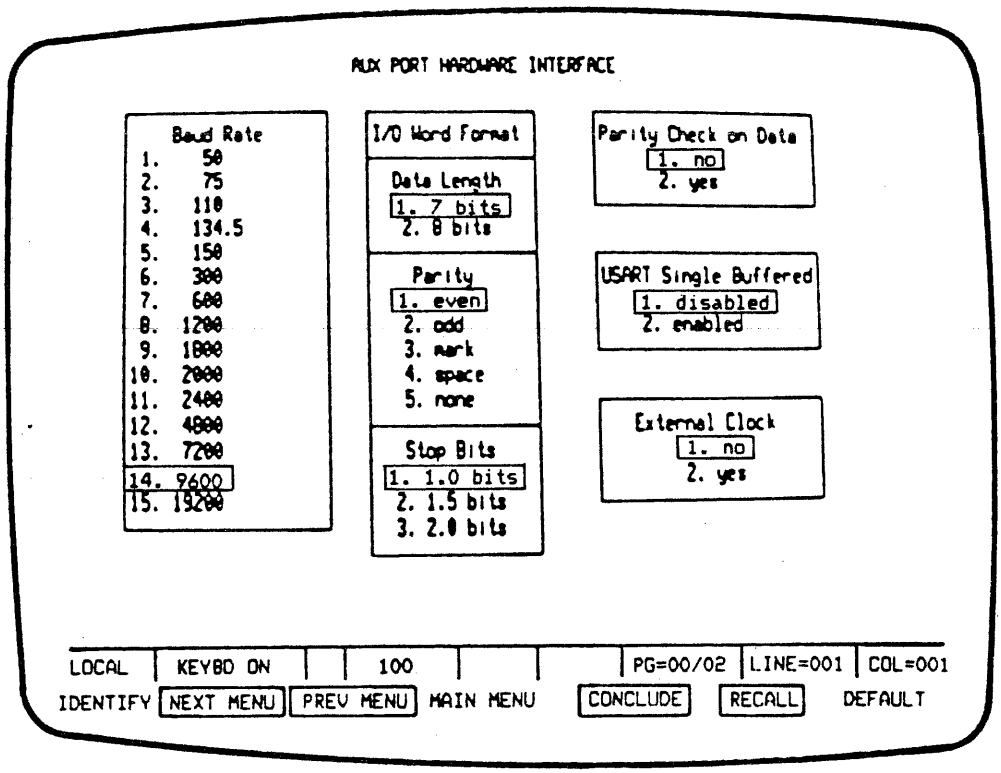


Figure 3-13 Aux Port Hardware Interface

Baud Rate -- Standard baud rates from 50 to 19200 are available in this menu.

I/O Word Format -- Standard data lengths, parity settings, and number of stop bits are specified with this menu.

Parity Check on Data -- Select yes to check parity; otherwise select no.

USART Single Buffered -- The USART employed by the ATL-008 terminal permits selection of either single or double buffering on output.

External Clock -- If you wish to use an external clock for aux port communications (for isosynchronous communications, for example), select yes; otherwise, select no.

3.3.9 Aux Port Protocol

The Aux Port Protocol screen allows specification of communications parameters for the Auxiliary Port as illustrated in Figure 3-14.

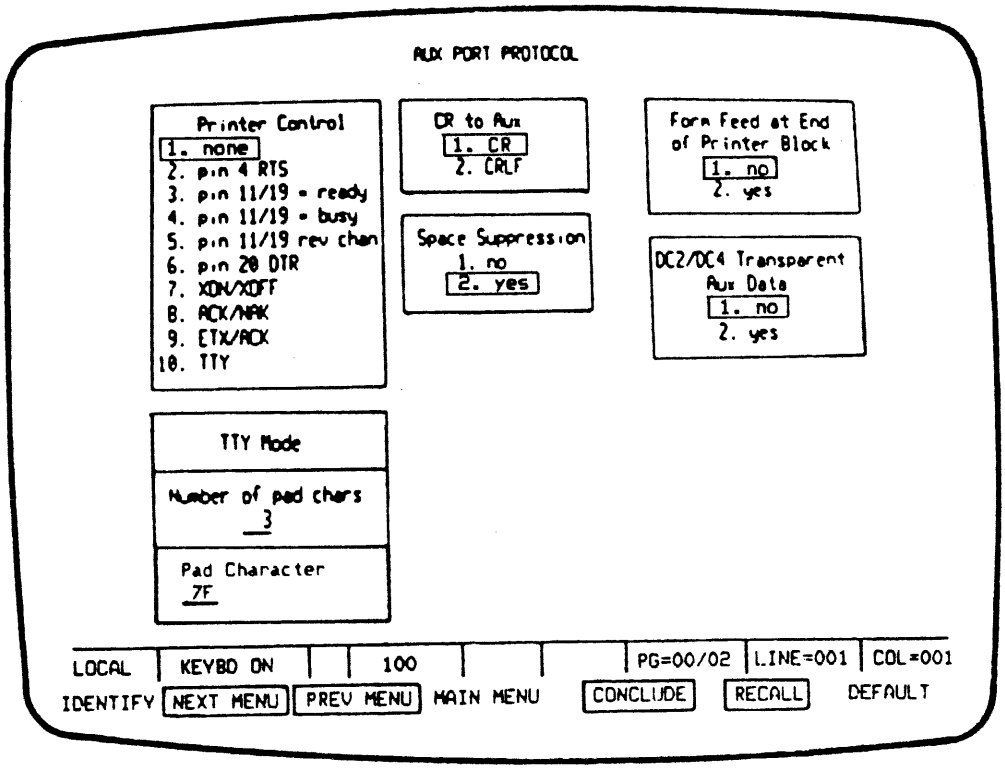


Figure 3-14 Aux Port Protocol

Printer Control -- This box lists a variety of printer control methods.

None -- No printer protocol. The printer must handle the data at the specified baud rate.

Pin 4 RTS -- Transmission out is enabled if pin 4 (RTS) is asserted.

Pin 11/19 = ready -- When pin 11/19 is negated, data transmission is enabled. When pin 11/19 is asserted, data transmission is disabled.

Pin 11/19 = busy -- When pin 11/19 is asserted, data transmission is enabled. When pin 11/19 is negated, data transmission is disabled.

Pin_11/19_rev_chan -- After sending a character to the printer that will take time to execute (CR, LF, LF, HT, VT), the terminal sends an EOM (03 Hex). Transmission stops until pin 11 or pin 19 goes on then off or until 4 seconds have passed. Transmission then resumes.

Pin_20_DTR -- When pin 20 (DTR) is asserted, data transmission out the AUX port is enabled.

XON/XOFF -- This is identical to and independent of the main port XON/XOFF with the following exceptions. If, while receiving data to the aux port, the terminal senses an XOFF condition (receiver buffer 80% full), it will transmit an XOFF and drop pin 5 (CTS). When the buffer is 20% full, the terminal will transmit an XOFF and raise pin 5 (CTS) to show that it can accept data again.

ACK/NAK -- This is identical to XON/XOFF except that ACK is the XON code and NAK is the XOFF code.

ETX/ACK -- After each block of 78 characters is transmitted, an ETX is transmitted. An ACK is expected for each ETX. If there are two outstanding ETX's which have not been ACKed, the terminal will cease transmission.

IIY -- Each transmission ends with the specified number of pad characters. The pad character and the number of pad characters to be sent with each transmission are specified in IIY_Mode in this same screen.

CR_to_Aux -- This specifies what type of carriage return is sent out the auxiliary port in block send operations. If CR is selected, each carriage return is sent as a simple carriage return. If CRLF is selected, each carriage return is sent as a carriage return-line feed.

Form_Feed_at_End_of_Printer_Block -- When doing any sort of block print operation, this selection allows addition of a form feed at the end of the print operation to eject an extra page from the printer. The yes selection adds the form feed.

Space_Suppression -- If space suppression is enabled (yes), spaces at the end of the line are not transmitted.

DC2/DC4_Transparent_Aux_Data -- This option provides the host with a method of sending data to a printer that is connected to an ATL-008 terminal without disturbing the operation of the terminal. If this selection is set to yes, then transparent data may be sent to the printer by first sending a DC2 to the terminal. All subsequent data will be received by the terminal and re-sent out the AUX port to the printer. Data entered from the keyboard is still sent to the host. This data may be echoed to the terminal by sending a DC4 to the terminal which will interrupt the printer long enough to display the character. The host may then resume the transparent print by sending another DC2. Using this method, data can become backed up in the auxout buffers. When the last buffer becomes two-thirds full, the terminal will send a DC4 to the host. When the buffers empty sufficiently to resume transmission, the

terminal sends a DC2 to the host. Data to the terminal not intended for the printer may be turned off and on by DC1 and DC3 characters respectively.

TTY Mode -- If TTY printer control is selected, this box allows specification of the number and type of pad characters to be sent after a Carriage Return. Position the cursor and enter the number of pad characters between 0 and 3 inclusive. Reposition the cursor and enter the hex code for the desired pad character.

3.3.10 Modem Definitions

This screen allows specification of communications parameters for modems as shown in Figure 3-15. These parameters include modem type, modem signals and various delay times.

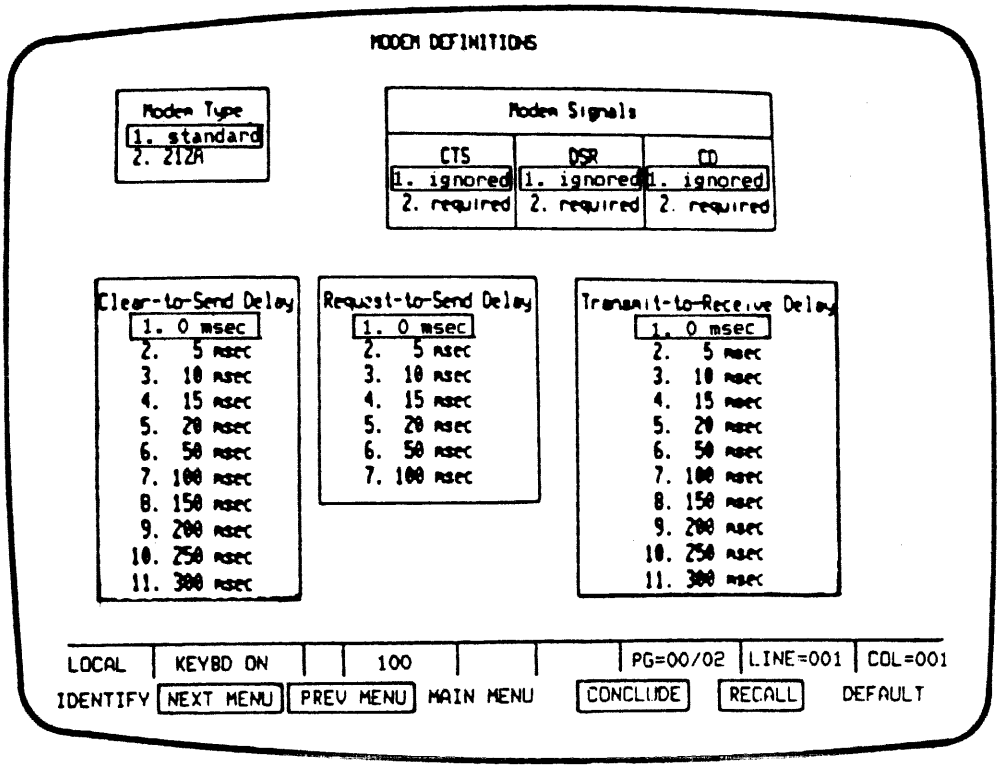


Figure 3-15 Modem Definitions

Modem_Type -- Either standard or 212A type may be selected.

Modem_Signals -- This box contains three selections:

- CTS -- This selection determines if a Clear-To-Send signal is required or ignored by the terminal.
- DSR -- This selection determines if a Data-Set-Ready signal is required or ignored by the terminal.
- CD -- This selection determines if a Carrier-Detect signal is required or ignored by the terminal.

Clear-to-Send_Delay --This box selects the amount of time the terminal waits between receipt of a CTS and the transmission of output data.

Request-to-Send_Delay -- This box selects the amount of time that RTS remains asserted after transmitting the data. This is necessary with some modems to ensure that the output characters clear the transmitter before RTS is dropped.

Transmit-to-Receive_Delay -- This box specifies the amount of time the terminal waits to receive data after a transmission.

3.4 OPERATOR ICM MAIN MENU

Operator TCM controls video parameters, keyboard parameters, and graphic key definition. It is accessed during normal operation by pressing CONFIGURE or at the end of Installation TCM by pressing NEXT MENU. The Operator TCM Main Menu is shown in Figure 3-16.

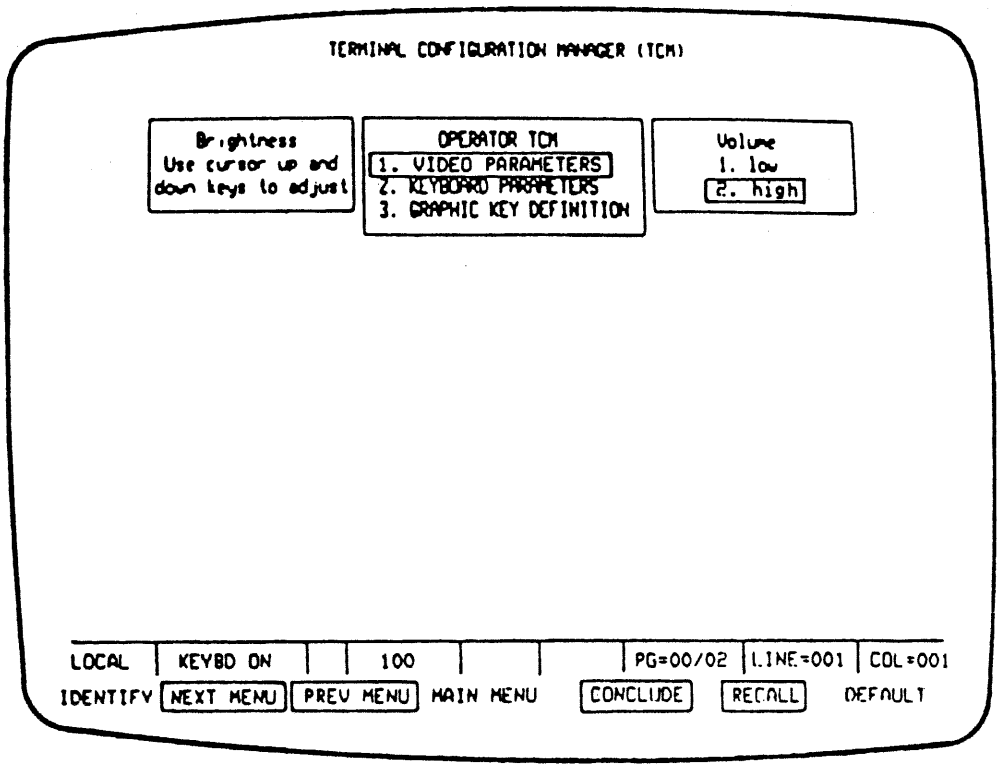


Figure 3-16 Operator ICM Main Menu Screen

This screen controls the following three parameters:

Brightness -- To change the brightness, position the cursor anywhere in the box. Press the cursor up key to increase the brightness, and the cursor down key to decrease the brightness. The audible alarm will sound when the brightest or faintest limit is reached. Normal or half intensity is determined by the setting of the Display parameter in the Video Parameters menu.

Operator ICM -- This menu leads to three additional menus for control of Video Parameters, Keyboard Parameters, and Graphic Key Definition.

Volume -- To select low or high volume level for the audible alarm and keyclick, use the cursor-positioning keys and place the cursor on either the low selection or the high selection. Press the IDENTIFY softkey. The selection you chose will be highlighted (displayed in reverse video). When you exit TCM or change to another TCM screen, the new volume level takes effect.

3.5 OPERATOR ICM SUBMENUS

3.5.1 Video Parameters

These settings control cursor appearance, display appearance, and scrolling. The screen is illustrated in Figure 3-17.

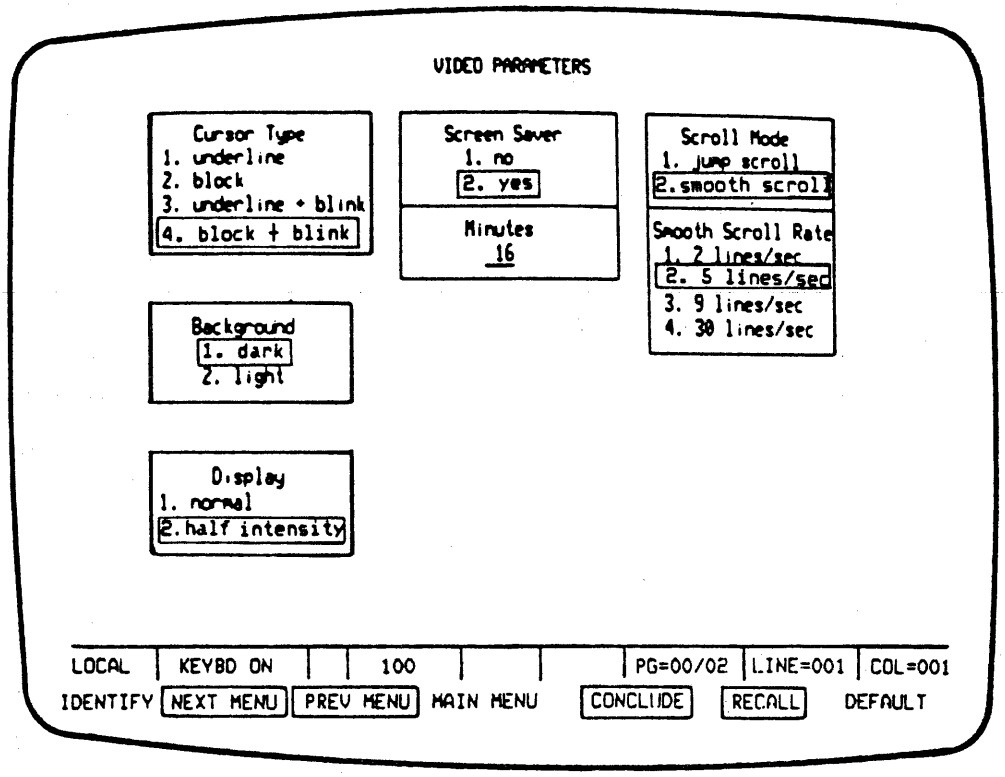


Figure 3-17 Video Parameters Screen

Cursor Type -- ATL-008 terminals offer four types of cursor: underline, block, blinking underline, and blinking block.

Background -- This box offers two selections: dark and light. If dark is selected, the screen display will consist of light characters on a dark background. If light is selected, the display will consist of dark characters on a light background.

Display -- This box offers two selections for controlling screen intensity or brightness: normal and half intensity. Half Intensity is not as bright as normal.

Screen Saver -- If screen saver is enabled (yes selection), the display will blank after a period of time has elapsed since the last keyboard or host input. To specify this time period, reposition the cursor to the line in the adjacent box labeled Minutes and type the desired number of minutes. A moving cursor is left on the screen to indicate that screen saver is

functioning and that the terminal is still turned on. This prevents screen burn-in. Only the display is blanked; memory and other terminal functions remain intact. For example, if yes is selected and Minutes is set to 10, if there is no input to the terminal from either the keyboard or the host computer for ten minutes, the screen will shut off except for the moving cursor. Pressing any key will cause the screen to return and that keystroke is lost. Any input from the host also causes the screen to reappear.

Scroll Mode -- This box offers a choice between smooth scrolling and jump scrolling. Jump scrolling means that any scrolling will involve entire text lines. Smooth scrolling means that individual scan lines scroll. Smooth scrolling is easier to read. If smooth scroll is selected, the scroll rate may be set in the adjacent box at 2, 5, 9, or 30 scan lines per second. Smooth scrolling only functions if the window width and page width equal the display width -- either 80 or 132 columns.

3.5.2 Keyboard Parameters

The keyboard parameters screen, shown in Figure 3-18, determines several keyboard functions.

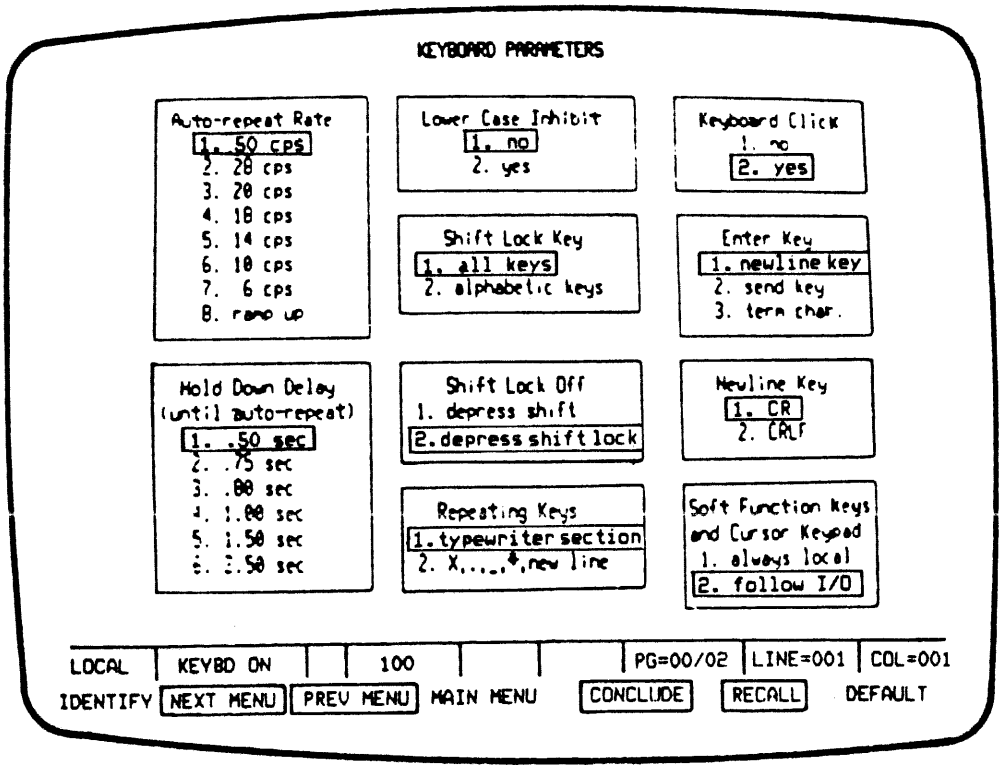


Figure 3-18 Keyboard Parameters Screen

Auto-Repeat_Rate -- This box determines how fast the repeating keys repeat. A variety of speeds are available. Cps means characters per second. The ramp up selection means that the longer you hold the key down, the faster it repeats. This selection interacts with the Hold_Down_Delay setting and with the Repeating_Keys setting.

Hold_Down_Delay -- This box specifies how long a key must be held down before it begins to repeat.

Lower_Case_Inhibit -- The no selection means that upper and lower case alpha keys function. The yes selection indicates upper case only.

Shift_Lock_Key -- This sets the scope of the shift lock key. If all_keys is selected, SHIFT LOCK affects all keys. If alphabetic_keys is selected, SHIFT LOCK acts as an alpha lock key. This setting interacts with the Shift_Lock Off setting.

Shift_Lock_Off -- This box determines how to get out of shift lock. If depress_shift is selected, press the shift key to unlock shift lock (like a typewriter). If depress_shift_lock is selected, press the shift lock key to unlock shift lock (SHIFT LOCK toggles).

Repeating_Keys -- This selection determines which keys repeat when held down. Either typewriter_section or a smaller group of keys (X, period, underscore, asterisk, semicolon, and NEW LINE) may be selected. In either case, the following keys do not repeat: SELECT, all PF keys, shift, shift lock, CTRL, ESC, SOE, HOLD SCROLL, UNLOCK, CONFIG, SEND PAGE/LINE, keypad minus, keypad comma, keypad period, ENTER, and BREAK in ANSI mode (BREAK can repeat in 100 mode).

Keyboard_Click -- This box determines if the keys click when pressed.

Enter_Key -- This selects the operation of the ENTER key in the numeric pad. The three selections are:

Newline_Key -- This selection sets the function of the ENTER key so that it is the same as the function of the NEWLINE key.

Send_Key -- This selection causes ENTER to function as a send key. The send key depends on the Transmission mode selected in Terminal Configuration in Installation TCM. If Line_transmission is selected as the transmission mode in Terminal Configuration and the ENTER key is defined as a send_key in this menu, the ENTER key duplicates the SEND LINE key function. If any other block transmission mode is selected as the transmission mode in Terminal Configuration and the ENTER key is defined as a send_key in this menu, the ENTER key duplicates the SEND PAGE key.

Termination_Character -- This selection sets the function of the ENTER key so that it sends the termination character selected in Main Port Protocol in Installation TCM.

Newline_Key -- This box determines the function of the NEWLINE key (the key with an arrow going down and to the left). CR causes this key to function as a carriage return; when it is pressed, the cursor goes to the beginning of the current line. CRLF causes this key to function as a carriage return-line feed key; when it is pressed, the cursor goes to the beginning of the next line.

Soft Function Keys and Cursor Keypad -- This box allows you to specify the action of the soft function keys and the cursor control keys. Always local means that these keys are not transmitted out any communications port. Follow I/O means that these keys will be transmitted during send operations during block transmissions or immediately during character transmission mode.

3.5.3 Graphic Key Definition

This screen allows easy redefinition of the graphics keys. The screen has three columns as shown in Figure 3-19. The first column lists the graphics keys by number, the second column lists the corresponding current hex codes, and the third column contains the corresponding display character.

To define a graphic key, position the cursor on the hex code for the desired key and type the hex code for the desired character. This is the only column you may change directly. The display code column will reflect the change and allow you to see the new display character. Hex codes for the various display characters may be found in the character generator table.

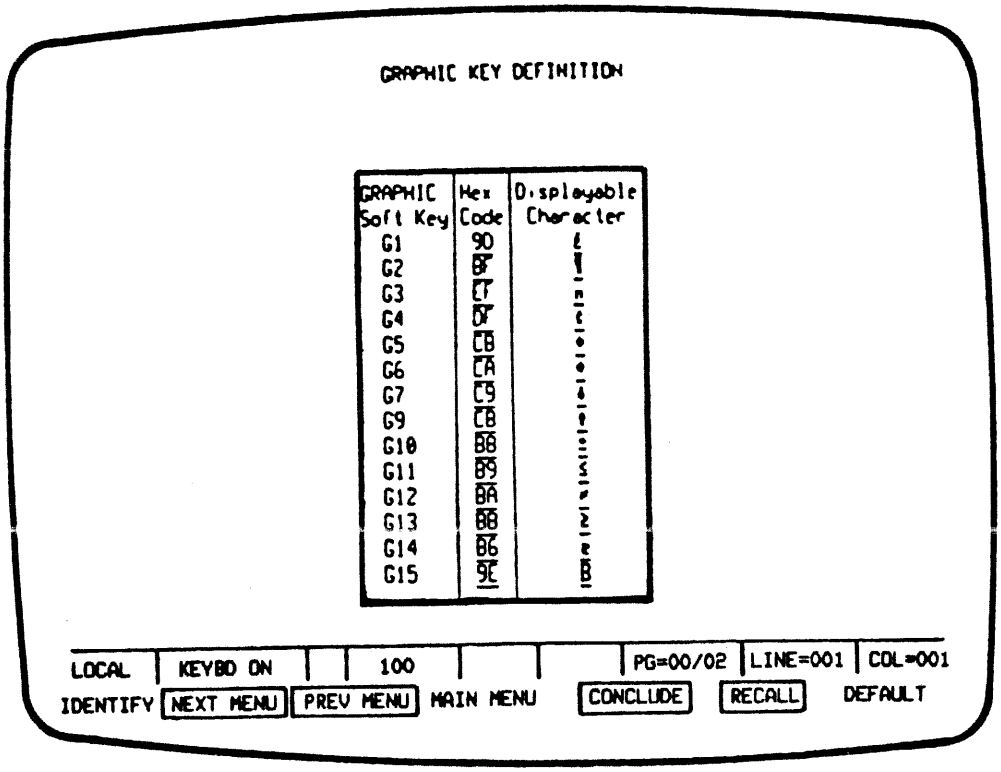


Figure 3-19 Graphic Keys Definition Screen

3.6 TCM SUMMARY REPORTING

Two of the second set of TCM soft keys provide a summary of the current TCM settings. The SUMMARY and PRNT SUM softkeys display and print this information. The summary may not be modified, except by changing the TCM setting with the correct menus. These screens are displayed for the operator's information only. They list the various TCM parameters and their corresponding selections. The screens for default settings are shown in this section; your screens may vary considerably depending on options and individual needs and preferences. The summary may be printed for a hard copy record by using the PRNT SUM softkey.

The Installation TCM Summary consists of seven screens of data. These screens are accessed by pressing the CONCLUDE softkey and then pressing the SUMMARY softkey repeatedly to view the Installation TCM Summary screens and the Operator TCM Summary screens in the order presented in this section. Typical screens are shown in Figures 3-20 through 3-27. The screens for PF key definition, screen page allocation, and buffer allocations are nearly identical to their corresponding TCM definition screens. The softkey labels are different.

The Operator TCM Summary consists of two screens. They are accessed by pressing the CONCLUDE softkey and then pressing the SUMMARY softkey repeatedly. Typical screens are shown in Figures 3-26 through 3-27. The screen for graphic key definition is nearly identical to its corresponding TCM definition screen. The softkey labels are different.

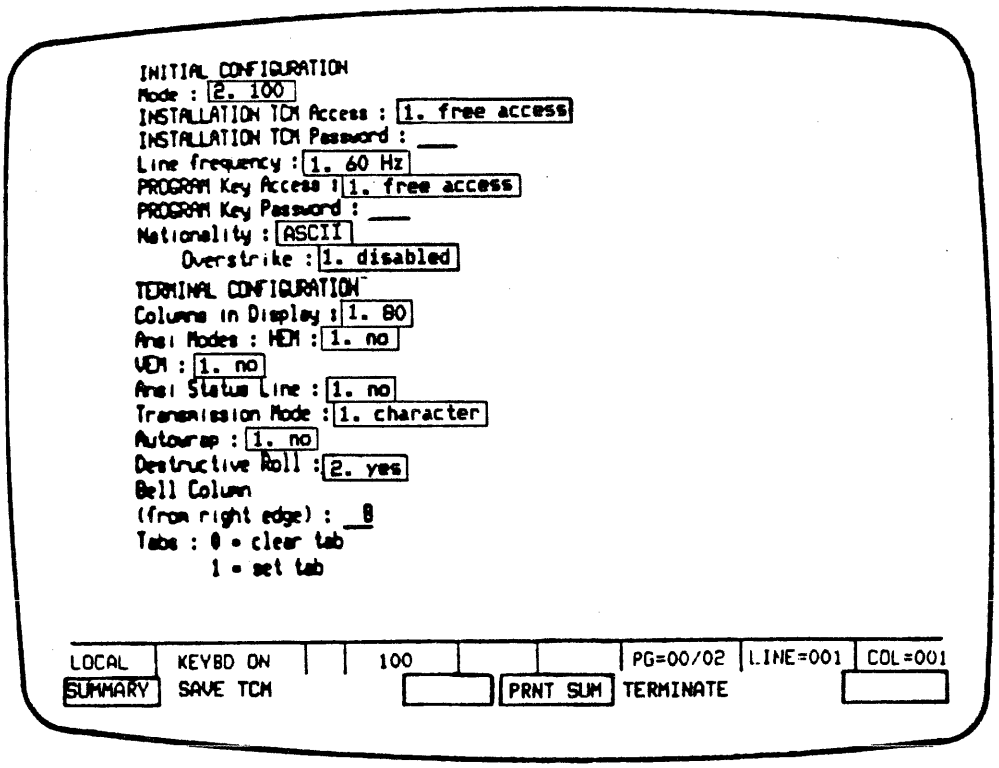


Figure 3-20 First Installation TCM Summary Screen

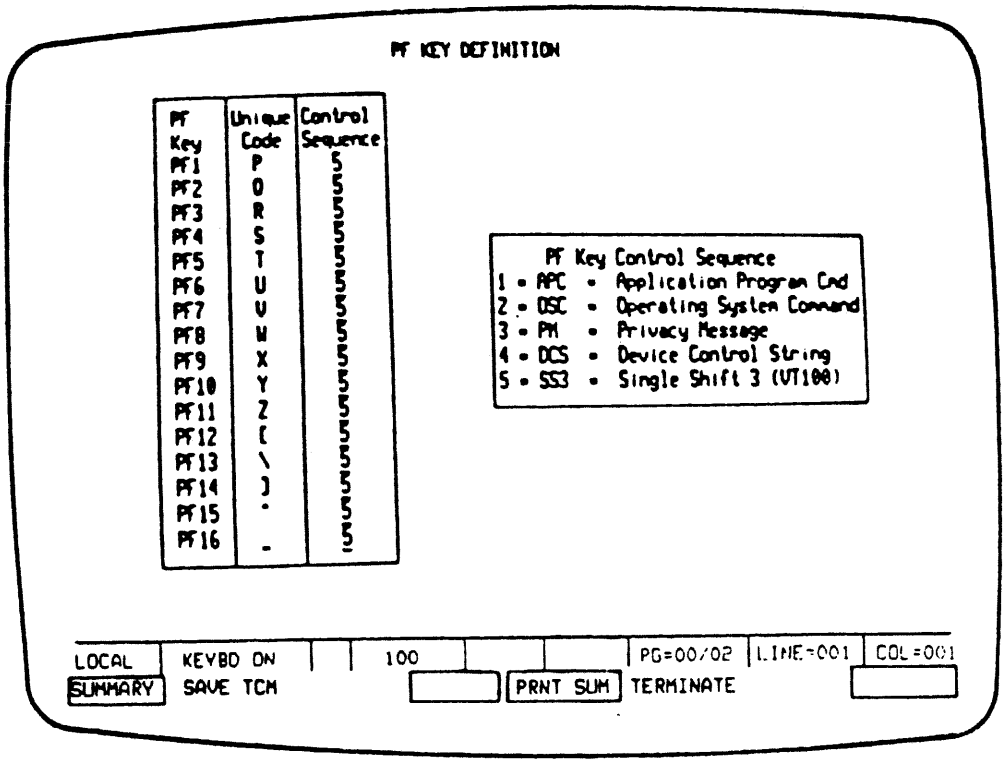


Figure 3-21 Second Installation ICM Summary Screen

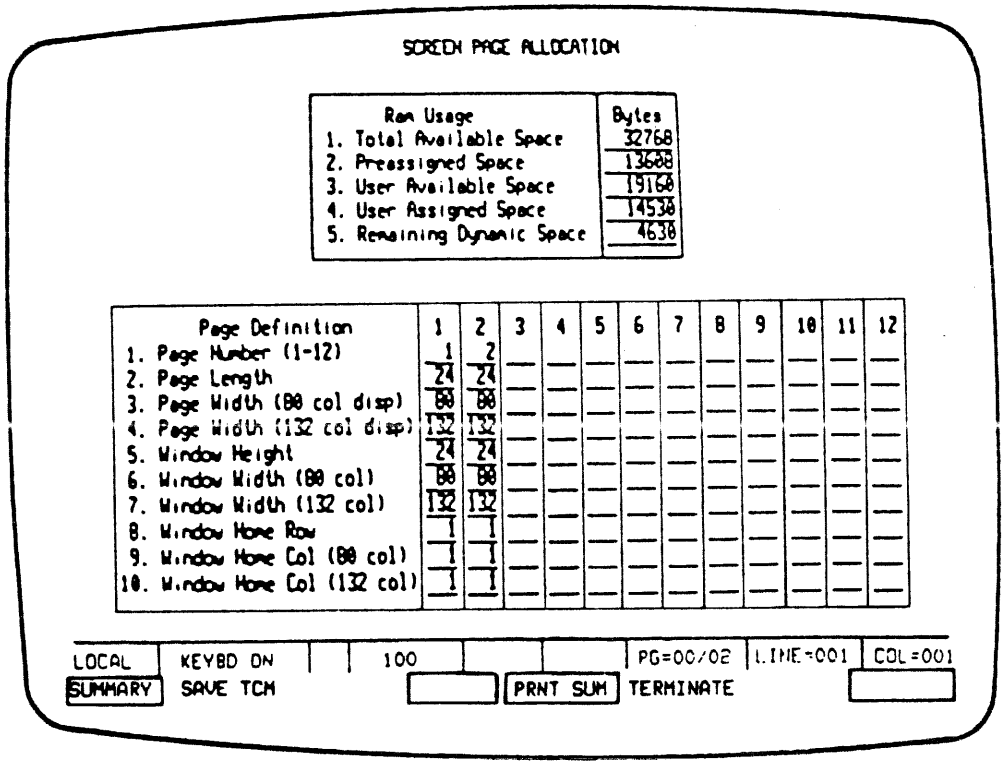


Figure 3-22 Third Installation ICM Summary Screen

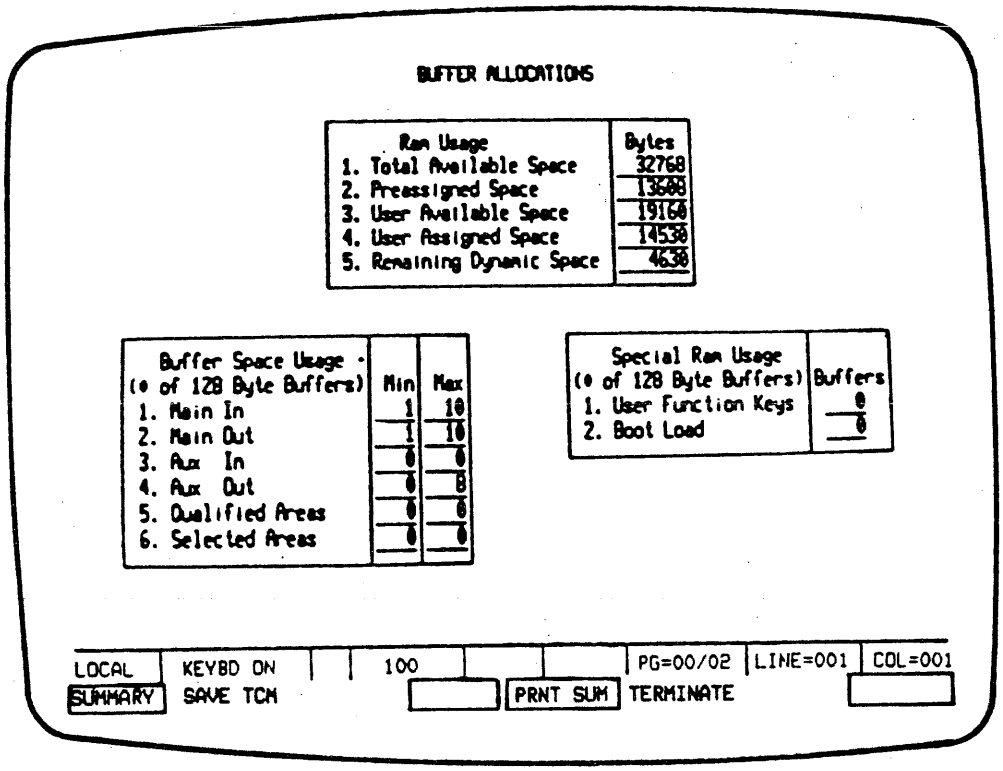


Figure 3-23 Fourth Installation ICM Summary Screen

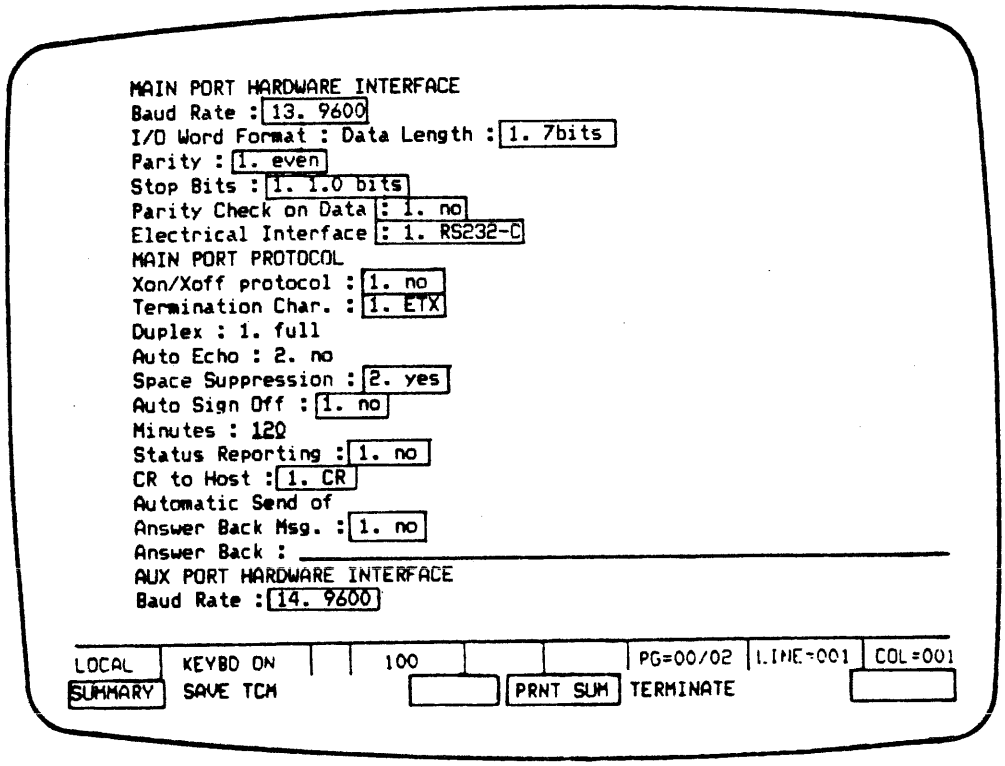


Figure 3-24 Fifth Installation ICM Summary Screen

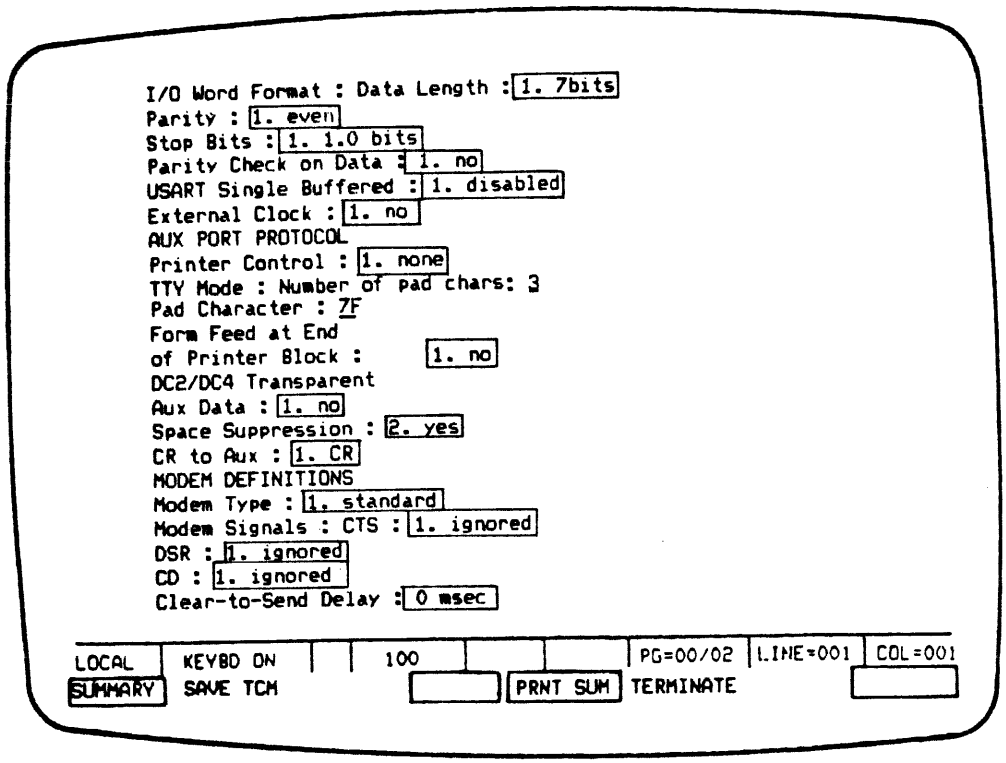


Figure 3-25 Sixth Installation ICM Summary Screen

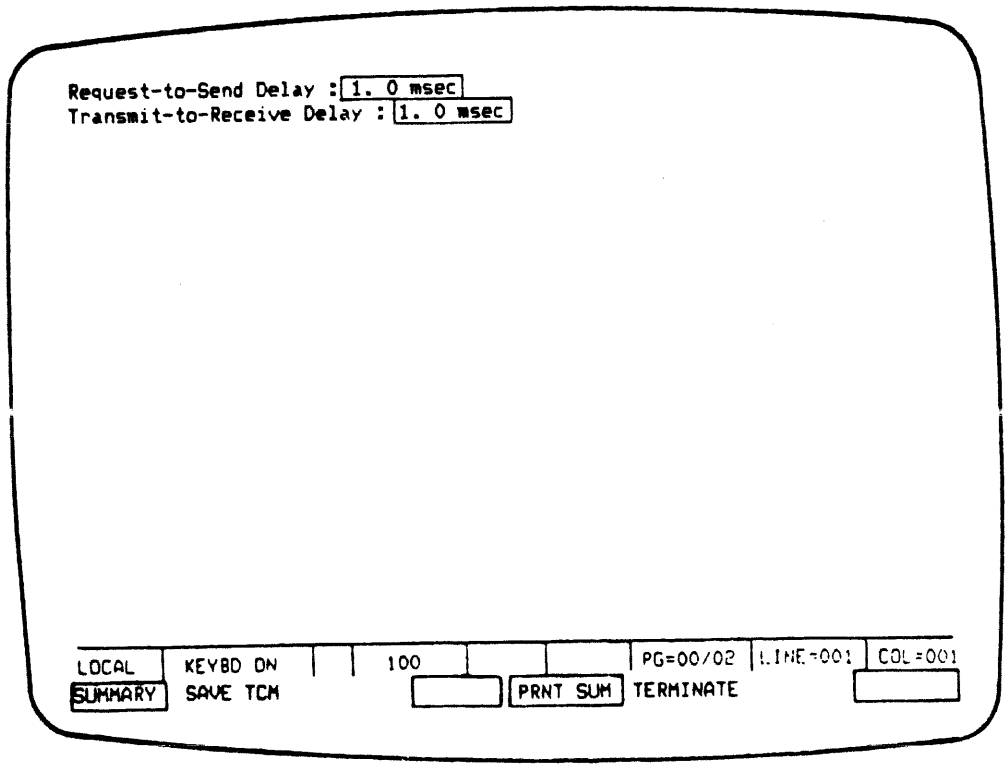


Figure 3-26 Seventh Installation ICM Summary Screen

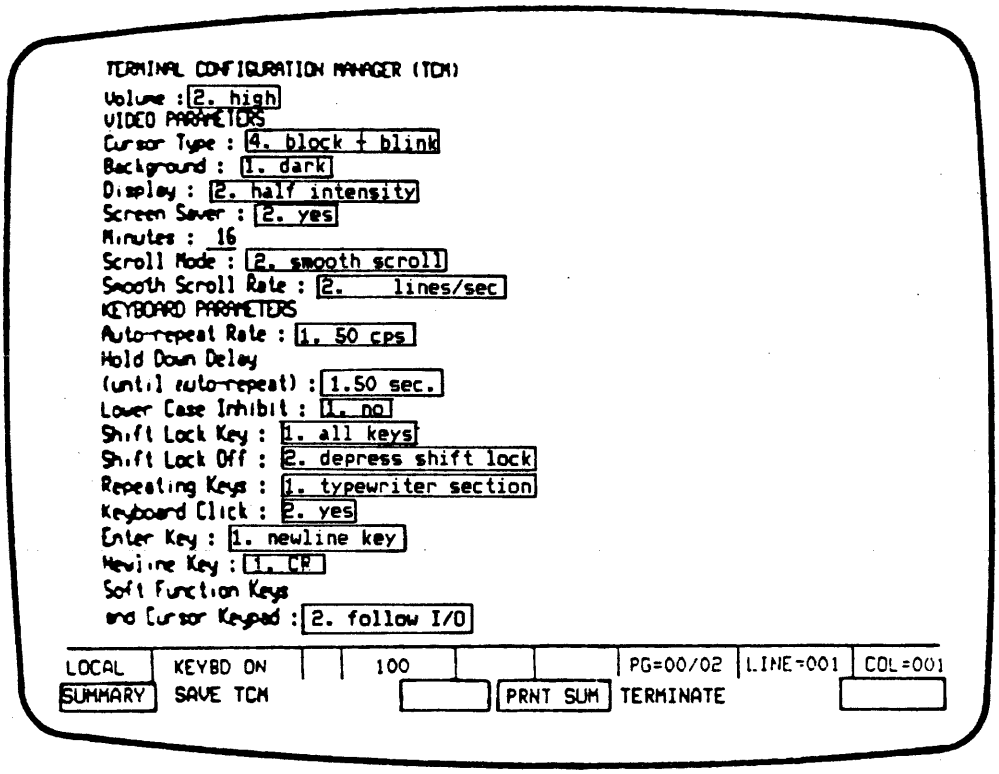


Figure 3-27 First Operator TCM Summary Screen

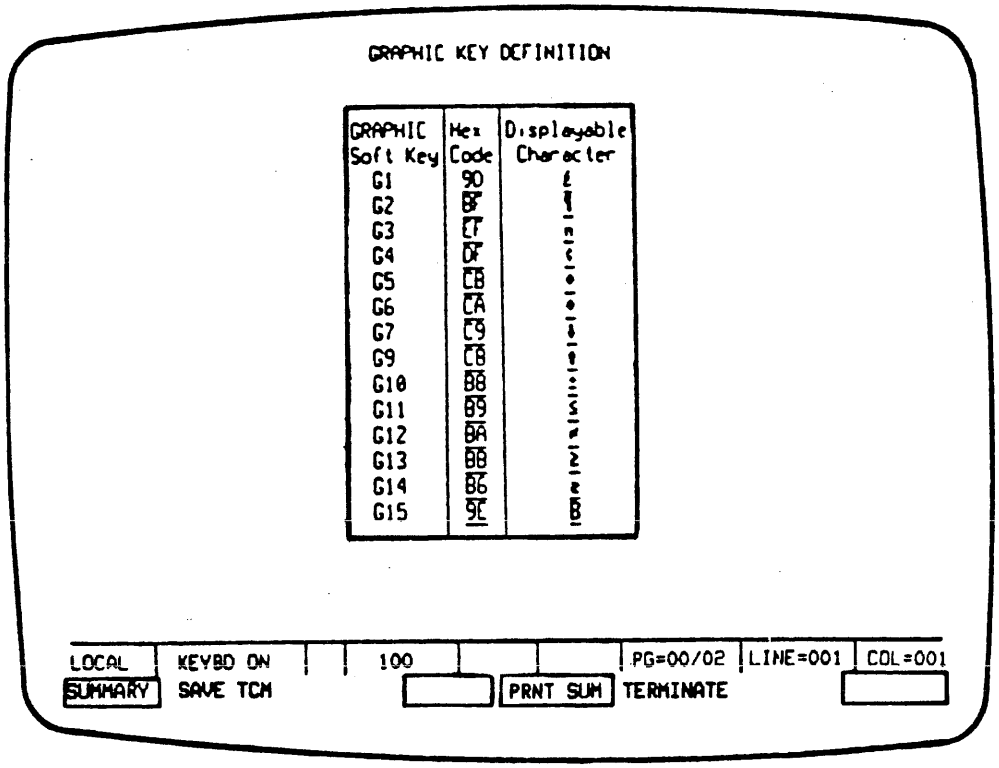


Figure 3-28 Second Operator TCM Summary Screen

SECTION 4. OPERATION

4.1 MODES OF OPERATION

The ATL-008 has three operating modes: ANSI mode, 100 mode, and 52 mode. The terminal powers up in either ANSI mode or 100 mode. This is selected in Installation TCM or by the host and is not normally operator-selectable. The control sequences which choose between 100 mode and 52 mode can be entered either from the keyboard or transmitted by the host computer.

4.1.1 ANSI Mode

In ANSI mode, the ATL-008 recognizes the control sequences defined in the ANSI X3.64-1979 standard. These control sequences are described in Section 6 of this manual. ANSI mode is selected in installation TCM.

4.1.2 100 Mode

In this mode, the ATL-008 emulates a DEC VT100 terminal. It recognizes those control sequences listed in Section 7 of this manual and non-conflicting ANSI X3.64 sequences listed in Section 6. 100 mode is selected in Installation TCM.

4.1.3 52 Mode

A sub-mode of 100 mode, this mode allows the ATL-008 to emulate a DEC VT52 terminal. Different control sequences are recognized by the terminal in 52 mode than in 100 mode. 52 mode is selected by entering a control sequence while in 100 mode.

4.2 USES OF THE WORD "MODE"

The computer industry has generated a wide variety of new terms. One of the most encompassing is **mode**. In this publication, **mode** means many different things. The ATL-008 terminal has two main operating modes -- ANSI mode and 100 mode. In ANSI mode, the terminal supports ANSI 3.64 sequences that pertain to terminals. Some of these sequences control modes defined within the ANSI standard. 100 mode poses a similar problem. 100 mode contains several modes including 52 mode, which in turn defines other modes. Either of the two main operating modes allows use of a softkey which activates monitor mode. In short, it is possible to have modes within modes within modes.

Figure 4-1 generally illustrates this problem. It shows the relationships between several types of modes. It does not account for things like Monitor Mode, which is not limited to any one branch of this tree, nor for modes set in one branch which affect operations in another branch.

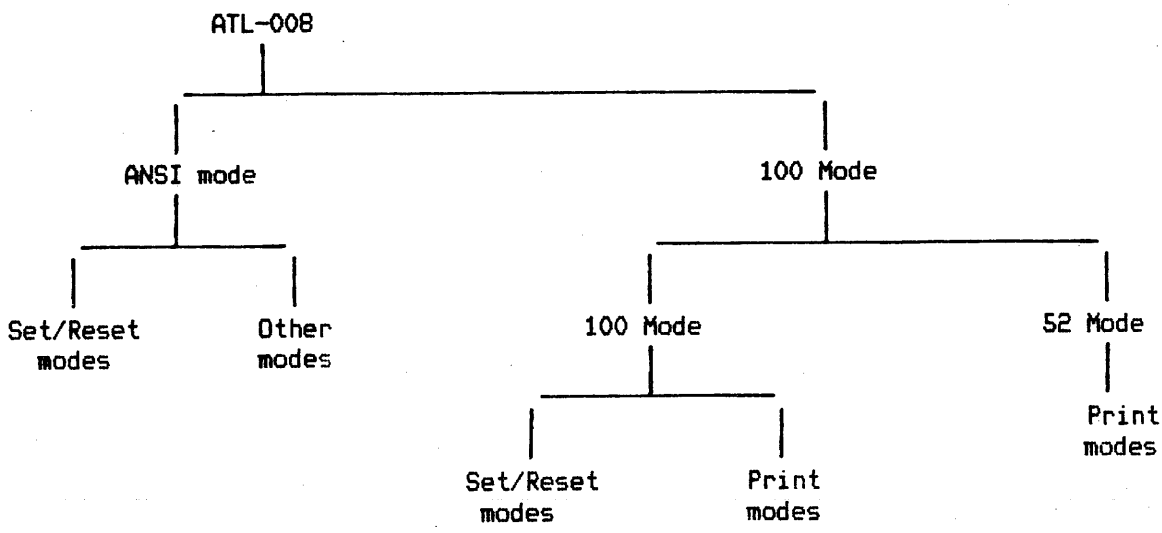


Figure 4-1
ATL-008 Modes

The word mode can cause a lot of confusion. This note is intended to alert the reader to the inevitability of the problem and to discuss the manner of dealing with it.

We cannot use another term because the various people who invented all of these modes insist that they be called modes. For example, we cannot change the name of the ANSI EBM (Editing Boundary Mode), because that is what the standard calls it. Nor can we change the name of DECOM (DEC Origin Mode), because that is what DEC calls it. Since the ATL-008 supports both these pre-defined modes, we have to continue to call them modes as well.

Just knowing about this problem simplifies it to a great degree. It is therefore crucial that the ATL-008 technical user understand that modes are not mutually exclusive when working with this terminal. Using one mode does not necessarily mean that all others are inactive. Different commands may function differently in different modes. For example, the Delete Character (DCH) command is influenced by combinations of Editing Extent Mode (EEM), Editing Boundary Mode (EBM), and Horizontal Editing Mode (HEM). Modes may interact with each other to a considerable degree. For example, Insertion-Replacement Mode (IRM) is dependent on the setting of Editing Boundary Mode (EBM). Sometimes the word mode is the object of a command while at other times it is the subject. For example, Set Mode (SM) is really a command to set a mode and not an actual mode at all. Modes may also interact with the Terminal Configuration Manager (TCM). The DEC private mode, Column Mode (DECCOLM), resets the terminal and changes the TCM display width setting to 132 columns. The ATL-008 also supports five transmission modes between the terminal and the host.

What all this means is that the ATL-008 is a very versatile terminal. It also means that you, the technical user, have to know what you are doing. You have to familiarize yourself with the variety of modes offered and select the correct ones for your application. If something does not work the way you expected, check the modes that may affect it. If it still does not work, check the TCM settings. If all modes and all TCM settings are correct, and it still does not work, it probably cannot be made to work.

4.3 USING THE KEYBOARD

4.3.1 Denoting Multiple Key Sequences

Keys are represented by the names of the key enclosed in a box. A series of keys in normal typeface means that those keys must be entered on the keyboard in the order shown. For example, [ESC [? 1 i] means "press those five keys in succession with no spaces between the keys."

4.3.2 Denoting Simultaneous Key Strokes

The SHIFT key and the CTRL key, when used simultaneously with other keys, perform various functions. When you see either of these keys underlined, it means press either key and hold it down while pressing whatever key is indicated next. For example, [CTRL N] means "hold down the CTRL key while pressing the N key at the same time."

4.3.3 Three Types of Codes

The keyboard produces three main types of codes:

- Displayable characters
- Control codes
- Escape sequences

Displayable Characters -- Displayable character codes are any codes which may be displayed on the screen. This includes the normal ASCII character set and the various alternate character sets described in Appendices A, B, and C. For example: pressing the a key when the terminal is set to display the normal character set causes the displayable character code for the letter [a] to be produced. (Depending on the mode of operation selected, the [a] is transmitted, displayed on the screen, or both.)

Shift Keys -- The shift keys are used to access the characters on the upper portion of the keyboard keycaps or to access upper-case characters.

SHIFT LOCK Key -- The SHIFT LOCK key causes keys in the central (typewriter) section of the keyboard to generate their shifted versions. The specific function of this key is set in Keyboard Parameters in Operator TCM. The message SHIFT appears in the status line to indicate that the SHIFT LOCK key has been pressed. Depending on the setting of Shift Lock

Off in Keyboard Parameters in Operator TCM, either pressing the SHIFT LOCK key again or pressing the SHIFT key causes the typewriter keys to generate their unshifted versions and removes the SHIFT message from the status line.

If the terminal is in A-SHIFT mode, the SHIFT key generates the lower-case version of the alphabetic characters. The non-alphabetic characters are accessed normally.

Control Codes -- The CTRL (CONTROL) key allows access to single-code command sequences. This key functions similarly to the SHIFT key on a standard typewriter: it is pressed and held down while another key is pressed. Multiple-key sequences involving the CTRL key cause specific functions to occur and are only displayed when the terminal is in monitor mode. Control codes occur in two forms: with escape sequences and by themselves.

- When independent, they are generated by pressing and holding down the CTRL key while simultaneously typing the required key. For example, [CTRL G] generates a BEL code and causes the terminal to ring the bell. The CTRL key also allows access to the functions shown on the front face of the keycap.

- When the CTRL key is used with the escape key, it is released after the escape key is typed, before any other keys are pressed. For example, [CTRL ESC c] resets the terminal from the keyboard. In this sequence, the CTRL key is pressed and held down while the escape key is pressed and released, then the CTRL key is released, after which the [c] key is pressed and released. The combination of [CTRL ESC] ensures local execution of the escape sequence even if the terminal is on line. ([CTRL ESC c] does a local [ESC c] in the above example.)

Escape Codes -- Escape codes are displayed only when the terminal is in monitor mode. They are performed by pressing and then releasing the ESC key, then pressing and releasing the designated key(s). For example, to cause subsequent text to be displayed in high intensity, type [ESC [1 m] . To force an escape sequence to be performed locally, press CTRL and hold it down while pressing ESC .

Single Key Escape Functions -- Single keys can also generate escape functions. Examples include programmed function keys and cursor control keys. These keys either perform the specific function or send the appropriate ESC sequence out the port.

Note: If the Soft Function Keys and Cursor Keypad selection in TCM is always local, these functions are always performed locally regardless of the status of the ports. See Keyboard Parameters under Operator TCM in Section 3.

4.4 KEYBOARD

The standard ATL terminal keyboard is divided into four sections: the Function Group, the Cursor-Control Group, the Typewriter Group, and the Keypad Group (see Figure 4-2).

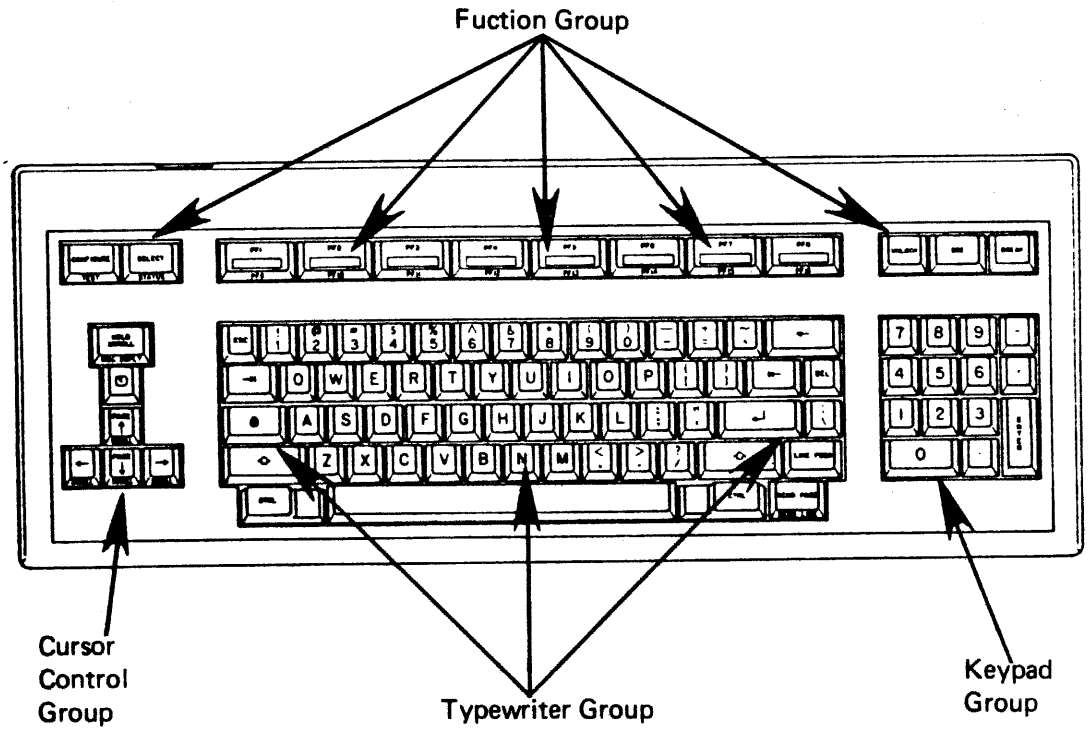


Figure 4-2
Standard ATL Keyboard

4.4.1 Typewriter Group

Most of the keys in the typewriter group generate displayable characters or perform other functions normally found on a typewriter (TAB (⟶|), SHIFT LOCK (🔒), SHIFT (⇧), BACKSPACE (←), BACKTAB (|←), NEWLINE (↵), and LINE FEED). The remainder of this paragraph describes the remaining keys in this group.

ESCAPE -- The function of this key is described in the subsection Using the Keyboard.

CTRL -- The function of the two control keys is described in the subsection Using the Keyboard.

DELETE -- This key transmits the ASCII DEL character (7FH). This key does not delete characters from the screen.

SEND PAGE/SEND LINE -- This key sends the current page of data to the main communications port. Pressing this key simultaneously with the CONTROL key causes transmission of the line (instead of the page) containing the cursor. See Transmission modes under Terminal Configuration in Installation TCM in Section 3 for a description of what is actually sent under various conditions.

4.4.2 Cursor Control Group

HOLD SCROLL/MSG DSPLY -- Pressing the HOLD SCROLL key once stops scrolling on the display; pressing it again allows scrolling to continue.

Pressing this key simultaneously with the CONTROL key alternately displays the CPU message or the status line in Line 26 of the display.

CURSOR UP -- This key normally performs the CUU (cursor up) control (with null parameter). If pressed simultaneously with the SHIFT key, this key performs the PP (preceding page) control (with null parameter). If pressed simultaneously with the CONTROL key, this key performs the SU (scroll up) control (with null parameter).

CURSOR HOME -- This key normally performs a CUP (cursor position) control (with null parameter).

CURSOR LEFT -- Performs the CUB (cursor backward) control (with null parameter). If pressed simultaneously with the SHIFT key, this key performs the SL (scroll left) control (with NULL parameter).

CURSOR DOWN -- This key normally performs a CUD (cursor down) control (with null parameter). If pressed simultaneously with the SHIFT key, this key performs the NP (next page) control (with null parameter). If pressed simultaneously with the CONTROL key, this key performs the SD (scroll down) control (with null parameter).

CURSOR RIGHT -- Performs the CUF (cursor forward) control (with null parameter). If pressed simultaneously with the SHIFT key, this key performs the SR (scroll right) control (with NULL parameter).

4.4.3 Function Group

CONFIGURE/TEST -- This key accesses the Operator Terminal Configuration Manager (TCM). While in TCM, the operator can examine or alter some of the terminal configuration parameters. When the CONFIGURE key is pressed simultaneously with the SHIFT key, the terminal will enter Installation TCM. See Section 3 for additional information about TCM. When the TEST key is pressed (CTRL CONFIGURE/TEST), the terminal displays the display test pattern. For more information, see the description of Display Test in Section 2.

SELECT/STATUS -- If a soft key menu other than the primary soft key menu is displayed, pressing this key causes the next higher level menu to be displayed. Pressing this key simultaneously with the SHIFT key causes the primary soft key menu to be displayed. See Section 5 for details on the operation of the soft function keys. Pressed this key simultaneously with the CONTROL key causes the terminal to display the alternate ANSI status line (if enabled in TCM) as long as the keys are held down.

SOFT_KEYS -- The soft keys are the row of large keys across the top center of the keyboard. Each has three functions. If pressed by itself, the key performs the function described by the label in the bottom row of the display. If pressed with SHIFT, the key acts as a programmed function key, and sends a sequence out the main port. The SHIFTed softkeys send PF1 to PF8. If pressed with CTRL, the key acts as a programmed function key and sends a different sequence out the main port. The CTRL softkeys send PF9 to PF16. Predefined and user-definable softkey functions are described in Section 5.

UNLOCK -- There are two types of keyboard locks: soft and hard. A soft lock results by issuing a Beehive private command for a soft keyboard lock (BEESKL). This key unlocks the soft lock. When the keyboard is soft locked, every key on the keyboard is disabled except the UNLOCK key. A hard lock results from setting Keyboard Action Mode (KAM) or by issuing a Disable Manual Input (DMI) command. When the keyboard is hard locked, every key on the keyboard is disabled, including UNLOCK. The hard lock must be unlocked by either resetting Keyboard Action Mode (KAM) or by an Enable Manual Input (EMI) command from the host. These two methods are equivalent. Cycling the power off and on again will also unlock the keyboard.

START-OF-ENTRY -- This key transmits a special graphic code (▶) which is interpreted as a Start-of-Entry symbol.

BREAK -- When the terminal is on-line, this key normally causes a 250-millisecond spacing signal to be transmitted from the main communications port. When pressed simultaneously with the CONTROL key, this key causes a 500-millisecond spacing signal to be transmitted.

4.4.4 Keypad Group

The keypad group consists of keys for the ten digits 0 through 9, the comma (,), period (.), and hyphen (-), and the ENTER key. The numeric and punctuation keys perform the same function as the corresponding keys in the typewriter group, except that they are not affected by the SHIFT, SHIFT LOCK, or CTRL keys.

ENTER -- See Terminal Configuration in Installation TCM in Section 3.

Keypad Application Mode (100)/Alternate Keypad Mode (52) -- Setting this mode while the terminal is in 100 or 52 mode causes the numeric and punctuation keys on the keypad to generate escape sequences. See Section 7 for more information.

4.5 STATUS LINE

The 26th line of the ATL-008 display is called the status line. See Figure 4-3. The following four different displays can be shown on this line:

Main Status Line
ANSI Mode Status Line
Host Message
Terminal Status Display

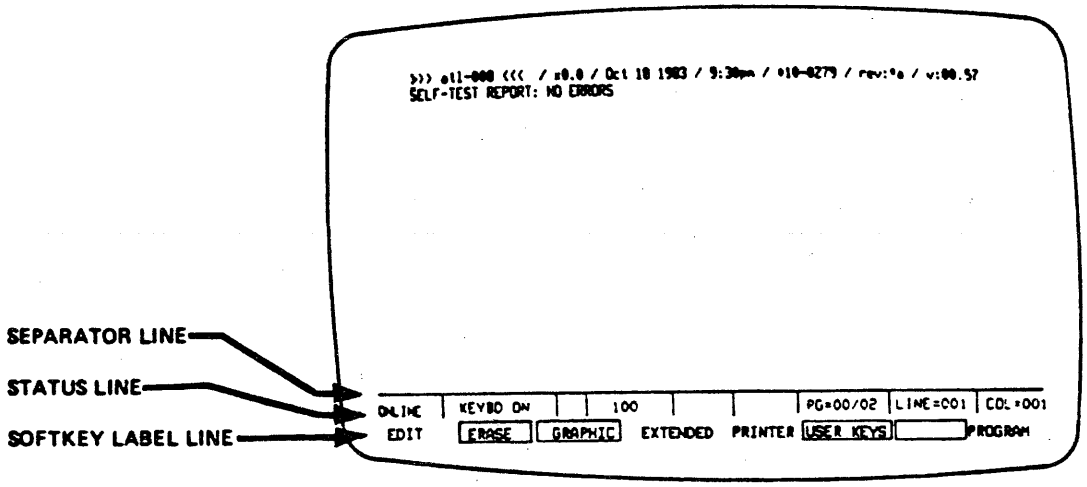


Figure 4-3
Location of Status Line

4.5.1 Main Status Line

The main status line is displayed when the terminal is first turned on. It shows useful information about the status of the terminal: what mode the terminal is in, the status of the I/O ports, status of the keyboard, whether the keyboard is shifted or shift-locked, messages, and other information depending on the current mode of the terminal.

The main status line displays in normal video with some messages appearing in reverse video. As shown in Figure 4-4, it is divided into nine fields, labeled A through I.

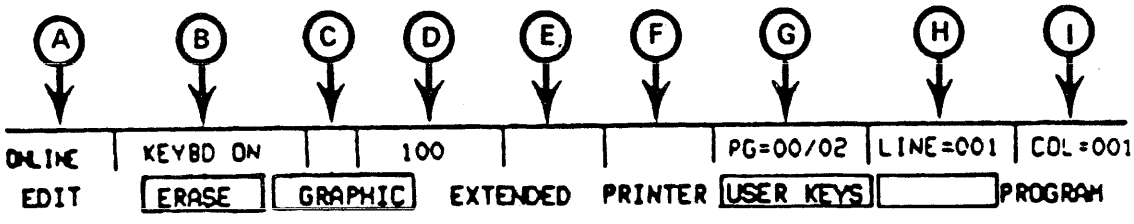


Figure 4-4 Main Status Line

The messages that appear in these fields in descending order of precedence are:

Field_A

MSG WAIT	A CPU message is awaiting display.
ONLINE	The terminal is on line.
LOCAL	The terminal is off line.

Field_B

KEYBD ON	The keyboard is active.
KEYBD LOCK	The keyboard is locked.

Field_C

	Blank field. DSR ignored is selected in TCM. (See Modem Definitions in Installation TCM in Section 3.)
DSR	Field blinks in reverse video. DSR required is selected in TCM and the terminal is not receiving a DSR signal.
DSR	Normal video. DSR required is selected in TCM and the terminal is receiving a DSR signal.

Field_D

A-SHIFT	The keyboard is in alpha-shift mode. (See the Shift Lock Key setting under Keyboard Parameters in Operator TCM.)
SHIFT	The keyboard is in shift mode.
	Blank field. The terminal is in ANSI mode.
100	The terminal is in 100 mode.
52	The terminal is in 52 mode.

Field_E

HOLD	The HOLD SCROLL key has been pressed, temporarily interrupting data transmission from the host. See the description of the HOLD/SCROLL key.
AUX BSY	Communications between the terminal and the aux port device has been enabled, but the aux port device is not ready.
AUX ENB	Bidirectional communication between the terminal and the auxiliary port is enabled. Keyboard data are sent out the aux port. Data received from the aux port device are processed locally.
AUX ON	Communication between the terminal main port and the auxiliary port is enabled. Data received by the terminal is sent to the aux port device.
AUX RDY	Communications between the terminal and the aux port device has been enabled, and the aux port device is ready.

Note: See the BEECM sequence for the interactions of these conditions.

Field_E

If the terminal is in ANSI mode:

	No logical attributes have been set for the current field.
PROTECT	The current field is protected.
CONSTANT	The current field contains constant data.
UNPROTECT	The current field may be altered from the keyboard.
ALPHABETIC	Any data entered in the current field must be letters.
NUMBERS/SYMB	Data entered in the current field must consist of numbers and punctuation marks.
NUMBERS	The current field will accept numbers only.
ALPHA	The current field will accept letters and numbers.
GRAPHIC	The current field will accept any printable character.
MUST ENTER	The current field must contain some data before the cursor can exit the field.
MUST FILL	The current field must be full of data before the cursor can exit the field.
USER DEF	The field has been defined by the user's bootloaded code.

If the terminal is in 100 mode:

	No LEDs are on.
Ln	LED n is on.

Field_G

[ANSI] PG=xx/yy	An ANSI mode has changed. The terminal is on page xx of yy. yy is defined in Screen Page Allocations in Installation TCM.

Field_H

LINE=xxx	The active position is in line xxx.

Field_I

COL=yyy	The active position is in column yyy.

4.5.2 ANSI Mode Status Line

There are twenty modes defined in the ANSI X3.64 standard. With the exception of SEM (Select Editing Extent Mode), which has four states, each of these modes is a two-state condition. The two states are called set and reset.

As the following list shows, the current status of twelve of these modes is found in the ANSI mode status line. The ANSI mode status line is an alternate status line enabled in TCM. It is displayed by pressing the CTRL key and holding it down while pressing the SELECT/STATUS key. The ANSI mode status line is shown in Figure 4-5.

The ANSI modes are:

Mode		Where Found
DEVICE SENDING MODES		
FETM	Format Effector Transfer Mode	ANSI mode status line
GATM	Guarded Area Transfer Mode	ANSI mode status line
MATM	Multiple Area Transfer Mode	ANSI mode status line
SATM	Selected Area Transfer Mode	ANSI mode status line
SRTM	Status Reporting Transfer Mode	ANSI mode status line
TTM	Transfer Termination Mode	ANSI mode status line
DEVICE LOCAL AND RECEIVING MODES		
EBM	Editing Boundary Mode	Terminal Status Display
ERM	Erasure Mode	ANSI mode status line
FEAM	Format Effector Action Mode	ANSI mode status line
HEM	Horizontal Editing Mode	ANSI mode status line
IRM	Insertion -- Replacement Mode	INS CHAR Softkey
PUM	Positioning Unit Mode	Not used
SEM	Select Editing Extent Mode	ANSI mode status line
TSM	Tabulation Stop Mode	ANSI mode status line
VEM	Vertical Editing Mode	ANSI mode status line
DEVICE LOCAL MODES		
CRM	Control Representation Mode	MONITOR Softkey
KAM	Keyboard Action Mode	Main Status Line
SRM	Send-Receive Mode	TCM Setting
SENDING, RECEIVING, AND LOCAL MODES		
LNM	Line Feed New Line Mode	TCM Setting

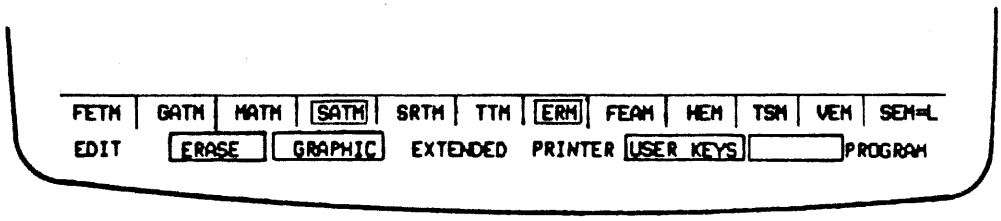


Figure 4-5
ANSI Mode Status Line

The ANSI mode status line is displayed as long as one of the keys is held down. When both the CTRL and the SELECT/STATUS keys are released, the display changes back to what it was previously.

Each of the twelve modes has its own field in the ANSI mode status line. With the exception of SEM, which is a four-state condition, the modes are displayed in normal video if reset and in reverse video if set. The current state of SEM is displayed as follows:

Display	Setting
SEM=D	Edit in Display
SEM=L	Edit in Line
SEM=F	Edit in Field (between horizontal tab stops)
SEM=Q	Edit in Qualified Area (as defined by SPA, EPA, and/or DAQ)

4.5.3 Host Message

A host can use the DCS control sequence to send a message to the terminal. When a host message arrives, MSG WAIT (reverse video and blinking) appears in Field A of the status line. Pressing CTRL HOLD SCROLL/MSG DSPLY causes the host message to be displayed in reverse video on Line 26 (Figure 4-6).

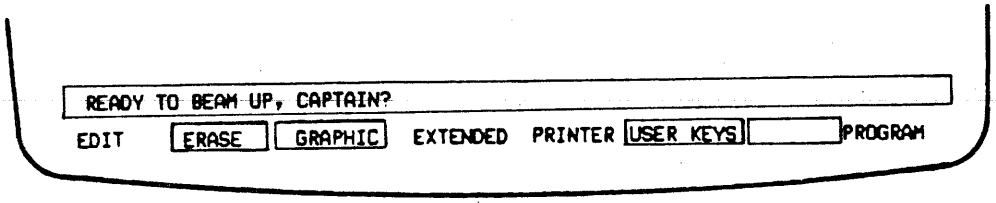


Figure 4-6
Typical Host Message

Pressing CTRL HOLD SCROLL/MSG DSPLY again returns Line 26 to its former state.

Alternate Status Line -- The host may send a message to the terminal which will be displayed automatically, without waiting for the operator to press the MSG DSPLY key. This feature allows the host to display an alternate status line. The user may display the original status line by pressing CTRL HOLD SCROLL/MSG DSPLY . An alternate status line may also be created by defining a logical page that occupies this area.

4.5.4 Terminal Status Display

When terminal status is requested by the appropriate DCS sequence, a string of characters is displayed in reverse video on Line 26. Each of these characters represents a status or error condition of the terminal (Figure 4-7).

The meaning of the various characters in the Terminal Status Display is given in Table 4-1.

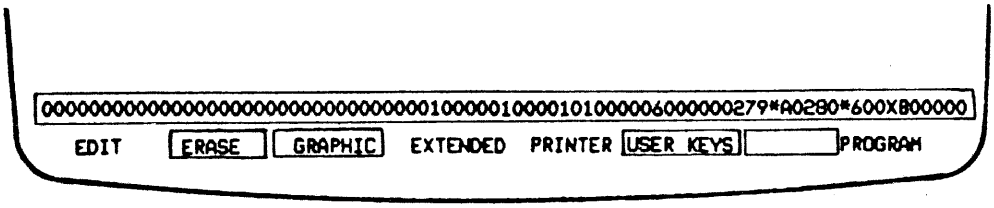


Figure 4-7
Terminal Status Display

Table 4-1
Status Message Character Interpretation

POSITION	DESCRIPTION OF FEATURE	CHARACTER DISPLAYED
	CONFIDENCE TEST RESULTS	
1	ROM/RAM	ASCII 0-3
2	RESERVED	ASCII 0
3	NON-VOLATILE MEMORY	ASCII 0-3
4	RESERVED	ASCII 0
5	TIMERS	ASCII 0-3
6	KEYBOARD	ASCII 0-7
7	RESERVED	ASCII 0
8	RESERVED	ASCII 0
9	RESERVED	ASCII 0
10	RESERVED	ASCII 0
	MAIN PORT ERRORS	
11	COMMUNICATIONS ERROR COUNT	ASCII 0-F
12	BUFFER OVERRUN ERROR COUNT	ASCII 0-F
	AUX PORT ERRORS	
13	COMMUNICATIONS ERROR COUNT	ASCII 0-F
14	BUFFER OVERRUN ERROR COUNT	ASCII 0-F
	KEYBOARD PORT ERRORS	
15	COMMUNICATIONS ERROR COUNT	ASCII 0-F
16	BUFFER OVERRUN ERROR COUNT	ASCII 0-F
17	UNDEFINED COMMAND COUNT	ASCII 0-F
18	POWER UP, SELFTEST INDICATION	ASCII 0 or 1
19,20	RESERVED	ASCII 00
21	RESERVED	ASCII 0
22	RESERVED	ASCII 0
23	RESERVED	ASCII 0

POSITION	DESCRIPTION OF FEATURE	CHARACTER DISPLAYED
	TERMINAL MODES	
24	MAIN PORT ONLINE/LOCAL	ASCII 0 or 1
25	GRAPHICS MODE ON	ASCII 0 or 1
26	RESERVED	ASCII 0-3
27	HARD/SOFT KEYBOARD LOCK	ASCII 0-3
28	PROGRAM OR FORMS BUILD MODE	ASCII 0 or 1
29	ANSI CRM	ASCII 0 or 1
30	ANSI EBM	ASCII 0 or 1
31	ANSI ERM	ASCII 0 or 1
32	ANSI FETM	ASCII 0 or 1
33	ANSI GATM	ASCII 0 or 1
34	ANSI HEM	ASCII 0 or 1
35	ANSI IRM	ASCII 0 or 1
36	ANSI KAM	ASCII 0 or 1
37	ANSI LNM	ASCII 0 or 1
38	ANSI MATM	ASCII 0 or 1
39	ANSI PUM	ASCII 0 or 1
40	ANSI SATM	ASCII 0 or 1
41	ANSI SEM	ASCII 0 -- 3
42	ANSI SRM	ASCII 0 or 1
43	ANSI SRTM	ASCII 0 or 1
44	ANSI TSM	ASCII 0 or 1
45	ANSI TTM	ASCII 0 or 1
46	ANSI VEM	ASCII 0 or 1
47	RESERVED	ASCII 0
48	RESERVED	ASCII 0
49	RESERVED	ASCII 0
	STATUS LINE	
50	DISPLAYED STATUS LINE	ASCII 0-F
51	CPU MESSAGE WAIT	ASCII 0 or 1
52	RESERVED	ASCII 0
53	SOFT KEY MENU DISPLAYED	ASCII 0-F
54	AUX PORT STATUS	ASCII 0-7
55	RESERVED	ASCII 0
56	RESERVED	ASCII 0
57	RESERVED	ASCII 0
58	RESERVED	ASCII 0

POSITION	DESCRIPTION OF FEATURE	CHARACTER DISPLAYED
	TERMINAL IDENTITY	
59-62	TERMINAL SOFTWARE CONTROL NUMBER	VARIOUS
63, 64	TERMINAL REVISION NUMBER	VARIOUS
65-68	KEYBOARD SOFTWARE CONTROL NUMBER	VARIOUS
69, 70	KEYBOARD REVISION NUMBER	VARIOUS
71, 72	LOGIC BOARD REVISION LEVEL	VARIOUS
73-80	RESERVED	VARIOUS

The various fields have the following meanings:

CONFIDENCE TEST RESULTS -- The first 8 characters of the status message indicate the results of the confidence tests. The following explains the received report.

In all cases below, an ASCII 0 is an indication that no errors occurred.

Character 1 is the RAM and ROM report:
- 1 a RAM error
- 2 a ROM error
- 3 both a RAM and a ROM error.

Character 2 is not used.

Character 3 is the non-volatile memory report:
- 1 a CRC error in the non-volatile memory

all others are reserved for future use.

Character 4 is not used.

Character 5 is the timer report. At the present time these are the only timers reported:
- 1 a video timer error
- 2 a baud rate timer error
- 3 a failure in both timers.

Character 6 is the keyboard report:
- 1 a RAM error
- 2 a ROM error
- 4 a communications error.

Any combination of these errors is reported by adding the error numbers together and reporting the sum. For example, if a RAM and a communications error occurred, a 5 would be reported. If an error occurs in all three tests, a 7 will be reported.

Characters 7 and 8 are not used.

ERROR CONDITIONS AND LOGGING -- Characters 9 through 16 comprise the Error Logging Report. In all cases where this report consists of a count, the

report consists of an ASCII character 0 through 9 or A through F. This allows logging of up to fifteen errors. If more than fifteen errors occur, the count remains at fifteen until cleared by a special escape sequence. This escape sequence clears all error counts to an ASCII 0, which indicates no errors.

Character 11 is the main port error count. This character is a count of communications errors. Parity, framing, and overrun errors are logged.

Character 12 is the main port buffer overrun error count. This count is incremented every time the main port receiver buffer is overrun.

Character 13 is the auxiliary port error count. This character is a count of communications errors. Parity, framing, and overrun errors are logged .

Character 14 is the auxiliary port buffer overrun error count. This count is incremented every time the auxiliary port receiver buffer is overrun.

Character 15 is the keyboard port error count. This character is a count of communications error between the keyboard and the terminal. Parity, framing, and overrun errors are logged.

Character 16 is the keyboard buffer overrun error count. This count is incremented every time the keyboard port receiver buffer is overrun.

Character 17 is the undefined command or parameter errors count. This is a count of the number of escape sequence errors that occur.

Character 18 is the power-up, selftest indication. This character indicates that power to the terminal has been interrupted since this indication was last reset. Performing the selftest diagnostics also sets this flag. An ASCII 1 indicates that power has been interrupted.

If Status Reporting has been enabled in Main Port Protocol in Installation TCM, at the beginning of the Confidence Tests, the terminal will send an [ESC, n, 2] to the host, indicating that the terminal is busy. At the end of the Confidence tests, the terminal will send an [ESC, n, 0]. These messages are sent anytime the Confidence tests are performed including those tests performed upon power-up. These sequences are compatible with the ANSI X3.64 Device Status Report (DSR).

TERMINAL MODES -- Character positions 24 through 47 indicate whether various terminal modes are on or off. In each case an ASCII 0 indicates the specific mode or feature is off.

Character 24 is the main port ONLINE or LOCAL indication. An ASCII 0 indicates that the terminal is in LOCAL mode; a 1 indicates that the terminal is in ONLINE mode.

Character 25 is the GRAPHICS mode indication. A 1 indicates GRAPHICS mode is on.

Character 26 is reserved for future use.

Character 27 is the KEYBOARD LOCK indication. A 1 indicates a soft keyboard lock. A soft keyboard lock is resettable by the operator by pressing the UNLOCK key. A 2 indicates a hard keyboard lock. Only the host can reset a hard keyboard lock. A 3 indicates that both keyboard lock modes are on.

Character 28 is the PROGRAM indication. This mode allows the host or the user to program protected and unprotected areas on the screen. A 1 indicates that this mode is on.

Characters 29 through 47 indicate the status of the ANSI modes. A 0 indicates that the mode is reset; a 1 indicates that the mode is set.

Character 50 is the status line display indication. This character indicates which of several possible message lines appear in the status line. The possible message lines have not been defined as of this time; the meaning of each character is not assigned yet.

Character 51 indicates that a CPU message, sent by the host, is waiting to be examined by the user.

SOFT KEY MENU DISPLAY -- Character position 53 indicates which soft key menu is currently being displayed. The ASCII character that is returned corresponds to the parameters of the BEESSKM command.

AUX PORT STATUS -- Character 54 is the aux port status indication. A 1 in this position indicates that the port is busy. A 2 indicates that the terminal is in aux on mode. This mode sends all data from the keyboard to the aux port. A 4 indicates the port is in aux enable mode. This mode sends all data received from the main port to the aux port. If the terminal is on-line as well, this same data is also sent to the screen of the terminal.

An indication of any combination of these modes is available as well. For example, if the aux port is in aux enabled mode and the aux port device is busy, the character in this position will be a 5.

TERMINAL IDENTIFICATION -- Character positions 59 through 72 display the various version and revision numbers that are used to identify the terminal.

Software control numbers are made up of four digits, 0 through 9. Revision levels are indicated by letters. Spaces are not used as part of version or revision number. Unused spaces are filled with periods (.). For example: 0000.A.

Character positions 59 through 62 are the last four characters of the software control number for the terminal firmware.

Character positions 63 and 64 indicate the revision level of the terminal firmware.

Character positions 65 through 68 are the last four characters of the software control number for the keyboard firmware.

Character positions 69 and 70 indicate the revision level of the keyboard firmware.

Character positions 71 and 72 indicate the revision level of the terminal logic board. This is available by reading a strapping block on the terminal logic board. This strapping is changed whenever the revision of the terminal logic board is changed.

Restoring the Main Status Line -- To remove the terminal status display, press any key in the typewriter portion of the keyboard.

4.5.5 Disable Status Line

To disable the main status line, press CTRL SHIFT SELECT/STATUS. A blank line will be displayed in Line 26. Pressing CTRL SHIFT SELECT/STATUS again restores the status line.

While the main status line is disabled, other displays that use Line 26 operate as follows:

- The ANSI mode status line cannot be displayed until the main status line is enabled.
- Host messages can be displayed by pressing CTRL HOLD SCROLL/MSG DSPLY.
- The terminal status display can be displayed by entering the proper DCS sequence.
- The user may define pages that use line 26.

SECTION 5. SOFTKEYS, USER DEFINED SOFTKEYS, AND PROGRAM FUNCTION KEYS

Softkeys, User-Defined Softkeys, and Program Function keys all use the eight large keys at the top of the keyboard immediately above the typewriter section of keys. Each keycap has three legends indicating the normal, shifted, and control functions of the key. The shifted and control functions are on the top and front of the key. These functions are all Programmed Function keys or PF keys. The normal key legend on all of these keys is a rectangle. Each rectangle indicates that the normal key operation is given in a corresponding area on the bottom line of the screen. They are called sofikeys because their functions change depending on the current mode or operation. The areas on the screen that describe the softkeys are called softkey labels. The colors of the keys correspond to the light and dark labels.

5.1 SOFTKEYS

Softkeys provide access to many intrinsic functions of the ATL-008 terminal. Pressing a particular softkey may lead to another level of softkeys in a sort of inverted tree structure. For example, the EDIT softkey leads to another set of softkeys labeled MODIFY, ERASE, INS CHAR, PRINTER, USER KEYS, INS LINE, and DEL LINE. This is considered to be a level lower than the set of softkeys containing the EDIT softkey. Some softkeys perform a definite function and do not lead to another level. For example, the INS CHAR softkey puts the terminal in insert character mode. The softkey label blinks to indicate that the insert function is active. Pressing the INS CHAR key again deactivates the function and the label stops blinking. Other softkeys activate a function and change the corresponding softkey label, but do not change levels. For example, the LOCAL softkey takes the terminal off line, displays a LOCAL message in the status line and changes the softkey label to ON-LINE so that the next time the key is pressed, it will put the terminal on line, put a message in the status line and change the softkey label back to LOCAL.

NOTE: It is important to realize that ANSI modes may affect softkey functions. Horizontal Editing Mode (HEM), Vertical Editing Mode (VEM), and Editing Boundary Mode (EBM) generally have the greatest effect, particularly on softkey editing functions.

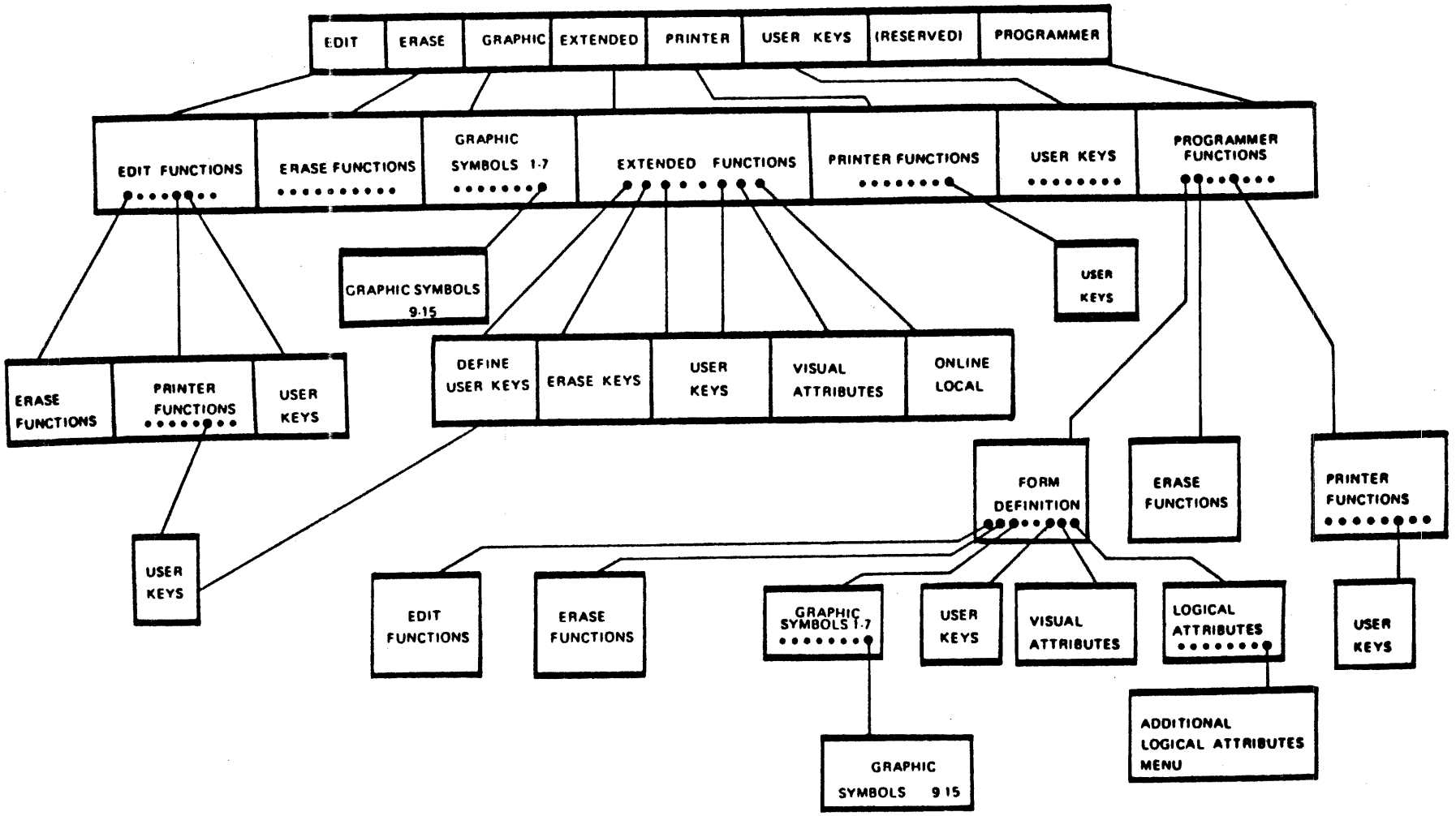


Figure 5-1 Softkey Organization

The SELECT key (next to the CONFIGURE KEY at the top left of the keyboard) is used to go from one level of softkeys to the next higher level. For example, if the current softkeys are MODIFY, ERASE, INS CHAR, PRINTER, USER KEYS, INS LINE, and DEL LINE, then SELECT will again display the EDIT softkey and its parallel branches in the tree. In this context, SELECT acts as an un-do key; however, any functions activated while in a lower level will remain active after returning to the next higher level. SHIFT SELECT will display the highest level of softkeys.

There is one primary set of softkeys which leads to the secondary softkeys organized in a modified inverted tree structure. It is not a true tree structure because it is sometimes possible to move from branch to branch without going to a previous junction. For example, the PRINTER softkey and the softkeys that it accesses are available from several different levels. Figure 5-1 illustrates the organization of the softkeys. If you like ADVENTURE, you'll love softkeys.

5.1.1 Primary Level Softkeys

The primary level softkeys are illustrated in Figure 5-2. This is the first level of softkeys displayed whenever the terminal is reset unless the default level has been changed. It is called the primary level because it leads to all other levels.

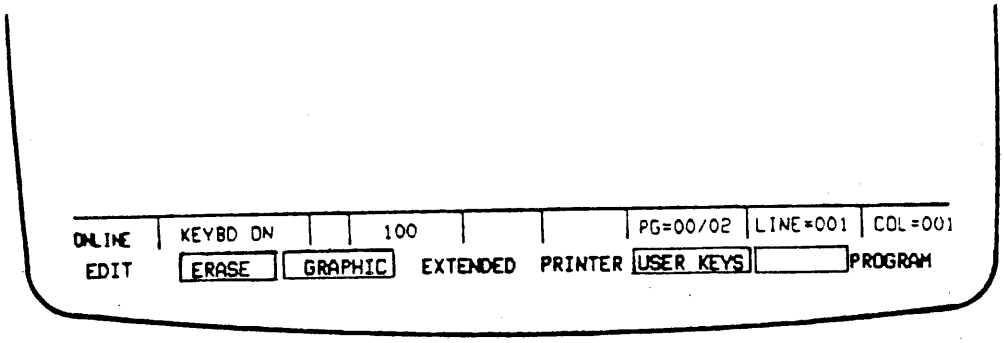


Figure 5-2 Primary Level Softkeys

EDIT -- This softkey leads to the softkeys which perform editing functions on text in display memory. Editing operations include: modify/clear memory, erase, insert/delete character, activate printer, insert/delete line, and user keys functions.

ERASE -- This softkey leads to the softkeys which erase various positions of the current page of memory. Erase operations include: clear entry, erase unprotected, erase from beginning of line or page, erase to end of line or page, and erase all.

GRAPHIC -- This softkey leads to the softkeys which produce the currently defined graphics characters. There are fourteen symbols available. This selection may be modified by Graphic Key Definition in Operator TCM. This graphic function also enables line drawing capabilities using the Shift and Cursor Control keys.

EXTENDED -- This softkey leads to an extended set of softkeys which in turn lead to more functions. Extended function menu provides the following features: define user keys, erase, save a form, user keys access, set visual attribute, and local or on-line mode.

PRINTER -- This softkey leads to the softkeys which control the printer. Printer commands include: print line, page, or unprotected, aux enable or on, printer cancel, and user-defined keys.

USER KEYS -- This softkey displays the labels of the user-definable softkeys and permits their use. The user may define keys to transmit a message to the host or to display a form or collection of characters on the screen.

PROGRAM -- This softkey leads to a set of softkeys of special use to programmers. Such functions include forms definition, erase, transmit a form, line monitor mode, printer control. Access to this function may require a password or simultaneous use of the shift and control keys as selected in Initial Configuration in Installation TCM. If access has been restricted, no notification is given.

The remainder of this section describes these functions in detail in the above order. Once you are familiar with the various functions, the map in Figure 5-1 becomes much clearer.

5.1.2 Secondary Level Softkeys

EDIT FUNCTIONS

EDIT softkeys are displayed in Figure 5-3. The edit functions allow various editing operations on text in display memory.

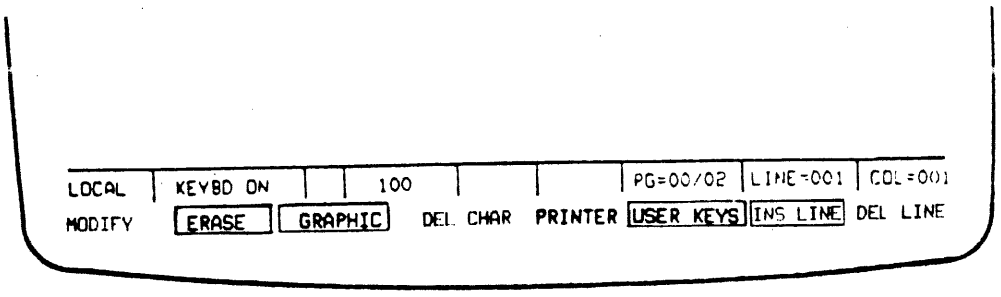


Figure 5-3 Edit Softkeys

MODIFY/CLEAR_ENTRY -- This softkey label depends on the Transmission Mode as selected in Terminal Configuration in Installation TCM. When the terminal is in character transmission mode, the softkey label is MODIFY. When the terminal is in a block transmission mode the label is CLR_ENTRY. This key enables modify mode while the user is in an interactive mode with a host computer. For example, if the user is in character mode and transmits an erroneous command string to the host computer and receives an error message in response, the user can enable modify mode, correct the error in the command string on the screen, and re-transmit the command without having to retype the entire string. To enable modify mode, press the softkey associated with this label. The label will blink to indicate that the modify mode is enabled. Move the cursor to the line to change and use the Edit Functions to make the desired changes. After changing the line, press ENTER or RETURN to transmit the changed line to the host computer and disable modify mode. The current line (marked by the cursor) is transmitted to the computer and the label stops blinking. As the line is transmitted to the host, it is re-written to the position where the cursor was when modify mode was enabled.

To disable modify mode (before either ENTER or RETURN is pressed), press the function key associated with the MODIFY screen label. The label stops blinking and modify mode is disabled.

CLEAR_ENTRY is displayed when the terminal is in block mode. This function causes the total erasure of the current unprotected field and moves the cursor to the first location of that field. Move the cursor to any character within the field to be erased and press the softkey associated with this label.

ERASE -- Displays choices for erasing various positions of the current page. Erase operations include: clear entry, erase unprotected, erase from beginning of line or page, erase to end of line or page, and erase all.

INS CHAR -- This function enables insert character mode. Characters may be inserted on either a line or page basis. Either of the insert character modes are disabled by simply pressing the softkey associated with this label, or by transmitting data. The label field is returned to normal and you cannot insert new characters. Any characters entered when insert character mode is disabled overwrite existing characters in the line. The action of this key may be affected by the setting of ANSI Modes VEM and HEM.

Insert-Character-in-Line - This is the initial condition. Move the cursor to the character position where you wish to insert characters, press the key associated with this label, and type the characters. The label will blink to indicate that the insert character mode is enabled. While inserting characters, any existing characters are shifted to the right one character position within the same line for each new character entered. Any characters shifted beyond the right margin setting are lost. No wraparound occurs.

Insert-Character-in-Page - If the control and shift keys are pressed together with this softkey, the label changes to INSERT PG. Pressing the new softkey activates the function. It is similar to insert-character-in-line except that wraparound occurs within the page. While inserting characters, any existing characters are shifted to the right and down one character position within the page for each new character entered. Any character shifted beyond the last position of the page is lost.

DEL_CHAR -- This function enables delete character mode. Characters may be deleted on either a line or page basis. The action of this key may be affected by the setting of ANSI Modes VEM and HEM.

Delete-Character-In-Line - This function deletes one character marked by the cursor. Move the cursor to the character to be deleted and press the softkey associated with this label. The character is erased from memory. Remaining characters within the same line are shifted to close the gap.

Delete-Character-In-Page - If the control and shift keys are pressed together with this function key, delete-character-in-page will be enabled. Move the cursor to the character to be deleted and press the softkey associated with this label. The character is erased from memory. Remaining characters within the page are shifted to close the gap.

PRINTER -- Displays a set of softkey labels used to copy data from display memory to a remote printer. Printer commands include: Print line, page or unprotected data, aux enable, aux on, printer cancel, and user defined keys.

USER_KEYS -- Displays a set of user defined function key labels that may be used to store user defined data. The user may define these keys to transmit a message to the host, or to display a form or collection of characters on the screen.

INS_LINE -- This function inserts a blank line into display memory at the line marked by the cursor. Move the cursor to the line where a new line is needed and press the softkey associated with this label. A new blank line is inserted into memory and the cursor position remains unchanged. The action of this key may be affected by the setting of ANSI Mode VEM.

DEL_LINE -- This function deletes an entire line at the current cursor position. Move the cursor to any character position within the line to be deleted and press the softkey associated with this label. The entire line is deleted from display memory. The cursor position is unchanged. The action of this key may be affected by the setting of ANSI Mode VEM.

ERASE FUNCTIONS

The erase functions allow the user to erase various portions of the page. Once the erase selection has been made and the function has been executed, the terminal returns to the calling menu. The ERASE functions keys are illustrated in Figure 5-4.

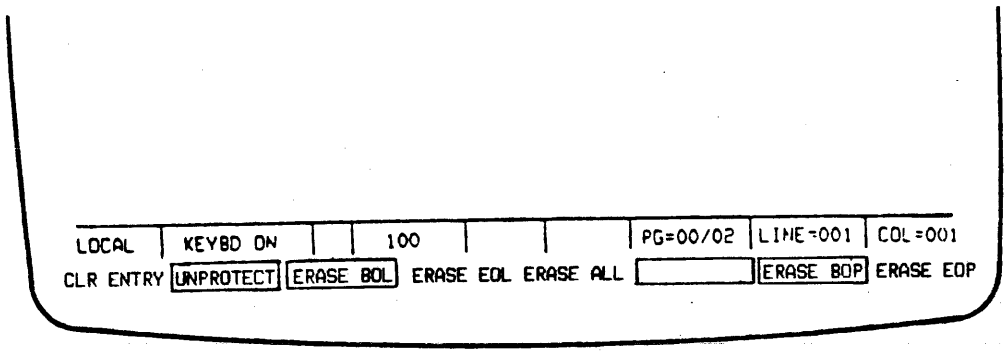


Figure 5-4 ERASE Softkeys

CLR_ENTRY -- This function causes the total erasure of the current unprotected field and moves the cursor to the first location of that field. Move the cursor to any character position within the field and press the softkey associated with this label.

UNPROTECT -- This function erases all unprotected fields on the page and then moves the cursor to the first character of the first unprotected field of the page.

ERASE_BOL -- This function erases all characters from the beginning of the line up to and including the cursor position.

ERASE_EOL -- This function erases all characters from and including the cursor position to the end of the line.

ERASE_ALL -- This function erases all data on the page, including area markers and attributes.

ERASE_BOP -- This function erases all data from the beginning of the current page up to and including the cursor position. Data that does not show on screen (on the page, but outside of the window) may be erased. If there is an area marker (SOE), the erase function will begin at the marker or beginning of the field.

ERASE_EOP -- This function erases all data from and including the cursor position to the end of the displayed page. Data that does not show on the screen (on the page, but outside of the window) may be erased. If there is an area marker (SOE), the erase function will end at the marker or end of the field.

GRAPHIC FUNCTIONS

The Graphics Functions allow the user to display fourteen different graphic characters as specified in Graphic Key Definition in Operator TCM. Figure 5-5 illustrates a typical set of graphics softkeys. The NEXT softkey is used to alternate between the two groups of Graphics keys. Each group contains seven characters and the NEXT softkey.

While this set of softkeys is displayed, the cursor control keys may be used to draw boxes using the standard line-drawing symbols. When the cursor control keys are held down along with the SHIFT key, the terminal generates the line-drawing symbols. This includes the corners, crosses, and junctions; no special key sequences for these cases are required.

The cursor control keys are used to draw the horizontal and vertical lines. When they intersect one another, the terminal will insert the correct junctions or corner, depending on the cursor's direction of travel. The line drawing symbols use the attribute of the field in which they are located.

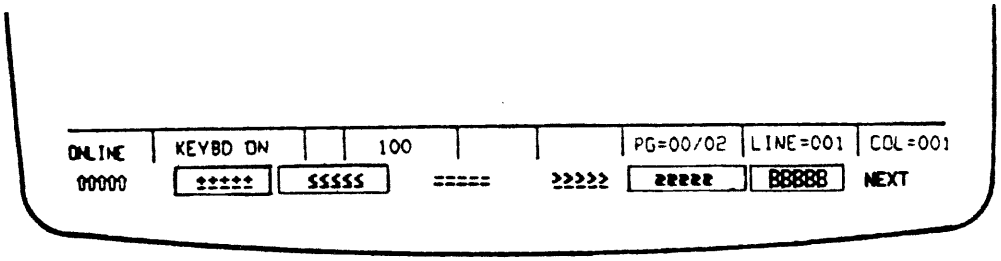


Figure 5-5 Graphics Softkeys

EXTENDED FUNCTIONS

The EXTENDED softkey in the primary level results in a second set of softkeys that provide extended terminal functions. The softkey labels are shown in Figure 5-6.

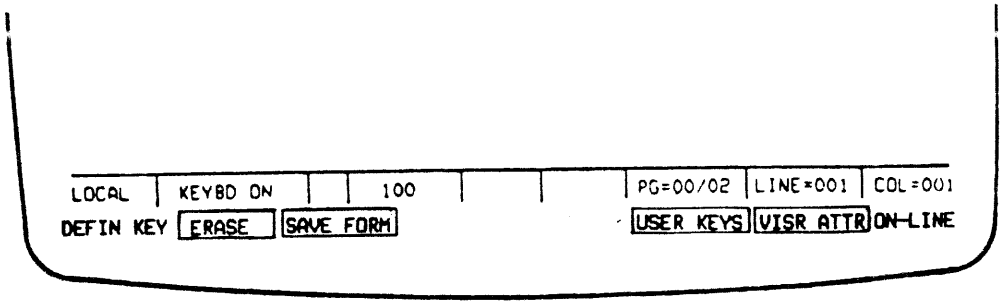


Figure 5-6 Extended Functions Softkeys

DEFIN_KEY -- This softkey provides the user with the capability to build user defined function keys. When the function key is pressed, the user function keys are displayed. See the section User_Defined_Softkeys for details.

ERASE -- Displays the ERASE softkey labels. Erase operations include: Clear Entry, Erase Unprotected, Erase from Beginning of Line or Page, Erase to End of Line or Page, and Erase All.

SAVE_FORM -- This softkey stores the form that is currently being displayed as a user defined function key and provides access to the current set of user keys.

USER_KEYS -- Displays a set of user defined function key labels that may be used to store user defined data or commands. The user may use these defined keys to transmit a message to the host or to display a form or collection of characters on the screen.

VISR_ATTR -- This softkey provides access to the softkeys that control visual attributes. Visual Attributes include: Normal, Highlight, Security, Underline, Reverse Video, Blinking, Double Width, Double Size.

ON-LINE/LOCAL -- This key toggles the terminal between on line and local operations. The current state is displayed in field A of the main status line.

> ON-LINE - When this function is enabled, the terminal is logically connected to the host, the message ON-LINE is displayed in the status line, and the softkey label changes to LOCAL.
> LOCAL - When this function is enabled, the terminal is logically disconnected from the host, the message LOCAL is displayed in the status line, and the softkey label changes to ON-LINE. Keyboard functions are executed locally and are not sent to the host. The terminal monitors the main port for a host sequence to put the terminal on-line.

PRINTER_FUNCTIONS

The PRINTER softkey in the primary level results in a second set of softkeys that provide extended terminal functions that are used to control a remote printer. The softkey labels are shown in Figure 5-7.

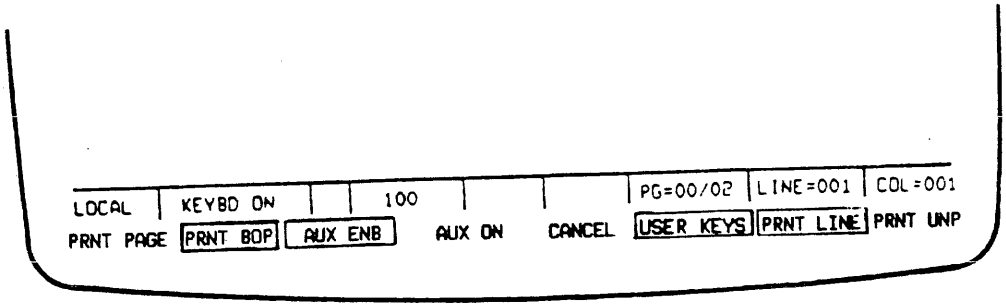


Figure 5-7 Printer Softkeys

PRNT_PAGE -- This softkey enables the transfer of all data from the home position to end of page to be transmitted to the remote printer. If there are markers (SOE) within the data, only the data from the most previous marker to the current cursor position will be transmitted.

PRNT_BOP -- This function enables the transfer of all data from the home position (beginning of page) to the cursor position to be transmitted to the remote printer. If there are markers (SOE) within the data, only the data from the previous marker to the current cursor position will be transmitted.

AUX_ENB -- This softkey enables the bidirectional communication between the terminal and the AUX port device.

AUX_ON -- This function enables the bidirectional communication between the AUX port device and the main port. Data are not processed by the terminal but are passed between ports.

CANCEL -- This softkey terminates the current print request. Before the next print request is started, a form feed may be issued to the printer, depending on TCM configuration.

USER_KEYS -- Displays a set of user defined function key labels that may be used to store user defined data. The user may use these defined keys to transmit a message to the host; or to display a form or collection of characters on the screen.

PRNT_LINE -- This function transmits the current line on the screen (marked by the cursor) to the remote printer.

PRNT_UNP -- This softkey key enables the user to have unprotected data printed on a pre-printed form. All protected data and attributes are transmitted to the printer as blanks for spacing, with only the unprotected and constant data actually being printed.

USER_KEYS_FUNCTIONS

The USER KEYS softkeys, illustrated in Figure 5-8, will be displayed when the user presses the USER KEYS softkey contained in most of the other sets of softkeys. The Define Key softkey contained in the Program Functions softkeys also leads to the USER KEYS.

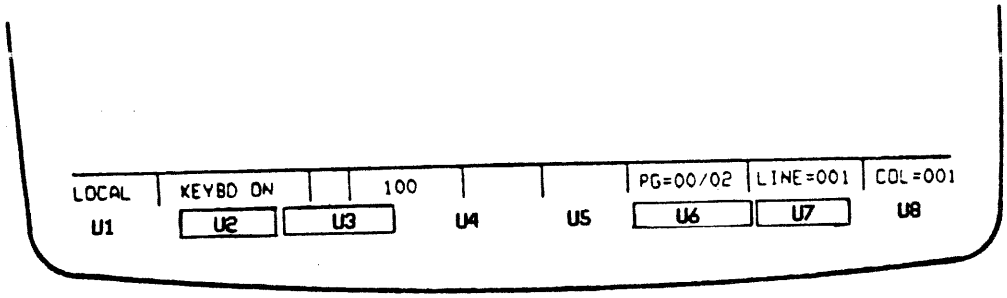


Figure 5-8 User Keys Softkeys

PROGRAMMER FUNCTIONS

The PROGRAMMER softkey in the primary level results in a second set of softkeys that provide extended terminal functions that are of special use to programmers. The softkey labels are shown in Figure 5-9.

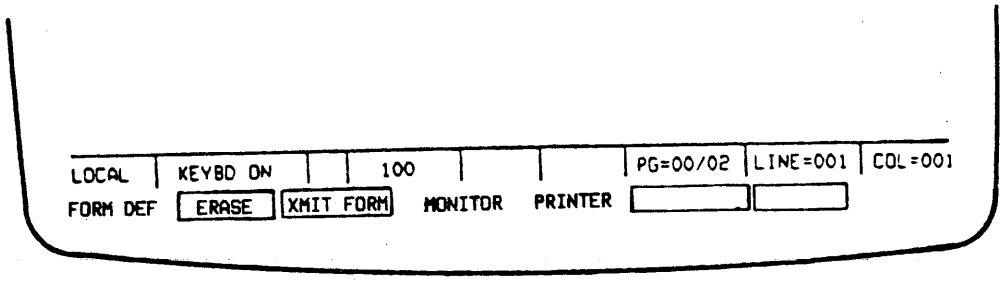


Figure 5-9 Programmer Softkeys

FORM DEF -- This key allows the user to define a form off-line for later transmission to the host or to be stored as a user key.

This softkey provides access to a set of softkeys used in defining forms. Forms definition operations include: editing operations, erase operations, graphic keys, user keys, and visual and logical attribute selection.

ERASE -- Displays the erase functions softkeys. Erase operations include: clear entry, erase unprotected, erase from beginning of line or page, erase to end of line or page, and erase all.

XMIT FORM -- This softkey transmits a form to the host with a special header signifying a form upload function.

MONITOR -- This softkey causes all entered and received data (including all control characters such as NULL and ESCape) to be displayed. All data are entered directly into display memory without execution. The label field blinks when this function is enabled. To terminate this function, press the softkey again. The label will return to normal intensity and control codes are no longer displayed on the screen.

PRINTER -- Displays a set of softkey labels to control a printer. Printer commands include: print line, page or unprotected, aux enable or on, printer cancel, and user defined keys.

FORMS DEFINE FUNCTIONS

The FORMS DEF softkey in the PROGRAM level results in an additional set of softkeys that provide forms definition functions. The softkey labels are shown in Figure 5-10.

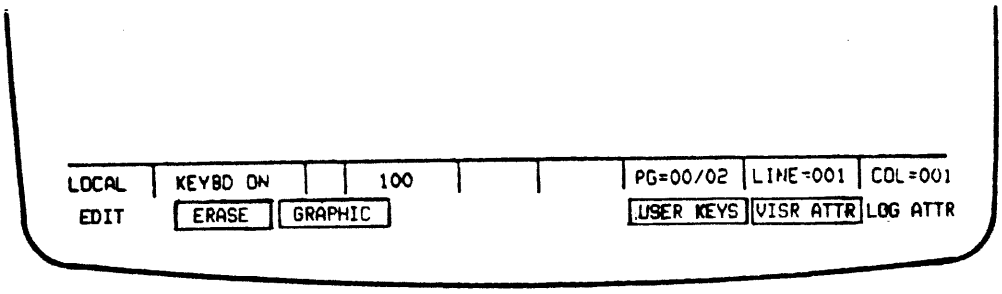


Figure 5-10 Forms Definition Softkeys

EDIT -- This softkey leads to the softkeys which perform editing functions on text in display memory. Editing operations include: modify/clear memory, erase, insert/delete character, activate printer, insert/delete line, and user keys functions.

ERASE -- This softkey leads to the softkeys which erase various positions of the current page of memory. Erase operations include: clear entry, erase unprotected, erase from beginning of line or page, erase to end of line or page, erase all.

GRAPHIC -- This softkey leads to the softkeys which produce the currently defined graphics characters. There are fourteen symbols available. This selection may be modified by Graphic Key Definition in Operator TCM. This graphic function also enables line-drawing capabilities using the Shift and Cursor Control keys.

USER KEYS -- This softkey displays the labels of the user-definable softkeys and permits their use. The user may define keys to transmit a message to the host or to display a form or collection of characters on the screen.

VISR ATTR -- This softkey provides access to the visual attributes. Visual attributes affect the display of the field.

LOG ATTR -- This softkey provides access to the logical attributes. Logical attributes affect the type of data that will be allowed in the field. They also affect the way in which data in the field is transmitted.

FORMS CREATION ON THE ATL-008

The FORM DEF softkey helps you define a form on the screen. Once the form has been created, it may be saved in a user key or transmitted to the host computer for future use. The process of defining a form has three major steps -- clearing all tabs, defining the visual appearance of the form, and defining the logical attributes of the various portions of the form.

Use the CTC (Cursor Tabulation Control) sequence to clear all horizontal tab stops. This enables automatic tabbing from one field to the next. Set TBM (Tab Stop Mode) so that tabs are independent on each line. This prevents the TAB SET logical attribute from setting a tab in every row every time it is used.

Define the visual portion of the form by entering all user prompts (NAME, ADDRESS, etc.) and their accompanying visual attributes. Be sure to leave enough space for operator input. The ERASE and EDIT functions may also be used.

Once the visual presentation is complete, select the LOG ATTR softkey to define the logical characteristics of the form. Typically, the entire screen, except the fields designed for operator input is PROTECTed or CONSTANT. PROTECTed means that these areas are not normally transmitted, nor can the operator alter them. CONSTANT means that these areas are transmitted, but may not be modified.

Each logical field is defined by moving the cursor to the first position of the field, selecting the type of field, and then moving the cursor to the position following the last position of the field and selecting one of the protected type of fields (PROTECT or CONSTANT). If automatic tabbing from one operator input field to the next is desired, the TAB SET attribute should be selected for the second field.

Identify all fields in this manner until the entire form is complete. The completed form may then be saved as a user key.

VISUAL ATTRIBUTES FUNCTIONS

The visual attribute softkeys allow definition of various visual attributes. There are two types of visual attributes: character oriented and row oriented. Character oriented attributes affect all subsequent characters. Character attributes are: NORMAL, HIGHLIGHT, SECURITY, UNDERLINE, REVERSE, and BLINK. Row oriented attributes affect existing and subsequent characters on the current row. The ATL-008 supports two row oriented attributes - DOUBLE WIDE and DOUBLE SIZE.

Attributes are selected by pressing the desired softkeys. The softkeys enable and disable the attributes. As character attributes are enabled, the corresponding softkey label blinks. When a character attribute is enabled, subsequent characters use that attribute until the attribute is disabled; i.e., as long as the label blinks, all characters in the field will use the attribute. Multiple selections are allowed. Combinations are stored as one attribute. Figure 5-11 illustrates the Visual Attributes Softkeys.

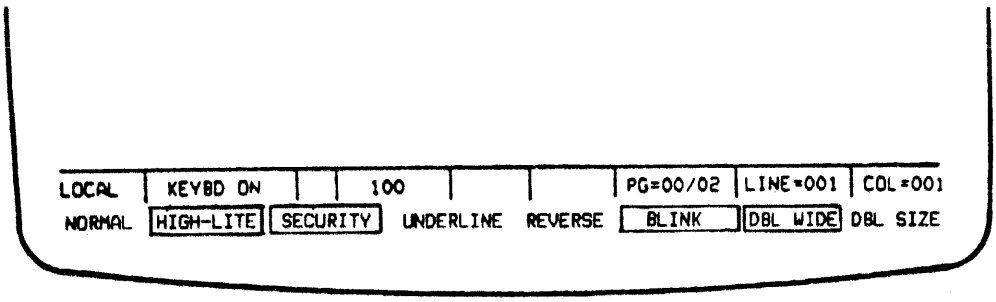


Figure 5-11 Visual Attributes Softkeys

NORMAL -- This softkey enables the normal video attribute and resets other character video attributes. Subsequent characters will be in normal video.

HIGHLIGHT -- This softkey enables the highlight video attribute and selects the alternate intensity as specified in Video Parameters in Operator TCM. Subsequent characters will be in highlighted.

SECURITY -- This softkey enables the security or no video attribute. Subsequent characters will not be displayed on the screen, but will be entered into display memory.

UNDERLINE -- This softkey enables the underline attribute. Subsequent characters will be underlined.

REVERSE -- This softkey enables the reverse video attribute. Subsequent characters will be in reverse video.

BLINK -- This softkey enables the blink attribute. New characters blink.

DBL_WIDE -- This softkey enables the the double wide/single high attribute. This attribute affects the entire current row. If this attribute is added to an existing row, the characters are all changed to be double wide. Any characters past the center of the screen are pushed off the right side of the screen. They are not lost and may be re-displayed by removing the attribute. The cursor is displayed as a double wide cursor and moves in increments of two columns. The softkey toggles the attribute on and off. The softkey label does not blink; the width of the cursor indicates that the attribute is active.

DBL_SIZE -- This softkey enables the double high/double wide attribute. This attribute affects the entire current row and the next row. Characters in the current line become the top half of a double high/double wide line. Characters in the next line become the bottom half of a double high/double wide line. This means that two identical lines must be entered for top and bottom unless you desire to type in lower level Klingonese. If this attribute is added to an existing pair of rows, existing and subsequent characters are

all changed to be double wide. Any characters past the center of the screen are pushed off the right side of the screen. They are not lost and may be re-displayed by removing the attribute. The softkey toggles the attribute on and off. The cursor is displayed as a double wide cursor and moves in increments of two columns. The softkey label does not blink; the width of the cursor indicates that the attribute is active.

LOGICAL ATTRIBUTES FUNCTIONS

The Logical Attribute Functions allow the user to specify a logical attribute for a particular field when defining a form. The Logical Attributes are shown in Figure 5-12. The first two softkeys allow selection of several different attributes. Pressing the softkey cycles the various attributes associated with the softkey.

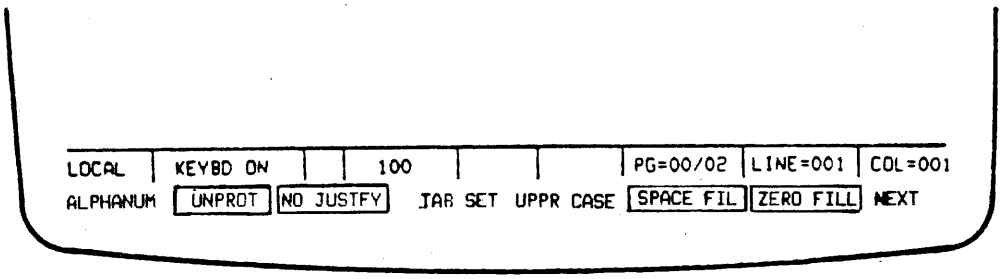


Figure 5-12 Logical Attributes Softkeys

ALPHANUM/GRAPHICS/ALPHA/NUMERIC/NUM ONLY -- This softkey allows specification of any one of five logical attributes concerning the type of data that will be allowed in a field. Press the softkey until the desired attribute is displayed and blinking.

ALPHANUM - This selection enables the alphanumeric attribute. Any character, including non-printable characters, may be entered in the field.
GRAPHICS - This selection enables the graphics attribute. Any printable character (greater than or equal to 20H) may be entered in the field.
ALPHA - This selection enables the alpha only attribute. This attribute allows only alphabetic characters to be entered into the field.
NUMERIC - This selection enables the numeric attribute. This attribute allows only numeric data and special characters related to numeric data (period, comma, minus, and plus) to be entered in the field.
NUM ONLY - This selection enables the numeric only attribute. This attribute allows only numeric data (0-9) to be entered into the field.

CONSTANT/UNPROTECT/PROTECT -- This softkey allows specification of three different field types. Press the softkey until the desired attribute is displayed and blinking.

UNPROT - This selection enables the unprotected attribute. This attribute allows keyboard entry into this field.
PROTECT - This selection enables the protect attribute. This attribute will not allow any keyboard entry into this field. This attribute automatically specifies an ALPHANUM selection.
CONSTANT - This selection enables the constant attribute. This attribute will protect this information field and will cause this field to be sent to the host when a transmit is initiated. This attribute automatically specifies an ALPHANUM selection.

NO_JUSTEY/R_JUSTIFY/L_JUSTIFY -- This softkey specifies justification.

NO_JUSTEY -- This selection specifies no justification.
R_JUSTIFY -- This selection specifies right justification. Any data entered into the field will be placed at the right edge of the field.
L_JUSTIFY -- This selection specifies left justification. Any data entered into the field will be placed at the left edge of the field.

TAB_SET -- This softkey enables the tab set attribute. This attribute sets a horizontal tab stop at the start of the field.

UPPR_CASE -- This softkey enables the upper case attribute. This attribute sets the field to act as though alpha lock is enabled.

SPACE_FIL -- This softkey enables the space fill attribute. All non-entered blanks become spaces.

ZERO_FILL -- This function enables the zero fill attribute. All non-entered blanks become zeros.

NEXT -- This softkey displays the next set of logical attribute softkeys. These additional functions are: must enter, total fill, transmit modified field, and transmit when exiting field.

NEXT LOGICAL ATTRIBUTE FUNCTIONS

The NEXT softkey in the logical attributes softkeys expands the set of logical attributes. These additional logical attribute softkeys are illustrated in Figure 5-13. To return to the first set of logical attribute softkeys, press the SELECT key. To exit back to the Define Forms menu, press SELECT twice.

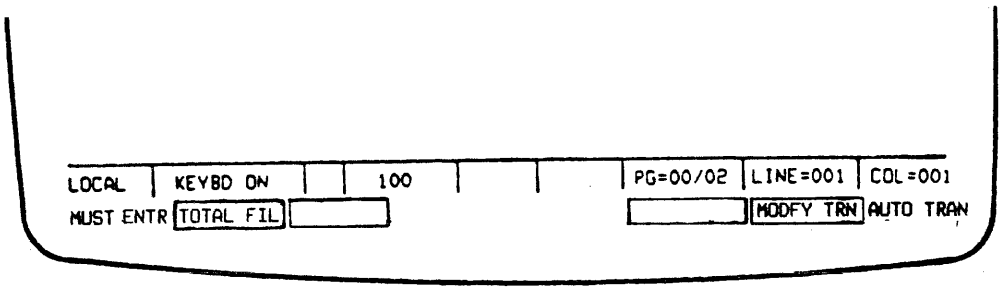


Figure 5-13 NEXT Set of Logical Attributes

MUST_ENTR -- This softkey enables the must enter attribute. This attribute forces entry of data into this field before leaving the field.

TOTAL_FIL -- This softkey enables the total fill attribute. This attribute forces the user to totally fill the field with data before leaving the field.

MODFY_TRN -- This softkey enables the modified data attribute. The field will be sent only if the field has been modified since the last transmission.

AUTO_TRAN -- This softkey enables the auto-transmit attribute. If the field has been modified, it is automatically transmitted when the field is exited.

5.2 USER DEFINED SOFTKEYS

5.2.1 Defining a User Key from the Keyboard

Before defining User Keys, be sure that enough special RAM usage space has been allocated in Buffer Allocations in Installation TCM.

To define a User Function key, follow these instructions:

1. Starting at the home position, type the desired contents of the user key. If control codes such as CRLF or ESCape sequences are needed, use the MONITOR facility provided by the PROGRAM softkey at the primary level. The MONITOR must be used prior to DEFINE KEY.

2. Press the EXTENDED softkey in the primary level set of softkeys.

3. Press the DEFINE KEY softkey to display the user keys labels.

4. Press the softkey that you wish to define.

5. A blinking underline cursor will appear in the corresponding softkey label. Type a nine-character label. The BACKSPACE key may be used to correct typing errors. Once all nine characters (blanks are accepted) have been typed, the cursor will return to the original position at the end of the text on the screen and the label will blink until the definition process is complete.

6. Press the user key again to store the softkey contents. Data between the home position and the cursor will be stored.

Example

The following example illustrates how to define and store a message in User Key U2. The first step is to enter the Buffer Allocations menu in Installation TCM and define some special RAM usage for user keys.

This example will illustrate how to define a user key so that it prints an underlined message.

Clear the screen (ERASE softkey, then ERASE ALL softkey). Press the HOME key. From the primary level, press the PROGRAM softkey, then the MONITOR softkey and press the following keys:

```
ESC[4mWhen programmed properly, U2 will display this message underlined.ESC[0m
```

Where ESC is the ESCape key. Because the MONITOR is active, do not use cursor controls or the BACKSPACE key -- the corresponding escape codes will be entered into the softkey string. If you make an error, temporarily disable the MONITOR, fix the mistake, enable the MONITOR and continue. The sequence ESC [4 m sets the underline attribute at the beginning of the text; ESC [0 m sets the normal video attribute at the end of the text. See the SGR (Select Graphic Rendition) sequence. While using the monitor, the sequence looks like Figure 5-14.

Leave the cursor at the end of the line, and press the MONITOR softkey to terminate the monitor. Press SHIFT SELECT to exit to the primary level. Press the EXTENDED softkey then the DEFIN KEY softkey.

Press the U2 softkey. An underline cursor will appear in the U2 field. Type Label_Two as the nine-character label. The screen will look like Figure 5-15.

Since the entire message and the the escape sequences to underline it are already on the screen, press the Label_Two softkey again to store the sequence.

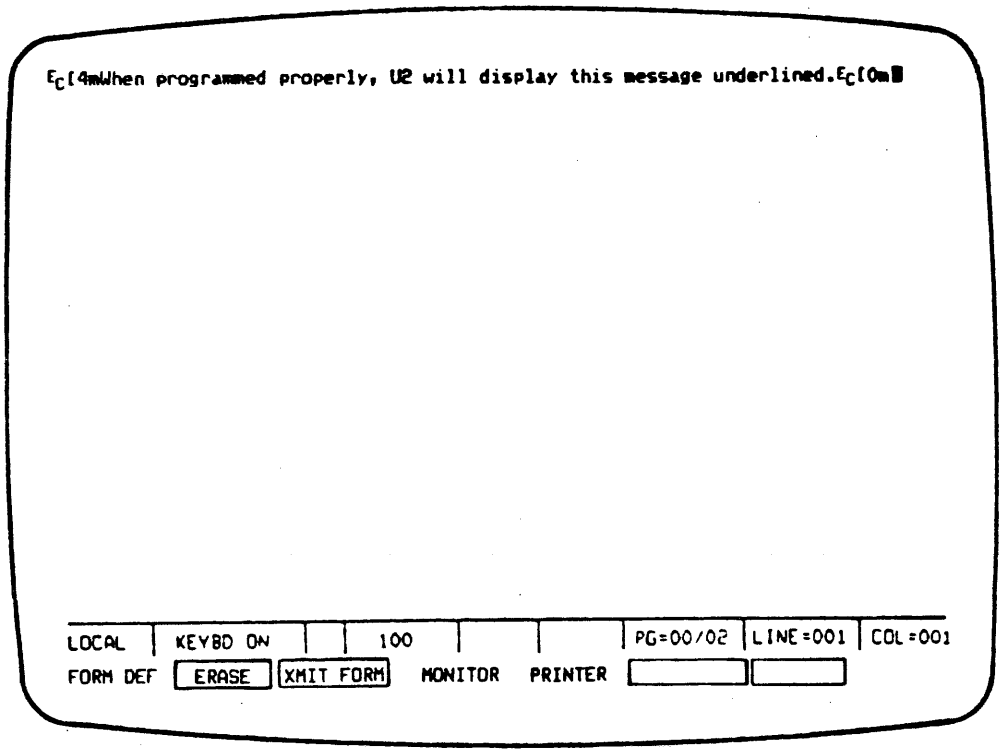


Figure 5-14 Using MONITOR to Define a User Key

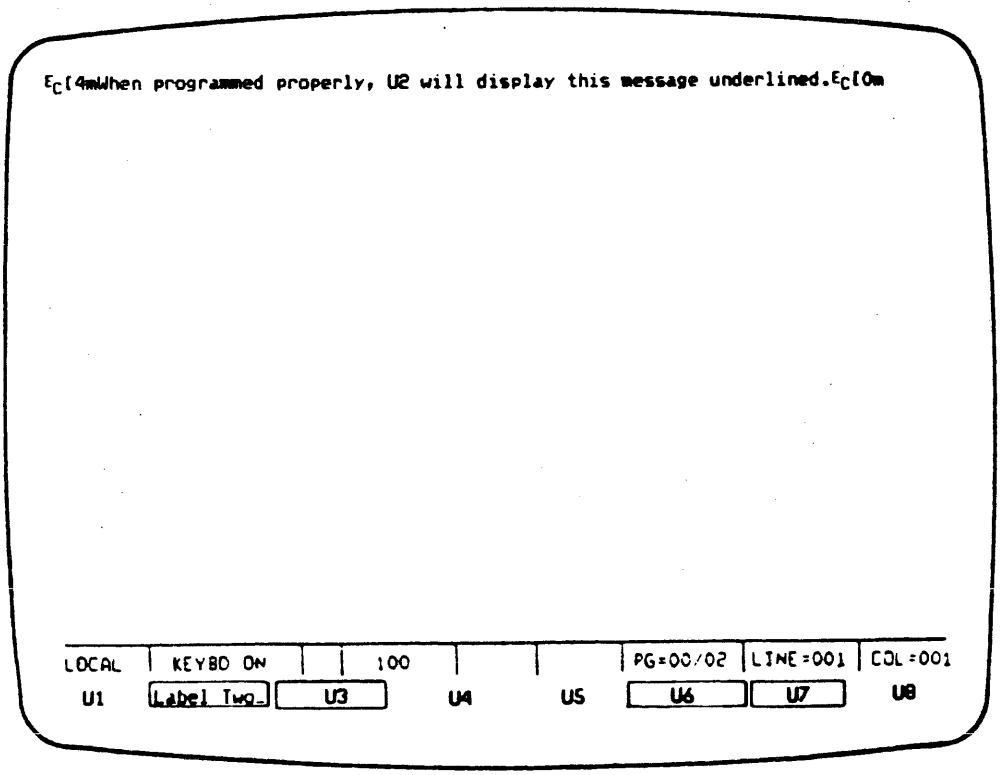


Figure 5-15 Typing the User Key Label

5.2.2 Using a User Key

The user-defined softkey labels are displayed when the USER KEY softkey is pressed. For convenience, the USER KEYS softkey is contained in most sets of softkeys. When the user keys labels are displayed, each label will either have the default label or a user-specified label. The user keys will not perform any functions unless a user label has been specified.

To use a User Key, follow the following instructions:

1. Find a set of softkeys that contains the softkey USER KEYS.
2. Press the USER KEYS softkey to display the User Keys Labels.
3. Press the desired user key and execution will be immediate.
4. Press the SELECT key to return to the calling softkey level.

Example

If you have done the example in the preceding section, press USER KEYS to display the user key labels, then press Label Two to display the underlined message on the screen.

5.2.3 Changing and Uploading Softkeys and Menus

The following functions are available:

Add a key:	ESC APC K n ; text ESC ST
Delete a key:	ESC APC K n ESC ST
Add a menu:	ESC APC R m;n;n;n;n;n;n;n ESC ST
Delete a menu:	ESC APC R m ESC ST
Execute key:	ESC APC E n ESC ST
Upload key:	ESC APC H n ESC ST
Upload reply:	ESC APC U n ; text ESC ST

Where:

n = key number in ASCII. n >= 0.
m = menu number in ASCII. m > 20.
text = 9-character label followed by contents of function key in transparency mode.

Text size is limited only by the amount of allocated user key RAM which is still free. (See Buffer Allocations in Installation TCM.) When a menu or key is deleted, the RAM space used by that entry is reclaimed into the remaining free RAM space.

Default_keys -- Key number 0 is a default with a blank label and null contents. When a key is deleted, all instances of that key in any menu are replaced with default key 0. If key 0 is deleted and a new key 0 is downloaded, the new key 0 will become the default key. If a new key 0 was not downloaded, key 0 reverts to a blank label and null contents.

Key numbers 1-8 are default keys for the keyboard user key menu. They are initialized to show a label 'Un' where n is the key number and have null contents. These keys can be loaded from the keyboard or downloaded from the host with the appropriate escape sequence. The labels are displayed and the keys are available when USER KEYS is selected. If one of these keys is deleted, it is replaced with the default user key.

Adding_a_key_or_menu -- New keys are added at the end of the key text area. If the key number being added was previously defined, it is deleted before the new key is downloaded. If the key overflows the available user key RAM, the key is truncated. If a menu overflows the user key RAM, it is not created. If a key or menu is truncated, the undefined command byte is incremented and the warble is sounded.

When a key or menu is deleted, the RAM is returned to the user key RAM pool. The labels for the current menu are redisplayed, in case the deleted key was part of the current menu. The deleted key is sought in all defined menus; when it is found it is replaced with the default key 0. If the menu being deleted is the current menu, the primary soft key menu is displayed (menu 0).

Execution_of_a_function_key_from_the_keyboard -- The function key text is sent to the command interpreter one byte at a time with the source set to the keyboard. The text is processed as though the user had typed it at the keyboard.

Execution_of_a_function_key_from_the_main_port - The function key text is sent to the command interpreter one byte at a time with the source set to the main port. The text is processed exactly as though the host had sent the text.

Upload_of_a_function_key -- The uploaded key is sent to the main port. If the key is not defined, the sequence ESC APC U ESC ST is sent. Transparency mode is used.

Selecting_a_soft_key_menu -- The Beehive sequence ESC [<Pn> p is used to select a new soft key menu. If the new menu number is greater than 20, the new menu is a user key menu. If the menu is not defined, the menu is not changed, the undefined command byte is incremented, and the warble is sounded.

See Table 5-1 for parameters. See the BEESKM command as well.

Table 5-1
Softkey Menu Access

ESC [0 p	primary functions
ESC [1 p	primary functions
ESC [2 p	edit functions
ESC [3 p	graphic functions (keys 1-7)
ESC [4 p	printer control functions
ESC [5 p	extended functions
ESC [6 p	programmer functions
ESC [7 p	reserved
ESC [8 p	user function keys
ESC [9 p	erase functions
ESC [10 p	forms define functions
ESC [11 p	graphics keys (keys 8-14)
ESC [12 p	reserved
ESC [13 p	reserved
ESC [14 p	visual attributes
ESC [15 p	logical attributes
ESC [16 p	next logical attributes

5.2.4 Saving and Deleting Softkeys and Menus in EEPROM

Saving a user key in EEPROM -- The following sequence saves the RAM definition of a softkey in EEPROM:

```
ESC APC L n [;n] ESC ST
```

This sequence is used for all keys (no distinction between keys 1-8 and the rest). Upon terminal power-up, all f key definitions saved in EEPROM would be loaded. The APC L sequence allows multiple parameters so that several keys may be saved with a single command. The [;n] in the above sequence indicates that additional key numbers may be used.

Saving a menu in EEPROM -- The following sequence stores the definition of a user-defined menu (consisting of the menu number and the key numbers for each of the eight key fields) in EEPROM:

```
ESC APC M n [;n] ESC ST
```

On terminal power-up, all menu structures are loaded from EEPROM into RAM. Also the APC M sequence allows multiple parameters so that several menus may be saved with a single command. The [;n] in the above sequence indicates that additional key numbers may be used.

Delete a key or menu from EEPROM -- The following sequence deletes a key from EEPROM:

```
ESC APC N 1 ; n [;n] ESC ST
```

This sequence frees allocated EEPROM space. The parameter string will consist of a 1 or a 2 followed by a variable length string of numbers separated by semicolons. The 1 indicates that the following numbers are softkey numbers to be removed from EEPROM. The 2 indicates that the following numbers are menu numbers to be removed from EEPROM. If no softkey or menu numbers are present in the parameter string, then all softkey or menu definitions (depending on the 1 or 2) will be removed from EEPROM.

Save a new power-up menu number -- If you wish to change the power-up default menu, use the following sequence:

ESC APC O n ESC ST

This sequence allows a menu number (either one of the standard menus or a user-defined menu) to be activated when the terminal powers up. The power-up menu is used as the primary menu when a SHIFT SELECT sequence is done from the keyboard. Transparency is used with this sequence.

Transparency -- Transparency allows user-defined function keys to include escape sequences as well as ANSI control characters while not violating the ANSI X3.64 specification. This method is used for the following sequences:

ESC APC K n text ESC ST
ESC APC U n text ESC ST

Bootload APC

During communications with a host computer, the following conventions must be observed:

Transmitter -- If a code less than 20H (SPACE) is to be transmitted, send a 26H (ampersand - &), followed by the original character augmented by 40H. If an ampersand is to be transmitted, send two ampersands. Other codes are transmitted normally.

Receiver -- If a 26H (ampersand - &) is received, examine the subsequent character. If it is also an ampersand, discard one of the ampersands and pass the other one through. If it is in the range of 40H - 5FH, discard the ampersand, subtract 40H from the second character and pass the result through. If the subsequent character is not an ampersand or in the range of 40H - 5FH, then the received APC sequence is in error and should be aborted.

5.2.5 PROGRAMMED FUNCTION KEYS

The program function keys are enabled by depressing the CONTROL or SHIFT along with one of the softkeys. Depending on TCM options (PF KEY DEFINITION in Installation TCM), the PF keys will cause one of the following sequences to be sent to the host. Each of these sequences is given here with its ANSI mnemonic, I/O, and keyboard equivalents.

APC option (ICM selection 1):

Mnemonic:	APC	P	<Ps>	ST	
From I/O:	1B 5F	50	<Ps>	1B 5C	(7-bit)
	9F	50	<Ps>	9C	(8-bit)
From keyboard:	ESC _	P	<Ps>	ESC	

Transparency is used with this sequence. See Table 5-2 for the <Ps> parameters.

OSC option (ICM selection 2):

Mnemonic:	OSC	P	<Ps>	ST	
From I/O	1B 5D	50	<Ps>	1B 5C	(7-bit)
	9D	50	<Ps>	9C	(8-bit)
From Keyboard:	ESC]	P	<Ps>	ESC	

See Table 5-2 for the <Ps> parameters.

PM option (ICM selection 3):

Mnemonic:	PM	P	<Ps>	ST	
From I/O	1B 5E	50	<Ps>	1B 5C	(7-bit)
	9E	50	<Ps>	9C	(8-bit)
From keyboard:	ESC ^	P	<Ps>	ESC	

See Table 5-2 for the <Ps> parameters.

DCS option (ICM selection 4):

Mnemonic:	DCS	P	<Ps>	ST	
From I/O	1B 50	50	<Ps>	1B 5C	(7-bit)
	90	50	<Ps>	9C	(8-bit)
From Keyboard:	ESC P	P	<Ps>	ESC	

See Table 5-2 for the <Ps> parameters.

SS3 option (ICM selection 5):

Mnemonic: SS3 P <Ps> ST

From I/O:

1B 4F	50	<Ps>	1B 5C	(7-bit)
9F	50	<Ps>	9C	(8-bit)

From Keyboard: ESC O <Ps>

See Table 5-2 for the <Ps> parameters.

Table 5-2
<Ps> Parameters for Individual PF Keys

PF_KEY	<Ps> char	hex	ANSI
1	P	50	5/0
2	Q	51	5/1
3	R	52	5/2
4	S	53	5/3
5	T	54	5/4
6	U	55	5/5
7	V	56	5/6
8	W	57	5/7
9	X	58	5/8
10	Y	59	5/9
11	Z	5A	5/10
12	[	5B	5/11
13	\	5C	5/12
14	]	5D	5/13
15	^	5E	5/14
16	_	5F	5/15

SECTION 6. ANSI MODE CONTROL FUNCTIONS

6.1. GENERAL

The ATL-008 implements all of the appropriate control sequences described in ANSI X3.64-1979. This section describes the types of sequences, their structure, their functions, and their effect on the terminal.

This section is organized as a reference to the portions of ANSI X3.64 that apply to the ATL-008 terminal. It is not an ANSI tutorial. It is, however, organized in such a way that an untutored user may gain a lot from it.

ANSI sequences have a variety of syntaxes or structures. The representation of these structures requires a few naming conventions that provide a basis for discussing the sequences. These conventions are discussed first.

The standard provides for strings to control software and to control hardware devices. The function of these strings is dependent on the particular installation and application. The structure of these strings is only dependent on the naming conventions. Because these strings use the various conventions, they are covered next.

The ANSI standard also defines a variety of modes that affect the way subsequent commands are handled. These modes are changed by ANSI sequences. Modes (and the sequences that change them) are discussed after control strings.

ANSI command sequences are discussed after modes that affect the commands. Beehive additions to the standard are discussed after the standard sequences.

Non-implemented standard sequences are listed at the end of this section.

The section is organized in this order:

- Notations and Conventions
- Control Strings and Definitions
- Delimiter Strings for Software Control and for Device Control
- Introducers
- Modes
    - Device Sending Modes
    - Device Local and Receiving Modes
    - Device Local Modes
    - Sending, Receiving, and Local Modes
- Format Effectors
- Controls to Move Cursor
- Editing to Alter Visual Display
- Form Filling
- Miscellaneous Controls
- Beehive Private Extensions to ANSI 3.64
- Non-implemented Sequences

6.2. NOTATIONS AND CONVENTIONS

This section discusses definitions and conventions for describing ANSI sequences.

6.2.1. ANSI Control Sequences

There are three types of ANSI control sequences. They are listed below with their corresponding structures:

Control Sequence	Structure	
Independent Control Functions	ESC Fe	(7-bit)
	Ce	(8-bit)
Control Sequences Without Intermediates	CSI P...P F	
Control Sequences With Intermediates	CSI P...P I F	

ESC is the ESCape character, 1BH. Fe is a Final character in an escape sequence. Ce is a Control character. CSI is a Control Sequence Introducer. P...P is a list of Parameters. F is a Final character, and I is an Intermediate character. These characters are discussed in this section.

Independent Control Functions -- An Independent Control Function has no parameters. In a 7-bit environment, an Independent Control Function is composed of an ESCape character followed by a single Final character abbreviated Fe. In an 8-bit environment, an Independent Control Function consists of a single Control character abbreviated Ce. To determine what the Ce character should be for a particular function in an 8-bit environment, add 40H to the Fe given for that function in a 7-bit environment.

Example: HTS Horizontal Tab Set

In a 7-bit environment, this control function consists of:

ESC	H	or	1BH	48H
Escape character	Final character			

In an 8-bit environment, this control function is 88H because 48H + 40H = 88H.

Control Sequences Without Intermediates -- This type of sequence consists of:

CSI	a Control Sequence Introducer
P...P	Optional parameters
F	a Final character

Example: DAQ Define Area Qualification

ESC [	<Ps>	o
CSI	Parameter	Final character

Definitions of all the various parts of this type of sequence follow.

Control Sequences With Intermediates -- This type of sequence consists of:

CSI	a Control Sequence Introducer
P...P	Optional parameters
I	an Intermediate character
F	a Final character

Example: SL Scroll Left

ESC [	<Pn>	SPACE	@
CSI	Parameter	Intermediate	Final character

Definitions of all the various parts of this type of sequence follow.

Definitions:

CSI (Control Sequence Introducer) -- A Control Sequence Introducer indicates that a Control Sequence follows (i.e. this is different from normal text). The ANSI definition of CSI in an 8-bit environment is 9BH. In 7-bit environments it is 1BH 5BH (ESC [ ). In an 8-bit environment either 9BH or 1BH 5BH is interpreted correctly; however, only 1BH 5BH is transmitted. The environment is determined by the setting of the Main Port Data Length. See the paragraph called Main Port Hardware Interface in Section 3.

P (Parameters) -- An ANSI control sequence may have two different types of parameters: numeric parameters and selective parameters.

- Numeric Parameter <Pn> - As the name suggests, this is a string of decimal digits (0 (30H) through 9 (39H)). It specifies a number. This number can indicate the number of times to perform a function. For example, a parameter of 5 to the IL (insert line) control sequence means to insert 5 lines. A numeric parameter could also indicate a row or column number, as with the CUP (cursor position) control sequence. A numeric parameter is represented by the symbol <Pn>.

- Selective Parameter <Ps> - A selective parameter can be made up entirely of decimal digits, or it may contain other characters. It specifies which one of several options to choose. For example, the EL (erase in line) control sequence uses a selective parameter to indicate whether a specified portion or the entire line should be erased. A selective parameter is represented by the symbol <Ps>.

- Parameter String - More than one parameter may be specified in some control sequences to indicate that more than one function is to be performed. For example, to specify multiple visual attributes, such as blinking underscored text, the selective parameters for blink and underscore would both have to be supplied to the SGR (select graphic rendition) control sequence. Two or more parameters separated by semicolons (;, 3BH) comprise a parameter string. In control sequence examples in this manual, a parameter string may appear as <Ps>;<Ps> (two parameters), or as <Ps>;<Ps>... or <Ps>...<Ps> (many parameters).

- Default Value - If no value is specified for a parameter, or if a value of zero (0, 30H) is specified, a default value is assumed. The default value depends on the particular control sequence. For example, supplying default values for both the row and column parameters to the CUP (cursor position) control sequence causes the cursor to be placed at the home position. Supplying the default value to the CUF (cursor forward) control sequence causes the cursor to move one position forward.

I (Intermediate Character) -- An Intermediate Character is used to extend the number of available control sequences. The ANSI standard allows any of the 16 characters between Space (20H) and / (2FH) to be used as an intermediate character. Thus, there are 16 times as many control sequences with one intermediate available as there are control sequences with no intermediates. The control sequences currently implemented in the ATL-008 which require intermediates all use the SPACE (20H) as the intermediate.

F (Final Character) -- The final character is a unique character for each particular control sequence. It serves to identify the particular control sequence as well as to terminate it.

NOTE: 1. There can be no imbedded blanks or spaces in an ANSI control sequence. Spaces are included in the control sequences shown in this manual for clarity only. If a Space character is part of a control sequence, the word SPACE will be used.

2. Unless otherwise noted, leading zeros in parameters are ignored. Thus, parameters written as 3, 03, and 00003 are all normally interpreted as having a value of 3. Some sequences (such as DCS) may require leading zeros.

6.2.2 Editing Functions and Format Effectors

There are two classes of control sequences that are easily confused: editing functions and format effectors. In many cases, a given editor function has a corresponding format effector. The primary purpose of an editor function is to edit or alter the visual arrangement of previously entered information. The primary purpose of format effectors is to be treated as data that happen to have a format representation rather than (or in addition to) a displayable representation. A particular application may accentuate, diminish, or even eliminate the difference between corresponding editor functions and format effectors.

6.2.3 Illegal Commands and Parameters

An illegal control sequence or parameter causes the following actions, whether the command came from the main or auxiliary port or from the keyboard:

the command is not executed
the Undefined Command bit in the status line is set
the Undefined Command field in the status byte is incremented
the warble sounds when in ANSI mode

6.2.4 User-defined Function Keys

User definable function keys may be programmed to contain any of the ANSI sequences described herein. A user may choose to do this for the following reasons:

- To provide the operator with a labeled single key which invokes an otherwise lengthy sequence.
- To send a special message (APC, OSC, PM, or DCS) to the host.

See Section 5 for information on how to define user-defined keys.

6.3 DELIMITER STRINGS FOR SOFTWARE CONTROL

There are three type of delimiter strings provided for software control. The function of each of these depends on the particular application and installation. The three delimiter strings for software control are the APC (Application Program Command), the OSC (Operating System Control), and the PM (Privacy Message). The structures are similar. They all end with a string terminator (ST) that consists of an ESC character and a backslash character (1BH and 5CH, respectively).

6.3.1 APC Application Program Command

STRUCTURE: ESC _ <data> ESC \

The APC sequence is used to load PF keys, to define softkeys and for bootloading. For information on its use with PF keys see the paragraph called PF Key Definition under Installation TCM in Section 3 of this manual. To see how it is used to define softkeys, see the paragraph called Changing and Uploading Softkeys and Menus in section 5. For information concerning bootloading, see the section on bootloading.

6.3.2 OSC - Operating System Control

STRUCTURE: `ESC ] <data> ESC \`

The OSC is used to pass a string of data whose interpretation is dependent upon the operating system.

NOTE: Via TCM, PF keys may be configured to send a OSC sequence to the host. (See the paragraph entitled PF Key Definition in Section 3.)

6.3.3 PM - Privacy Message

STRUCTURE: `ESC ^ <data> ESC \`

PM passes a string of data whose interpretation is dependent upon the individual privacy and security methods in effect.

NOTE: Via TCM, PF keys may be configured to send a PM sequence to the host. (See the paragraph entitled PF Key Definition in Section 3.)

ST - String Terminator -- All the software control sequences end with a string terminator, abbreviated ST. ST has the following structure:

`<cmd> <data> ESC \`

6.4 DELIMITER STRING FOR DEVICE CONTROL (DCS)

There is one delimiter string for device control. It is the DCS Device Control String. Leading zeros are required with this sequence for all numeric parameters (<Pn>. It has a variety of defined parameters and functions:

STRUCTURE: `ESC P <Ps> <data> ESC \`

The following parameters and functions have been defined for the ATL-008:

`ESC P P <Pn> <data> ESC`	The P parameter specifies a TCM download to RAM registers. TCM parameters will be not retained after turning the terminal off. See Appendix D.
`ESC P W <Pn> <data> ESC`	The W parameter specifies a TCM download to EEROM registers. TCM parameters will be retained after turning the terminal off. See Appendix D.
`ESC P N <Pn> ESC`	The N parameter specifies a request to send the TCM parameters From RAM to the host. See Appendix D.
`ESC P H <Pn> ESC`	The H parameter specifies a request to send the TCM parameters from EEROM to the host. See Appendix D.

ESC P I <data> ESC

The I parameter specifies a report of TCM parameters to the host. This is sent in response to a request for a TCM parameter report. See Appendix D.

ESC P S <data> ESC

The S specifies that a selftest is to be executed <data> has the following format:

S/C <Ps> ; <Ps> ; ...

The S or C indicates that the test is to be executed a single time or continuously. The following table lists the parameters that specify the actual test(s) to be run. Multiple parameters are separated with semicolons.

Selftest parameters

<Ps>		Function
ASCII	Hex	
0	30	ROM test
1	31	EEROM test
2	32	RAM test
3	33	Keyboard test
4	34	Timer ratio test
5	35	Main I/O test
6	36	Aux I/O test
7	37	Reserved
8	38	Reserved
9	39	Reserved

ESC P U <text> ESC

The U parameter specifies a user message from the host. The text is displayed in the position of the status line. <Text> is the text to be displayed. If more than 80 characters are sent they will be truncated.

ESC P O ss <Text> ESC

The O parameter specifies that a user-defined optional status line is being downloaded. The user-defined status line that take the place of the standard ATL-008 status line. ss specifies the offset from the beginning of the status line to where the text will be placed (expressed as two ASCII digits). <Text> is the text to be stored in the status line. Text that extends beyond the end of the 80 character status line will be truncated.

ESC P T ESC

The T parameter specifies that the terminal status is requested. See Section 4.

ESC P R <data> ESC	The R parameter identifies the <data> as a status report in response to a status request. The status data is described in Section 4.
ESC P C ESC	The C parameter requests a cursor position report. The report is made with the CPR sequence.

In addition to these functions, PF keys may be configured though TCM to send a DCS sequence to the host. (See the paragraph entitled PF Key Definition in Section 3.)

6.5 INTRODUCERS

There are three different sequences used to introduce a change in the type of data being sent. The Control Sequence Introducer indicates that subsequent characters are part of a control sequence as previously described. The CSI is repeated here for your convenience. The Single Shift 2 and Single Shift 3 introducers affect only the subsequent character (not characters) and are used to display additional characters in a manner analogous to a shift key on a typewriter.

6.5.1 CSI Control Sequence Introducer

STRUCTURE: ESC [

The CSI command is used to introduce a command string. These various command strings are defined in the following pages.

6.5.2 SS2 Single Shift 2

STRUCTURE: ESC N (character code (! - ~))

The SS2 specifies that a character from Optional Character Set #1 is to be displayed. The effect of SS2 applies to only one character, the one that immediately follows the SS2. Using SS2, the characters from the character generator (see Appendix C) occupying positions 80 Hex to DD Hex may be displayed.

6.5.3 SS3 Single Shift 3

STRUCTURE: ESC O (character code (! - ~))

The SS3 specifies that a character from Optional Character Set #2 is to be displayed. The effect of SS3 applies to only one character, the one that immediately follows the SS3. Using SS3, the characters from the character generator (see Appendix C) occupying positions DE Hex to FF Hex may be displayed.

6.6 CONTROL OF ANSI MODES

There are twenty ANSI control modes. With one exception they have two states. The two states are called set and reset. Changes between these two states are made by means of two control sequences. The sequences are called Set Mode and Reset Mode, abbreviated SM and RM respectively. Set Mode and Reset Mode are not modes -- they are commands to set or reset a mode or modes. These two commands are discussed here because of their relationship to the modes that follow. The differences between the set and reset states of the various modes are detailed in the sections pertaining to each individual mode.

6.6.1 RM - Reset Mode

STRUCTURE: ESC [<Ps> l (Lowercase L)

RM resets one or more modes as specified by the parameters in the parameter string. Each mode to be reset is specified by a separate parameter. The modes defined are specified separately. This sequence turns off any mode indication in the status line and resets the corresponding indicators in the status byte. The parameters are as follows:

<Ps>		Mode	* indicates a mode listed in the ANSI status line.
ASCII	Hex		
1	31	GATM	Guarded Area Transfer Mode *
2	32	KAM	Keyboard Action Mode
3	33	CRM	Control Representation Mode
4	34	IRM	Insertion-Replacement Mode
5	35	SRTM	Status Reporting Transfer Mode *
6	36	ERM	Erasure Mode *
7	37	VEM	Vertical Editing Mode *
10	31 30	HEM	Horizontal Editing Mode *
11	31 31	PUM	Positioning Unit Mode
12	31 32	SRM	Send-Receive Mode
13	31 33	FEAM	Format Effector Action Mode *
14	31 34	FETM	Format Effector Transfer Mode *
15	31 35	MATM	Multiple Area Transfer Mode *
16	31 36	TTM	Transfer Termination Mode *
17	31 37	SATM	Selected Area Transfer Mode *
18	31 38	TSM	Tabulation Stop Mode
19	31 39	EBM	Editing Boundary Mode
20	32 30	LNM	Line Feed New Line Mode
;	3B		Standard Separator for Parameters

Example

To reset the Tabulation Stop Mode (TSM) and Editing Boundary Mode (EBM), use the RM sequence with parameters of 18 and 19. The sequence would be

ESC [18; 19 l

6.6.2 SM - Set Mode

STRUCTURE: ESC [<Ps> h

SM sets one or more modes of the receiving device as specified by each parameter in the parameter string. Each mode to be set is specified by a separate parameter. A mode is set until reset by a Reset Mode (RM) control sequence. The defined modes are separately specified. The parameter definitions are listed under the Reset Mode (RM) command.

This sequence turns on mode indications in the status line and sets the corresponding indicators in the status byte.

6.7 DEVICE SENDING MODES

These modes affect the manner in which data is transferred between the terminal and the host.

6.7.1 FETM - Format Effector Transfer Mode

PARAMETER FOR RM AND SM: 14

FETM determines how format effectors are sent by the terminal. This mode is set or reset by means of the SM or RM commands, respectively. See the discussion of these commands for the sequence to use this mode.

- If the mode is set by SM, only the control codes that are stored on the screen will be transmitted along with the data when transmitting data from the screen to the host or to an auxiliary device. No control codes will be inserted by the terminal.

- If the mode is reset by RM, the terminal will insert control codes into the data stream (in addition to those stored on the screen) when transmitting data from the screen to the host or to an auxiliary device. CR (or CR, LF if selected in TCM) will be inserted at the end of each line.

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.7.2 GATM - Guarded Area Transfer Mode

PARAMETER FOR RM AND SM: 1

GATM determines whether or not guarded areas will be transmitted. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If the GATM is set by SM, both guarded and unguarded data will be sent when transmitting data from the screen to the host or to an auxiliary device.

- If the mode is reset by RM, only unguarded data will be sent when transmitting data from the screen to the host or to an auxiliary device.

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.7.3 MATM Multiple Area Transfer Mode

PARAMETER FOR RM AND SM: 15

MATM determines whether or not multiple areas (in addition to the area containing the cursor) are transmitted. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

NOTE: For this mode setting to take effect, the Selected Area Transfer Mode (SATM) must be in the reset state.

- If MATM is set by SM, all selected fields will be sent when transmitting data from the screen to the host or to an auxiliary device.

- If MATM is reset by RM, only the field that contains the cursor will be sent when transmitting data from the screen to the host or to an auxiliary device.

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.7.4 SATM Selected Area Transfer Mode

PARAMETER FOR RM AND SM: 17

SATM selects the entire page or portions of the current page (selected areas) for transmission. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If SATM is set by SM, the full contents of the page are sent when transmitting data from the screen to the host or to an auxiliary device.

- If SRTM is reset by RM, only the area(s) that are selected by SSA (Start of Selected Area), ESA (End of Selected Area) and DAQ (Define Area Qualification) will be sent when transmitting data from the screen to the host or to an auxiliary device.

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.7.5 SRTM - Status Reporting Transfer Mode

PARAMETER FOR RM AND SM: 5

SRTM determines the function of status reporting through use of the DCS sequence. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If SRTM is set by SM, every transmission to the host will include a Device Control String (DCS) Status Report at the front of the data.
- If SRTM is reset by RM, the Device Control String (DCS) will only be sent to the host upon request.

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.7.6 TTM - Transfer Termination Mode

PARAMETER FOR RM AND SM: 16

TTM determines whether or not the cursor is used to indicate the end of the data to be sent. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If TTM is set by SM, the position of the cursor has no effect on specifying the end of the data to be sent when transmitting data from the screen to the host or to an auxiliary device. See descriptions of SSA, ESA, and DAQ.
- If TTM is reset by RM, the position of the cursor is used to determine the end of the data to be sent when transmitting data from the screen to the host or to an auxiliary device. See descriptions SSA, ESA, and DAQ.

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.8 DEVICE LOCAL AND RECEIVING MODES

These modes determine how data is received at the terminal. They also control how it is displayed.

6.8.1 EBM - Editing Boundary Mode

PARAMETER FOR RM AND SM: 19

This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If EBM is set by SM, the effects of certain editor functions (see below) are not limited to the currently displayed window; they may affect the entire page.

- If EBM is reset by RM, the effects of certain editor functions (see below) are limited to the currently displayed window.

A variety of functions depend on this mode. It is called Editing Boundary mode because it specifies the bounds of the data to be edited. For example, the ICH function (insert character) depends on this mode. If EBM is set, then characters are inserted on a page basis; if EBM is reset, characters are inserted only in the current window.

EBM affects the following editing functions:

DL	Delete Line
EA	Erase In Area
ECH	Erase Character
ED	Erase In Display
EF	Erase In Field
EL	Erase In Line
ICH	Insert Character
IL	Insert Line

EBM affects the following editing positioning functions:

CBT	Cursor Back Tab
CHT	Cursor Horizontal Tab
CNL	Cursor Next Line
CPL	Cursor Previous Line
CUB	Cursor Backward (Left)
CUD	Cursor Down
CUF	Cursor Forward (Right)
CUU	Cursor Up
CVT	Cursor Vertical Tab

EBM does not affect:

NP	Next Page
PP	Previous Page
SD	Scroll Down
SL	Scroll Left
SR	Scroll Right
SU	Scroll Up
CHA	Cursor Horizontal Absolute
CUP	Cursor Position

6.8.2 ERM - Erasure Mode

PARAMETER FOR RM AND SM: 6

This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If ERM is set by SM, erase functions obliterate characters regardless of their protected state.
- If ERM is reset by RM, erase functions erase only unprotected characters.

Refer to SPA (Start of Protected Area), EPA (End of Protected Area), and DAQ (Define Area Qualification).

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.8.3 FEAM Format Effector Action Mode

PARAMETER FOR RM AND SM: 13

This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If FEAM is set by SM, format effectors are not performed when received in a data stream. They are stored.
- If FEAM is reset by RM, format effectors take action immediately when received in a data stream.

If CRM is set, format effectors will also have a displayable representation regardless of the state of FEAM.

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.8.4 HEM - Horizontal Editing Mode

PARAMETER FOR RM AND SM: 10

HEM determines how characters are shifted in editing operations. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If HEM is set by SM, a character insert causes a string of data at and preceding the cursor position to be shifted backward (left), and a character deletion causes a string of data preceding the cursor to be shifted forward (right).

- If HEM is reset by RM, a character insert causes a string of data at and following the cursor position to be shifted forward (right), and a character deletion causes a string of data following the cursor position to be shifted backward (left).

- HEM affects the action of:

  DCH DELETE CHARACTER
  ICH INSERT CHARACTER

- Entered or received data is affected by IRM (Insertion Replacement Mode).

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.8.5 IRM Insection - Replacement Mode

PARAMETER FOR RM AND SM: 4

IRM determines whether new characters are inserted or whether they replace existing characters. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If IRM is set by SM, the character under the cursor and all following (or preceding, if HEM is set) are shifted right (or left if HEM is set). Then the received character is written under the cursor. Finally, the cursor is advanced one position to the right (if HEM is set, then the cursor is not advanced).

- If IRM is reset by RM, the received character is stored at the current cursor position, replacing whatever was there before.

The operation of editing functions used by this mode is also affected by HEM.

6.8.6 SEM Select Editing Extent Mode

STRUCTURE: ESC [<Ps> Q

SEM is different from other ANSI modes because it has four states instead of the two states provided by the SM and RM sequences. Because of this, it is not used with SM and RM; it has its own sequence. SEM defines the extent of the display to be affected by ICH (insert character) and DCH (delete character). SEM also defines the extent of inserted data. This means that it affects the displaying of data if IRM is set. Contrast with EBM.

The parameter is specified as one of the following:

<Ps>		Function
ASCII	Hex	
0	30	Edit in display (default)
1	31	Edit in line
2	32	Edit in field (between horizontal tab stops)
3	33	Edit in qualified area (as defined by SPA, EPA, and DAQ)

If the parameter is omitted, edit in display is selected

The SEM field in the ANSI status line is set to D, L, F, or Q for display, line, field, or qualified respectively. The corresponding flag in the status byte is also set.

6.8.7 TSM Tabulation Stop Mode

PARAMETER FOR RM AND SM: 18

TSM determines the action of subsequent tabs. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If TSM is set by SM, the setting and clearing of horizontal tab stops is independent for each line. All lines may have tab stops at different columns.
- If TSM is reset by RM, the setting and clearing of horizontal tab stops apply to the corresponding positions of all lines. All lines will have tab stops in the same columns.

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.8.8 VEM Vertical Editing Mode

PARAMETER FOR RM AND SM: 7

VEM determines how the lines are shifted in editing functions This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If VEM is set by SM, Insert Line (IL) and Delete Line (DL) act upon the current line, and the lines above the current line are moved upward or downward as necessary.
- If VEM is reset by RM, Insert Line (IL) and Delete Line (DL) act upon the current line, and the lines below the current line are moved downward or upward as necessary.

The appropriate indication in the ANSI status line is set or reset along with the corresponding flag in the status byte.

6.9 DEVICE LOCAL MODES

These modes determine how the terminal handles received data.

6.9.1 CRM - Control Representation Mode

PARAMETER FOR RM AND SM: 3

CRM determines whether or not control characters have a displayable representation. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If CRM is set by SM, control characters or sequences are stored in the display memory and are not executed.
- If CRM is reset by RM, control characters or sequences are executed and are not stored in display memory.

6.9.2 KAM - Keyboard Action Mode

PARAMETER FOR RM AND SM: 2

KAM enables or disables the keyboard. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If KAM is set by SM, the keyboard is disabled.
- If KAM is reset by RM, the keyboard is enabled.

6.9.3 SRM - Send-Receive Mode

PARAMETER FOR RM AND SM: 12

SRM specifies whether characters are displayed as they are entered from the keyboard, or only when they are received from the host (essentially auto echo).

- If SRM is set by SM, characters are not displayed as they are keyed in and sent to the host. Only when the characters are echoed from the host are they placed into display memory. (Auto echo is off.)
- If SRM is reset by RM, as characters are entered from the keyboard, they are placed into display memory. (Auto echo is on.)

6.9.4 LNM - Line Feed New Line Mode

PARAMETER FOR RM AND SM: 20

LNM specifies the action performed by the LF (Line Feed) control. This mode is set or reset by means of the SM or RM commands respectively. See the discussion of these commands for the sequence to use this mode.

- If LNM is set by SM, a LF does a CR LF.
- If LNM is reset by RM, a LF does a LF only (no CR action is performed).

6.10 FORMAT EFFECTORS

The primary purpose of format effectors is to be treated as data that happen to have a format representation rather than (or in addition to) a displayable representation. Many format effectors have a corresponding editor function.

6.10.1 BS - Back Space

STRUCTURE: BS
FROM KEYBOARD: BACKSPACE (←) key

The Backspace causes the cursor to move backward one position on the same line. If the current active position is the first position of a qualified area, no action is taken.

6.10.2 CR - Carriage Return

STRUCTURE: CR
FROM KEYBOARD: RETURN (↵) key

The Carriage Return causes the cursor to move to the first position of the active line. If enabled by TCM, this function may perform a carriage return/line feed combination. See the Keyboard Parameters paragraph in Section 3.

The audible alarm sounds on entry to the last line of page if the terminal is in a block transmission mode.

6.10.3 FF - Form Feed

STRUCTURE: FF
FROM KEYBOARD: CTRL L

FF moves the cursor to the same column of the next line.

6.10.4 HPA--Horizontal Position Absolute

STRUCTURE: ESC [<Pn> `

HPA moves the cursor within the active line to the position specified by the parameter Pn. A parameter value of zero or one moves the active position to the first position of the active line. A parameter value of N moves the active position to position N of the active line. If Pn exceeds the maximum defined value for a line, the command is ignored and the UNDEFINED COMMAND bit is set.

6.10.5 HPR--Horizontal Position Relative

STRUCTURE: ESC [<Pn> a

HPR moves the cursor forward the number of positions specified by the parameter Pn. A parameter value of zero or one specifies a single-position move. A parameter value of N indicates an N-position move.

If the move would place the active position beyond the end of the active line, wrap around to following lines and to the beginning of memory occurs if Autowrap is selected. See the Terminal Configuration paragraph in Section 3.

The cursor is allowed outside of qualified areas only when commanded from the host. If a HPR from the keyboard attempts to move the cursor outside of a qualified area, the command is aborted.

6.10.6 HT--Horizontal Tab

STRUCTURE: HT
FROM KEYBOARD: TAB (⟶|) key

HT moves the cursor forward to the next tabulation stop. The tabulation stop may be in a following line. If the next tabulation stop is beyond the end of the memory, the active position moves to the Home position of the page if Autowrap is selected. See the Terminal Configuration paragraph in Section 3.

Qualified areas are also interpreted as tab stops. Thus, if qualified areas are defined, an HT will move the cursor to either the next tab stop or the first position of a qualified area.

6.10.7 HTJ--Horizontal Tab with Justification

STRUCTURE: ESC I

HTJ moves the cursor (and the active position) to the next horizontal tab stop. The characters between the previous tab stop and the previous active position (not including the character at the previous active position) are right justified by space fill to the position immediately prior to the new active position.

If the current active position is a horizontal tab stop, this clears all characters from that position to (but not including) the next tab stop. The active position is moved to the next tab stop.

If qualified areas are defined, justification occurs only up to the end of the current qualified area.

If the next tab stop is beyond the end of memory, the cursor moves to the Home position if Autowrap is selected. See Terminal Configuration in Section 3.

6.10.8 HTS--Horizontal Tab Set

STRUCTURE: ESC H

HTS sets one horizontal tabulation stop at the active position. This sequence is affected by the state of the Tabulation Stop Mode (TSM).

6.10.9 HVP--Horizontal and Vertical Positioning

STRUCTURE: ESC [<Pn> ; <Pn> f

HVP moves the cursor to the position specified by the parameters. The first parameter specifies the vertical position (line) and the second parameter specifies the horizontal position (column). A parameter of zero or one specifies the first line or column. If neither parameter is present, the active position moves to the Home position.

If the line or column parameter is illegal (out of range), this sequence is not executed and the Undefined Command flag is set.

6.10.10 IND--Index

STRUCTURE: ESC D

IND moves the cursor downward one line without changing the horizontal position.

If the current active line is the last line, the active position moves to the first line if Autowrap is selected. See the Terminal Configuration paragraph in Section 3.

6.10.11 LF--Line Feed

STRUCTURE: LF
FROM KEYBOARD: LINE FEED key

LF moves the cursor downward one line without changing the horizontal position. Depending upon the setting of the Line Feed/New Line Mode, this character may act as a new line character (NL), which moves the active position to the first position of the next line.

This sequence is affected by the TCM settings for autowrap, destructive roll and VEM. See the Terminal Configuration paragraph in Section 3.

6.10.12 NEL Next Line

STRUCTURE: ESC E

NEL moves the cursor to the first position on the next line downward. If the current active line is the last line, the active position moves to the first line.

6.10.13 NL New Line

STRUCTURE: NL
FROM KEYBOARD: CTRL J

NL moves the cursor to the first character position of the next line (see Line Feed).

6.10.14 RI Reverse Index

STRUCTURE: ESC M

RI moves the cursor to the same horizontal position on the preceding line.

This sequence is affected by the TCM settings for autowrap, destructive roll and VEM. See the Terminal Configuration paragraph in Section 3.

6.10.15 SGR Select Graphic Rendition

STRUCTURE: ESC [<Ps> m

SGR specifies a visual attribute (the graphic rendition). The selected attribute remains in use until changed by the next SGR. The cursor does not move. SGR specifies a visual attribute, but the attribute does not occupy a screen position.

The legal parameters are:

<Ps>		Function
ASCII	Hex	
0	30	Normal
1	31	High intensity
2	32	Half intensity
4	34	Underscore
5	35	Blink
6	36	Blink
7	37	Reverse video
12	31 32	Security

6.10.16 TBC Tab Clear

STRUCTURE: ESC [<Ps> g

TBC clears tabulation stops according to the specified parameter(s). The following parameters are defined:

<Ps>		Function
ASCII	Hex	
0	30	Clear the horizontal tab stop at the active position
1	31	Clear the vertical tab stop at the active line
2	32	Clear all horizontal tab stops in the active line
3	33	Clear all horizontal tab stops
4	34	Clear all vertical tab stops

6.10.17 VPA Vertical Position Absolute

STRUCTURE: ESC [<Pn> d

VPA moves the cursor to the line specified by the parameter <Pn> without changing horizontal position. A parameter value of 0 or 1 moves the active position to the first line.

This function may move the cursor either upward or downward, depending on the command parameter and the current cursor position.

6.10.18 VPR Vertical Position Relative

STRUCTURE: ESC [<Pn> e

VPR moves the cursor downward the number of lines specified by the parameter <Pn> without changing the horizontal position. A parameter value of 0 or 1 moves the active position downward one line.

Upon reaching the top or bottom of the window or page, this control sequence acts as if <Pn> CUU or CUD control sequences had been received with respect to wrapping to the top of the window or page.

6.10.19 VT Vertical Tab

STRUCTURE: VT
FROM KEYBOARD: CTRL K

VT moves the cursor to the same character position on the next line containing a vertical tab stop. The horizontal position of the cursor is not changed. If a vertical tab stop is not found in the remainder of memory, the active position moves to the last line in the current page.

6.10.20 VTS Vertical Tabulation Set

STRUCTURE: ESC J

VTS sets a vertical tab stop at the active line

6.11 CONTROLS TO MOVE CURSOR

The majority of the sequences described in this section are editor functions. The cursor position report (CPR) sequence is the exception. It is not an editor function because it does not alter the visual arrangement of the display; it is used to report the position the cursor. The primary purpose of an editor function is to edit or alter the visual arrangement of previously entered information.

6.11.1 CBT Cursor Back Tab

STRUCTURE: ESC [<Pn> Z

CBT tabs backwards. The logic scans backwards from the cursor position for either a tab stop or the first (left-most) position of a qualified area. The search continues backwards until the Home position of the current page is encountered. If a tab stop or the first position of a qualified area is found, the cursor is moved to that position and the CBT function is complete. If the scan is not completed successfully at the Home position, the following occurs:

- If the original cursor position was the first position in a qualified area, the audible alarm is sounded, the cursor remains in its original position and the CBT function is complete.

- If the original cursor position was not the first position in a qualified area, the cursor is moved to the Home position of the page and the CBT function is complete.

If the original cursor position is at the beginning of a line or at the Home position and Autowrap (see the Terminal Configuration paragraph in Section 3) is disabled, the audible alarm sounds, the cursor is not moved, and the CBT function is complete. The alarm also sounds if an illegal cursor movement is attempted.

The preceding description applies to a single CBT function. If Pn is not present, zero, or one, then the CBT is executed only once. If Pn specifies a count of greater than one, multiple CBT functions are executed.

6.11.2 CHA Cursor Horizontal Absolute

STRUCTURE: ESC [<Pn> G

CHA positions the cursor the absolute position (column) in the current line as specified by Pn. If Pn specifies an illegal parameter, the CHA sequence is ignored.

6.11.3 CHT Cursor Horizontal Tab

STRUCTURE: ESC [<Pn> I

CHT tabs forwards. The logic scans forwards from the cursor position (if the cursor is in the last position of the current page, the scan starts at the Home position of the current page) looking for either a tab stop or the first (left-most) position of a qualified area. The search continues forwards until the last position of the current page is encountered. If a tab stop or the first position of a qualified area is found, the cursor is moved to that position and the CHT function is complete. If the scan is not completed successfully at the last page position, the following occurs:

- If the original cursor position was the first position in an qualified area, the audible alarm is sounded, the cursor remains in its original position and the CHT function is complete.
- If the original cursor position was not the first position in a qualified area, the cursor is advanced to the Home position of the current page.

If the cursor is at the end of a line or at the last position of the page and Autowrap (see the Terminal Configuration paragraph in Section 3) is disabled, the audible alarm sounds, the cursor is not moved, and the CHT function is complete. The alarm also sounds if an illegal cursor movement is attempted.

The preceding description applies to a single CHT function. If Pn is not present, zero, or one, then the CHT is executed only once. If Pn specifies a count of greater than one, multiple CHT functions are executed.

6.11.4 CNL Cursor Next Line

STRUCTURE: ESC [<Pn> E

CNL moves the cursor to the first column of the next line. If Pn is not specified, or is equal to zero or one, then CNL is executed only once. Otherwise, Pn specifies the number of times that CNL will be repeated. If a parameter is specified which would move the cursor past the bottom of the window, the alarm sounds (if in ANSI mode) and the cursor moves to the first position of the last row in the window.

6.11.5 CPL Cursor Preceding Line

STRUCTURE: ESC [<Pn> F

CPL moves the cursor to the first column of the previous line. If Pn is not specified, or is equal to zero or one, then CPL is executed only once. Otherwise, Pn specifies the number of times that CPL will be repeated. If a parameter is specified which would move the cursor past the top of the window, the alarm sounds (if in ANSI mode) and the cursor moves to the first position of the first row in the window.

6.11.6 CPR Cursor Position Report

STRUCTURE: ESC [<row> ; <column> R
FROM KEYBOARD: None

The CPR sequence is the means by which the terminal reports the position of the cursor. If the cursor is at Home, the "row" and "column" parameters may be omitted.

The CPR sequence is requested by a DCS command sequence, or by a request for a device status report.

6.11.7 CTC Cursor Tabulation Control

STRUCTURE: ESC [<Ps> W
FROM KEYBOARD: TCM tab setting (see Terminal Configuration) and/or softkeys

CTC either sets or clears one or more tabulation stops according to the parameters. When tabs are set or cleared, TSM specifies whether all lines, or just the active line, is affected. The parameters are defined as follows:

<Ps>		Function
ASCII	Hex	
0	30	Set horizontal tab at current position (default)
1	31	Set vertical tab at current line
2	32	Clear horizontal tab at current position
3	33	Clear vertical tab at current line
4	34	Clear all horizontal tabs in current line
5	35	Clear all horizontal tabs in the page
6	36	Clear all vertical tabs in the page

If an illegal parameter is detected, the undefined command flag is set.

6.11.8 CUB Cursor Backward

STRUCTURE: ESC [<Pn> D

CUB moves the cursor backward (to the left). If Pn is not specified, or is equal to zero or one, then the movement is only one character position. Otherwise Pn specifies the number of positions to be moved.

The cursor will move from the first position of the current line to the last position of the previous line only if Autowrap (see the Terminal Configuration paragraph in Section 3) is enabled. If Autowrap is disabled, a CUB command received while the cursor is in the first position of a line causes the cursor to remain stationary and sounds the audible alarm. Likewise, a CUB received while the cursor is at the Home position moves the cursor to the last position of the same page only if Autowrap is enabled. If Autowrap is disabled, a CUB received while the cursor is at the Home position sounds the audible alarm and the cursor does not move. The alarm also sounds if an attempt is made to move the cursor in an illegal direction.

6.11.9. CUD--Cursor Down

STRUCTURE: ESC [<Pn> B

CUD moves the cursor down, without changing its horizontal position. If Pn is not specified, or is equal to zero or one, then the movement is only one line. Otherwise, Pn specifies the number of lines to be moved.

The cursor wraps from the last line in the page to the first line in the same page only if Autowrap (see the Terminal Configuration paragraph in Section 3) is enabled. If Autowrap is disabled, the cursor is on the bottom line, and a CUD is received, the audible alarm is sounded and the cursor does not move. The alarm also sounds if an attempt is made to move the cursor in an illegal direction.

6.11.10. CUF--Cursor Forward

STRUCTURE: ESC [<Pn> C

CUF moves the cursor forward (to the right). If Pn is not specified or is equal to zero or one, then the movement is only one character position. Otherwise, Pn specifies the number of positions to be moved.

The cursor moves from the last position of the current line to the first position of the following line only if Autowrap (see the Terminal Configuration paragraph in Section 3) is enabled. If Autowrap is disabled, a CUF command while the cursor is in the last line position will not move the cursor and will sound the audible alarm.

Likewise, a CUF received while the cursor is at the last position of the page will move the cursor to the page Home position only if Autowrap is enabled. If Autowrap is disabled, a CUF received while the cursor is at the last screen position sounds the audible alarm and the cursor does not move.

6.11.11. CUP--Cursor Position

STRUCTURE: ESC [<row> ; <column> H

CUP moves the cursor to a position relative to the Home position of the current page. The "row" and "column" parameters specify the new cursor

position. If either of the parameters is omitted, the corresponding default (first row or first column) is used.

6.11.12 CUU Cursor Up

STRUCTURE: ESC [<Pn> A

CUU moves the cursor up without changing its horizontal position. If Pn is not specified or is equal to zero or one, then the movement is only one line. Otherwise Pn specifies the number of lines to be moved.

The cursor wraps from the first line in the page to the last line in the same page only if Autowrap (see the Terminal Configuration paragraph in Section 3) is enabled. If Autowrap is disabled, the cursor is on the top line, and a CUU is received, the audible alarm is sounded and the cursor does not move.

6.11.13 CVT Cursor Vertical Tabulation

STRUCTURE: ESC [<Pn> Y

CVT moves the cursor downward to the next line that contains a vertical tab stop. The horizontal position of the cursor is not changed. If Pn is not specified or is equal to zero or one, then a single CVT is executed. Otherwise, Pn specifies the number of CVT commands to execute.

If the current page does not have a line that is beneath the current line that contains a vertical tab stop, one of the two following actions take place:

- If Autowrap (see the Terminal Configuration paragraph in Section 3) is enabled, the cursor wraps around and is placed on the first line of the current page.
- If Autowrap is disabled, the cursor stops on the last line of the current page and the audible alarm sounds.

6.11.14 NP Next Page

STRUCTURE: ESC [<Pn> U
FROM KEYBOARD: SHIFT PAGE ↑

NP causes the data pointer of the source of the command (keyboard or ports) to be moved to a subsequent page. This sequence is used with the BEEAP sequence to align pointers. The Pn parameter specifies how many pages to move forward. If Pn is not specified, or is equal to zero or one, the immediately following page is displayed. Otherwise some number of pages are skipped to arrive at the Nth subsequent page. If NP is received while on the last page, the audible alarm is sounded and the first page is then displayed. In any event, the page indicator in the status line reflects the change.

If the keyboard sequence is used, the cursor moves to the last accessed location on the subsequent page. If the subsequent page has not been previously accessed, the cursor goes to the Home position on the subsequent page.

6.11.15 PP - Preceding Page

STRUCTURE: ESC [<Pn> V
FROM KEYBOARD: SHIFT PAGE ↓

PP causes the data pointer of the source of the command (keyboard or ports) to be moved to a preceding page. This sequence is used with the BEEAP sequence to align pointers. The Pn parameter specifies how many pages to move backward. If Pn is not specified, or is equal to zero or one, the immediately preceding page is displayed. Otherwise some number of pages are skipped to arrive at the Nth prior page. If NP is received while on the first page, the audible alarm is sounded and the last page is then displayed. In any event, the page indicator in the status line reflects the change.

If the keyboard sequence is used, the cursor moves to the last accessed location on the preceding page. If the preceding page has not been previously accessed, the cursor goes to the Home position on the subsequent page.

6.11.16 SL - Scroll Left

STRUCTURE: ESC [<Pn> SPACE @
FROM KEYBOARD: CTRL ← SCROLL

SL functions only when a page or window is wider than the displayed portion of that page or windows. The entire contents of the visible display moves left N columns (as specified by Pn).

The position of the cursor on the screen remains unchanged. Thus, the cursor position in relation to the display page is changed.

6.11.17 SR - Scroll Right

STRUCTURE: ESC [<Pn> SPACE A
FROM KEYBOARD: CTRL → SCROLL

SR functions only when a page or window is wider than the displayed portion of that page or window. The entire contents of the visible display moves right N columns (as specified by Pn).

The position of the cursor on the screen remains unchanged. Thus, the cursor position in relation to the display page is changed.

6.11.18 SU Scroll Up

STRUCTURE: ESC [<Pn> S
FROM KEYBOARD: CTRL ↑ SCROLL

The entire contents of the visible display move upward N lines (as specified by Pn). For each line moved, the top line is removed from sight and another line moves into the last line position.

The position of the cursor on the screen remains unchanged. Thus, the cursor position in relation to the display page is changed.

6.11.19 SD Scroll Down

STRUCTURE: ESC [<Pn> T
FROM KEYBOARD: CTRL ↓ SCROLL

The entire contents of the visible display move downward N lines (as specified by Pn). For each line moved, the bottom line is removed from sight and another line moves into the first line position.

The position of the cursor on the screen remains unchanged. Thus, the cursor position in relation to the display page is changed.

6.12 EDITING TO ALTER VISUAL DISPLAY

The sequences described in this section are editor functions. The primary purpose of an editor function is to edit or alter the visual arrangement of previously entered information.

6.12.1 DCH Delete Character

STRUCTURE: ESC [<Pn> P
FROM KEYBOARD: Softkey

DCH Deletes the character at the cursor location and possibly other adjacent characters, according to the parameter <Pn>. The parameter indicates the number of characters to be deleted, with the adjacent characters shifted toward the cursor position to fill in for the deleted character(s). Spaces are inserted at the end or start of the shifted string. If Pn is omitted or is equal to zero or one, then only one character is deleted.

The positions of selected areas, qualified areas, or tabulation stops are not affected by this command. The effect of this sequence depends on SEM, EBM, and HEM.

6.12.2 DL - Delete Line

STRUCTURE: ESC [<Pn> M
FROM KEYBOARD: Softkey

DL removes the cursor line and possibly adjacent lines. Remaining lines shift to fill the space left by the removed line(s). The cursor moves to the start of the line that replaces the line previously occupied by the cursor. The parameter <Pn> indicates the number of lines to be removed. A parameter of zero indicates that one line is to be removed.

This sequence is affected by VEM and EBM.

6.12.3 EA - Erase in Area

STRUCTURE: ESC [<Ps> O
FROM KEYBOARD: Softkey

EA erases some or all of the characters in the qualified area in which the cursor resides. This erasure is affected by the following parameter <Ps>:

<Ps>		Function
ASCII	Hex	
0	30	Erase from cursor position to end of qualified area, inclusive. (Default)
1	31	Erase from start of qualified area to cursor position, inclusive.
2	31	Erase all of qualified area.

This sequence is affected by ERM and EBM.

6.12.4 ECH - Erase Character

STRUCTURE: ESC [<Pn> X
FROM KEYBOARD: Softkey

ECH erases the character at the active position and possibly other following characters, depending on the parameter Pn. A parameter value of '0' or '1' indicates that only one character is erased. A numeric value of N indicates that N characters are to be erased.

This sequence is affected by ERM and EBM.

6.12.5 ED - Erase in Display

STRUCTURE: ESC [<Ps> J
FROM KEYBOARD: Softkey

ED erases some or all of the characters on the display according to the parameter <Ps>:

<Ps>		Function
ASCII	Hex	
0	30	From cursor position to End of Screen, inclusive (default)
1	31	From start to cursor position, inclusive
2	32	All of display (erasing all of display clears all qualified areas in page)

This sequence is affected by ERM and EBM.

6.12.6 EF - Erase in Field

STRUCTURE: ESC [<Ps> N
FROM KEYBOARD: Softkey

EF erases some or all of the characters in the current field area between the preceding and following horizontal tab stops according to the parameter <Ps>:

<Ps>		Function
ASCII	Hex	
0	30	From cursor position to End of Field, inclusive (default)
1	31	From start of field to cursor position, inclusive
2	32	All of the field

This sequence is affected by ERM and EBM.

6.12.7 EL - Erase in Line

STRUCTURE: ESC [<Ps> K
FROM KEYBOARD: Softkey

EL erases some or all of the characters on the current cursor line according to the parameter <Ps>. The <Ps> parameter can have the following values:

<Ps>		Function
ASCII	Hex	
0	30	Erase cursor position to end of line
1	31	Erase from start of line to cursor position
2	32	Erase all characters on line

This sequence is affected by ERM and EBM.

6.12.8 ICH - Insert Character

STRUCTURE: ESC [<Pn> @
FROM KEYBOARD: Softkey

ICH inserts <Pn> blank character positions at and adjacent to the cursor position. One character at the end of the shifted string is removed for every position shifted.

This sequence is affected by SEM, EEM, and HEM.

6.12.9 IL - Insert Line

STRUCTURE: ESC [<Pn> L
FROM KEYBOARD: Softkey

IL inserts <Pn> blank lines at and adjacent to the cursor position. The cursor remains on the same line as it was previous to the command, but moves to the first column of that line.

This sequence is affected by VEM and EBM.

6.13 FORM DEFINITION

The following sequences are neither format effectors nor editor functions. They are used to define forms by setting logical attributes. There are five sequences in this section. The first starts and ends qualified areas. The next pair starts and ends protected areas. The last pair starts and ends selected areas.

A qualified area is an area that has some sort of restriction on the type of data that is legal in the area. A protected area is an area that may not be modified through keyboard input. A selected area is an area that has been selected for transmission. A guarded area is not normally transmitted. All guarded areas are also protected areas. Refer to Section 5 for a description of the forms definition softkeys. When defining forms with the softkeys, a protected, guarded area is indicated by PROT. A protected unguarded area is indicated by CONTENT.

6.13.1 DAQ - Define Area Qualification

STRUCTURE: ESC [<Ps> o
FROM KEYBOARD: Softkey

DAQ defines the cursor position as the start of a qualified field area. The end of a qualified field area is marked by the start of the following qualified field area. The characteristics of the area are specified according to the following parameter:

<Ps>		Function
ASCII	Hex	
0	30	Accept all input (default)
1	31	Accept no input (protected) and do not transmit (guarded) (equivalent to "Start of Protected Area" command)
2	32	Accept all graphic (printable) data
3	33	Accept all numeric characters
4	34	Accept alphabetic characters
5	35	Right justify in area
6	36	Zero fill in area
7	37	Horizontal tab stop at start of area (this is equivalent to a horizontal tab stop and delimits a field)
8	38	Accept no input (protected) but select for transmission (unguarded) (constant field)
9	39	Space fill in area
10	31 30	Transmit when modified
13	31 33	Numbers only (0-9)
14	31 34	Uppercase (Alpha shift)
16	31 36	Total Fill
19	31 39	Must Enter
20	32 30	Transmit when modified and then exited

Guarded areas may not be transmitted unless GATM is set. A qualified area not specified by a 1 parameter may be transmitted. Refer to: MATM, SATM, and TTM.

6.13.2 SPA Start of Protected Area

STRUCTURE: ESC V
FROM KEYBOARD: Softkey

SPA defines the current cursor position as the start of a string of consecutive character positions which are not alterable manually and are not transmitted in a data stream or transferred to an auxiliary device.

The end of this string of character positions is indicated by the EPA sequence.

6.13.3 EPA End of Protected Area

STRUCTURE: ESC W
FROM KEYBOARD: None

EPA defines the current cursor position as the end of a string of consecutive character positions that are protected against alteration from the keyboard and are not transmitted in a data stream or transferred to an auxiliary device.

This field can be started by the SPA command or the DAQ command with a parameter of 1.

6.13.4 SSA - Start of Selected Area

STRUCTURE: ESC F
FROM KEYBOARD: Softkey

SSA defines the current cursor position as the start of a string of consecutive character positions whose contents are selected to be subsequently transmitted in a data stream or transferred to an auxiliary device.

The end of this string is indicated by either the end of the screen or the ESA command. The cursor position also terminates the string if TTM is reset.

NOTE: The actual transmission is not initiated by this command.

6.13.5 ESA - End of Selected Area

STRUCTURE: ESC G
FROM KEYBOARD: None

ESA defines the current cursor position as the end of a string of consecutive character positions whose contents are selected to be subsequently transmitted in a data stream or transferred to an auxiliary device.

The beginning of this string is indicated by the SSA command.

NOTE: The actual transmission is not initiated by this command

6.14 MISCELLANEOUS CONTROLS

6.14.1 CCH - Cancel Character

STRUCTURE: ESC T

CCH causes the preceding displayable character in the data stream to be ignored.

6.14.2 DA - Device Attributes

STRUCTURE: ESC [<Pn> c

This sequence either requests the receiving device to identify itself or is the requested identifying response. A parameter value of 0 is used by the sender to request the identifying response.

<Pn>		Function
ASCII	Hex	
0	30	Requests the receiving device to identify itself (default)
1	31	Identifies an ATL-008 with 32K of RAM
2	32	Identifies an ATL-008 with 64K of RAM
other		DEC type responses

6.14.3 DMI Disable Manual Input (Disable Keyboard)

STRUCTURE: ESC ` (Grave Accent)

DMI disables the keyboard (including the UNLOCK key). DMI is equivalent to the set state of the Keyboard Action Mode (KAM). The keyboard disable status will be displayed. The audible alarm sounds if any key (other than CONTROL, SHIFT, or LOCK) is pressed while the keyboard is locked.

6.14.4 EMI Enable Manual Input (Enable Keyboard)

STRUCTURE: ESC b
FROM KEYBOARD: None

EMI enables the keyboard for input. EMI is equivalent to the reset state of the Keyboard Action Mode (KAM). The keyboard enable status will be displayed

6.14.5 DSR Device Status Report

STRUCTURE: ESC [<Ps> n

DSR reports or requests a report of the general status of the device according to the parameter <Ps>. Parameter values 0 or 2 report status and parameter values 5 or 6 request status.

A DSR with a parameter value of 0 or 2 may be sent as a response to a requesting DSR (parameter value of 5), as a response to an MW (MESSAGE WAITING), or it may be sent unsolicited.

<Ps>		Function
ASCII	Hex	
0	30	Report: Ready, no malfunctions detected (default)
2	32	Report: Busy, will notify when ready using a DSR control
5	35	Request: Please report status using a DSR control or a DCS string
6	36	Request: Please report active position using a CPR control

6.14.6 INT Interrupt

STRUCTURE: ESC a

INT stops the current processes and performs the following:

- Unlocks keyboard
- Clears receive, transmit, and aux port buffers
- Clears receive, transmit, and print operations
- Momentarily sounds bell

6.14.7 MW Message Waiting

STRUCTURE: ESC U

MW sets a message waiting indicator in the receiving device. This function permits both priority (or overriding) interchange and solicited interchange. Message Waiting is displayed in status line, the message waiting byte is set in the status byte the audible alarm sounds momentarily

The response to this action may be accomplished using a DSR control.

6.14.8 MC Media Copy

STRUCTURE: ESC [<Ps> I

MC controls the transfer of data to and from an auxiliary I/O device as determined by the parameter Ps:

<Ps>		Function
ASCII	Hex	
0	30	Copy screen to primary aux device (default)
1	31	Copy data from primary aux device to screen
4	32	Disable AUX-ON
5	33	Enable AUX-ON

Parameter values 0 and 1 initiate a transfer of data between ATL-008 display memory and an auxiliary I/O device. Parameter values 4 and 5 are modes that disable or enable the auxiliary device's ability to copy subsequently received data from the received data stream to an auxiliary I/O device (AUX-ON).

The received data stream may or may not be displayed depending on the ONLINE/LOCAL setting.

6.14.9 REP Repeat

STRUCTURE: ESC [<Pn> b

REP repeats the single graphic or control character immediately preceding the control sequence. The number of repetitions is specified by the parameter Pn. A parameter value of zero or one repeats the preceding single graphic or control character once.

6.14.10 RIS Reset to Initial State

STRUCTURE: ESC c

RIS resets a device to its initial (power-up) state. This includes:

- Reset tab stops to values in EEPROM
- Reinitialize all I/O buffers
- Reset graphic rendition
- Clear screen
- Move cursor to home
- Reset any programmable modes to values in EEPROM
- Cancel print
- Display primary status line
- Display power-up softkey menu
- Run selftest
- Move cursor and communications pointers to the first allocated page
- Reset status byte indicators
- Momentarily sound alarm

6.15 BLOCK TRANSMISSION

In block transmission modes, transmission will occur if the terminal receives a DECXMIT command from the host or when any of the send keys (RETURN, ENTER, SEND PAGE/LINE) is pressed. If the host needs prior notice of an impending transmission, the STS sequence is provided for this purpose. A user key, programmed to send STS, may be used in such an environment in place of a send key.

6.15.1 DECXMIT Block Transmit

STRUCTURE: ESC 5
KEYBOARD: SEND PAGE/SEND LINE, ENTER, or RETURN

DECXMIT causes transmission of a block of text. The block to be transmitted is determined by the Transmission Mode selected in Terminal Configuration in Installation TCM. If the host must be informed that data is ready prior to transmission, program a user key to send the STS sequence.

6.15.2 STS Set Transmit State

STRUCTURE: ESC S

STS notifies the host that there are data ready for transfer from the terminal. This control does not cause the terminal to initiate a transmission of a data stream. Once the host has determined that data can be transmitted, the transmission must be initiated by a subsequent DECXMIT.

6.16 BEEHIVE EXTENSIONS TO ANSI X3.64

The following sequences have been developed at Beehive International as extensions to the ANSI X3.64 standard sequences.

6.16.1 BEESSKM Set Softkey Menu

STRUCTURE: ESC [<Pn> p

BEESSKM selects a softkey menu for display. See descriptions of the various menus in the Softkey section. The following parameters correspond to the various softkey menus:

<Pn>		Function
ASCII	Hex	
0	30	Primary Menu (Default)
1	31	Primary Menu
2	32	Edit Menu
3	33	Graphic Keys 1 through 7
4	34	Printer Control Menu
5	35	Extended Mode Menu
6	36	Programmer Function Menu
7	37	Reserved
8	38	User Function Key Menu
9	39	Erase Menu
10	31 30	Forms Define Menu
11	31 31	Graphics Keys 9 through 15
12	31 32	Reserved
13	31 33	Reserved
14	31 34	Visual Attributes Menu
15	31 35	Logical Attributes Menu
16	31 36	Additional Logical Attributes Menu
17	31 37	TCM Softkey Menu
18	31 38	TCM Conclude Menu
19	31 39	Reserved
20	32 30	Reserved
21 - up		Available for user-defined menus

This updates identifier for the currently displayed softkey menu in the status byte.

6.16.2 BEEBDS Blank Data Stream

STRUCTURE: ESC [<Ps> z

FUNCTION: Is used to cause the subsequent keyboard data to be either blanked or displayed on the screen, depending on the <Ps> parameter.

<Ps>		Function
ASCII	Hex	
0	30	Unblank the data stream (display keyboard data on the screen). This is the default.
1	31	Blank the data stream. The following keyboard data will be transmitted, but not sent to the display. The terminal will remain in this mode until another BEEBDS is received with a parameter of 30H.

This command is useful when confidential data needs to be entered, but should not be displayed. It also sets or clears the Blank Data Stream flag in the status byte.

6.16.3 BEECM Communications Mode

STRUCTURE: ESC [<Ps> {

The BEECM command sets or resets the various communications modes as follows:

<Ps>		Function
ASCII	Hex	
0	30	Enable ONLINE (disable LOCAL - default condition)
1	31	Disable ONLINE (enable LOCAL)
2	32	Enable AUX ENABLE
3	33	Disable AUX ENABLE
4	34	Enable AUX ON
5	35	Disable AUX ON

ATL-008 operation as a function of its communications modes is shown in the chart below:

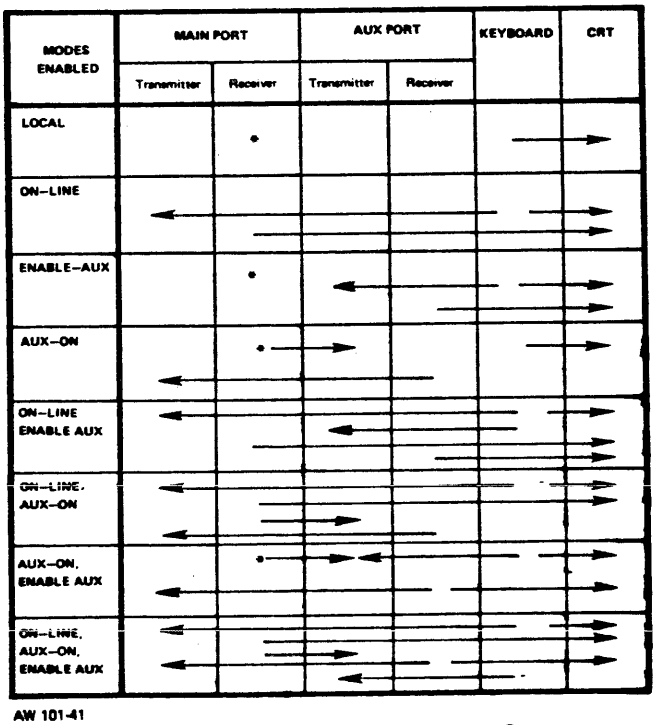


Figure 6-1
I/O Data Flow

The appropriate field in the status line is updated along with the corresponding flag in the status byte.

6.16.4 BEEAP Align Pointer

STRUCTURE: ESC [<Ps> s

BEEAP aligns various position pointers depending on the parameter as follows:

<Ps>		Function
ASCII	Hex	
0	30	Align cursor to Main Data Comm Pointer
1	31	Align cursor to Aux Data Comm Pointer
2	32	Align Main Data Comm Pointer to cursor
3	33	Align Aux Data Comm Pointer to cursor
4	34	Align Main Data Comm Pointer to Aux Data Comm Pointer
5	35	Align Aux Data Comm Pointer to Main Data Comm Pointer

6.16.5 BEESAP Set Absolute Page

STRUCTURE: ESC [<Pn> ; <Pn> t

BEESAP moves the communications pointer for the source of the command (keyboard, aux port, or main port) to the specified page. This command is typically followed by the BEEAP command to align the cursor. There are two parameters needed by this command. The first parameter determines whether the comms pointer will be aligned to the position marker or to the home position. The second parameter specifies the desired page. The first parameter is interpreted as follows:

<Pn>		Function
ASCII	Hex	
0	30	(Default) Position the comm pointer to the window corresponding to the specified page at the position indicated by the position marker
1	31	Identical to 0
2	32	Position the comm pointer to the window corresponding to the specified page at the home position

6.16.6 BEESKL Soft Keyboard Lock

STRUCTURE: ESC [<Pn> u

This sequence sets or clears the keyboard soft lock and the corresponding indicator in the status line. This lock may also be cleared via the UNLOCK key on the keyboard. The parameter is interpreted as follows:

<Pn>		Function
ASCII	Hex	
0	30	Clear soft keyboard lock
1	31	Set soft keyboard lock

6.16.7 BEEMAP Memory Address Pointer

STRUCTURE: ESC [<Ps> v

Each of the data input sources (keyboard, main comm port, and aux comm port) has a corresponding memory address pointer. The memory address pointer may be used as an invisible alternate cursor. When a memory address pointer is active, all subsequent commands and data from the respective source use the memory address pointer (instead of the cursor or comm pointer) until the memory address pointer has been deactivated. During this time the corresponding communications pointer and the cursor are maintained without change. When the memory address pointer is deactivated, cursor and comms pointer functions return to normal.

The parameters are defined as follows:

<Ps>		Function
ASCII	Hex	
0	30	Deactivate the appropriate memory address pointer
1	31	Activate the memory address pointer

6.16.8 BEEBEL Select Audible Alarm

STRUCTURE: ESC [<Ps> w

This command sounds the various audible alarms. The parameter determines the alarm to sound, as follows:

<Ps>		Function
ASCII	Hex	
0	30	Turn all sounds off
1	31	1200 Hz short duration (85 ms)
2	32	1200 Hz continuous
3	33	600 Hz short duration (85 ms)
4	34	600 Hz continuous
5	35	Warble short duration (1 second)
6	36	Warble continuous
7	37	Ring short duration (1 second)
8	38	Ring continuous

6.16.9 BEESND - Send Data To Host

STRUCTURE: ESC [<Ps> | (Vertical Bar)

This command sequence allows data entered into the definition of a user-defined function key to be sent to the host. For example, the following sequence:

ESC [2 | A B C ESC [0 | D E F ESC [2 | G

results in the characters A B C G being executed locally (parameter = 2) and the characters D E F being sent out the main port (parameter = 0), if the terminal is in Character mode (see the description of Transmission mode under Terminal Configuration in Section 3 of this manual) and is Online.

The parameters are interpreted as follows:

<Ps>		Function
ASCII	Hex	
0	30	Transmit if in Character Mode and Online
1	31	Transmit regardless of mode if Online
2	32	Execute locally

The setting of Auto Echo mode (see the description of Auto Echo mode under Main Port Protocol in Section 3 of this manual) determines whether data transmitted out the port will also be written to the screen.

6.17 NON-IMPLEMENTED STANDARD SEQUENCES

The following sequences are part of the ANSI X3.62 standard but are not implemented on the ATL-004. The majority of these do not apply to terminal.

PLD Partial Line Down
PLU Partial Line Up
FNT Font Selection
GSM Graphic Size Modification
GSS Graphic Size Selection
JFY Justify
PUM Positioning Unit Mode
QUAD QUAD
SPI Spacing Increment
TSS Thin Space Specification
PU1 Private Use 1 (ESC Q) *
PU2 Private Use 2 (ESC R) *

* Although the ATL-008 does not respond to these control sequences, they are available as control sequences which may be sent to a host or auxiliary device.

SECTION 7. DEC EMULATION COMMANDS

7.1 INTRODUCTION

The ATL-008 can be operated in two modes, both of which are accessible through TCM:

- ANSI Mode
- 100 mode

In 100 mode, the ATL-008 emulates a DEC VT100 terminal. From 100 mode, and by entering the appropriate escape sequences, the ATL-008 can be made to function in a VT52-compatible mode called 52 Mode. While in either ANSI or 100 mode, the ATL-008 can be attached directly to a serial printer for local printing capabilities. This section addresses all these DEC-compatible aspects.

7.2 100 MODE OPERATION

In this mode, the ATL-008 will emulate a VT100 terminal. The structure and function of the following commands should be familiar to any user of a VT100 terminal.

7.2.1 ATL-008 and VT100 Configuration Procedures

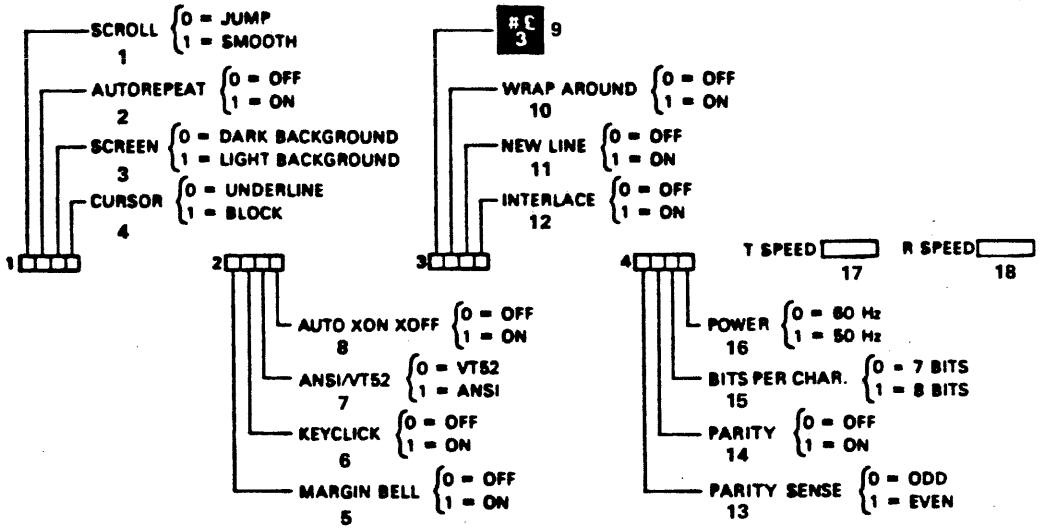
The ATL-008 configuration procedure differs from that of the VT100 terminal. The ATL-008 uses the Terminal Configuration Manager (TCM) program to set terminal parameters such as screen brightness, scroll mode, cursor blinking, etc. Most of the characteristics of the VT100 terminal (and some additional ones) can thus be emulated. Because TCM uses a different method for this procedure, Table 7-1 is provided.

Figure 7-1 illustrates the VT100 set-up summary as it appears at the bottom of a VT100 terminal. Each setting has a reference number. These numbers are duplicated below the illustration with a reference to the corresponding TCM parameter. Parameters that must be set to a specific value are indicated.

Table 7-1 lists other parameters that must be defined to make the ATL-008 emulate a VT100. This table lists the parameters and tells you how to set them.

Exit 100 Mode to ANSI Mode -- The BEEVTL sequence is used to exit from 100 mode to ANSI mode. The sequence is:

ESC . E



#	VT 100 SET-UP Parameters	ATL-008 ICM Menu (Box)
1	Scroll	Video Parameters (Scroll Mode)
2	Autorepeat	Keyboard Parameters (Auto-repeat Rate and Hold Down Delay)
3	Screen	Video Parameters (Background)
4	Cursor	Video Parameters (Cursor Type)
5	Margin Bell	Terminal Configuration (Bell Column)
6	Keyclick	Keyboard Parameters (Keyboard Click)
7	ANSI/VT52	Initial Configuration (Mode - Select 100 Mode. If 52 is desired use the DECANM sequence.)
8	Auto XON/XOFF	Main Port Protocol (Xon/Xoff Protocol - should be set to yes.)
9		Ignored
10	Wrap Around	Terminal Configuration (Autowrap - should be set to no.)
11	New Line	Keyboard Parameters (Newline Key - should be set to CR.)
12	Interlace	Ignored
13	Parity Sense	Main Port Hardware Interface (Parity - select even, odd, or none.)
14	Parity	Main Port Hardware Interface (Parity - select even, odd, or none.)
15	Bits Per Character	Main Port Hardware Interface (I/O Word Format)
16	Power (60 Hz/50 Hz)	Initial Configuration (Line Frequency)
17	T Speed	Main Port Hardware Interface (Baud Rate - Transmit and Receive speeds are equal.)
18	R Speed	Main Port Hardware Interface (Baud Rate - Transmit and Receive speeds are equal.)

Figure 7-1
DEC Emulation Set-Up Table

Table 7-1
VT100 Emulation Parameters

Parameter	Setting
Window size	Must equal the page size. See Screen Page Allocations in Installation TCM.
ANSI Modes: SEM CRM EBM ERM FEAM FETM GATM HEM IRM KAM MATM SATM SRTM TSM TTM VEM	See the section detailing each mode. Line Reset Set Set Reset Reset Set Reset Reset Reset Set Set Reset Reset Reset Reset
Transmission Mode	Set to Character. See Terminal Configuration in Installation TCM.
Overstrike	Set to Disabled. See Initial Configuration in Installation TCM.
Destructive Roll	Set to no. See Terminal Configuration in Installation TCM.
Lower Case Inhibit	Set to no. See Keyboard Parameters in Operator TCM.
Display	Set to half_intensity. See Video Parameters in Operator TCM.
Soft Function Keys and Cursor Keypad	Set to follow_I/O. See Keyboard Parameters in Operator TCM.
Memory Address Pointer	Deactivated. See the BEEMAP sequence.

7.2.2. Terminal Control Commands

Control Characters

The ATL-008 recognizes the same control codes as the VT-100 terminal does, as shown in Table 7-2.

Table 7-2
ATL-008 Control Codes

Control Code	Key Sequence	Hex Code	Action taken
ENQ	CTRL E	05H	Transmits answerback message (if enabled in TCM).
BEL	CTRL G	07H	Sounds bell tone from keyboard.
BS	CTRL H	08H	Moves the cursor one character to the left, unless it is at the left margin, in which case no action occurs.
HT	CTRL I	09H	Moves the cursor to the next tab stop, or to the right margin if no further tab stops are present.
LF	CTRL J	0AH	Causes a line feed or a new line operation.
VT	CTRL K	0BH	Performs the same function as LF.
FF	CTRL L	0CH	Performs the same function as LF.
CR	CTRL M	0DH	Moves the cursor to column 1 of the current line.
SO	CTRL N	0EH	Causes subsequent characters to be taken from the current alternate (G1) character set.
SI	CTRL O	0FH	Causes subsequent characters to be taken from the current regular (G0) character set.
XOFF	CTRL S	11H	Stops transmission to the host or printer.
XON	CTRL Q	13H	Resumes transmission to the host or printer.
CAN	CTRL X	18H	Terminates the current control sequence.
SUB	CTRL Z	1AH	Performs the same function as CAN.
ESC	CTRL I	1BH	Introduces a control sequence

Control Sequences

The ATL-008 also recognizes ANSI control sequences. ANSI sequences are described in Section 6. Refer to that section for information on the structure and parameters of ANSI sequences.

7.2.3 Set Mode and Reset Mode

Many terminal modes have two logical states; for example, on/off or either/or. Set mode and reset mode are used to select the state of these modes.

Set Mode (SM)

```
ESC [ <Ps> ; ... ; <Ps> h
```

Default Value: None

To set several modes at one time, enter a separate parameter (from the following list) for each mode. A mode remains set until a reset mode (RM) control sequence resets it. For explanations of these modes, see their various sections.

Parameter (<Ps>)	Mode	Function
0		Error
? 1	DECCKM	Cursor Key
? 2	DECANM	100/52
? 3	DECCOLM	Column
? 4	DECSCLM	Scrolling
4	IRM	Insert/Replace
? 5	DECSCNM	Screen
? 6	DECOM	Origin
? 7	DECAWM	Auto Wrap
? 8	DECARM	Auto Repeat
? 9	DECINLM	Interlace
? 18	DECFF	Form Feed
? 19	DECEXT	Print Extent
20	LNM	Line Feed/New Line

Reset Mode (RM)

```
ESC [ <Ps> ; <Ps> ; ... ; <Ps> l
```

Default Value: None

Enter a separate parameter for each mode to be reset, as in the set mode (SM) control sequence.

7.2.9. Cursor Movement Functions

Cursor Up (CUU)

ESC [<Pn> A
Default Value: 1

Moves the cursor upward. The column position remains the same. The cursor cannot be moved above the top margin.

Parameter (Pn)	Meaning
0,1	Move cursor one line upward
n	Move cursor n lines upward

Cursor Down (CUD)

ESC [<Pn> B
Default Value: 1

Moves the cursor downward. The column position remains the same. The cursor cannot be moved below the bottom margin with this function.

Parameter (Pn)	Meaning
0,1	Move cursor down one line
n	Move cursor down n lines

Cursor Forward (CUF)

ESC [<Pn> C
Default Value: 1

Moves the cursor to the right. The cursor cannot be moved past the right side of the screen.

Parameter (Pn)	Meaning
0,1	Move cursor one space to the right
n	Move cursor n spaces to the right

Cursor Backward (CUB)

ESC [<Pn> D
Default Value: 1

Moves the cursor to the left. The cursor cannot be moved to the left of the left side of the screen.

Parameter (Pn)	Meaning
0,1	Move cursor one space to the left
n	Move cursor n spaces to the left

7.2.5 Cursor Addressing

Cursor Position (CUP)

ESC [<Pn> ; <Pn> H
Default Values: 1,1

Moves the cursor to a specified position.

Parameter (Pn)	Meaning
First parameter	
0,1	Move cursor to first line in the display
n	Move cursor to nth line in the display
Second parameter	
0,1	Move cursor to first column in the display
n	Move cursor to nth column in the display

If both parameters are omitted (default condition), the cursor returns to the home position. This function is identical to HVP.

The current setting of the origin mode (DECOM) determines how lines are numbered.

Horizontal and Vertical Position (HVP)

ESC [<Pn> ; <Pn> f
Default Values: 1,1

Moves the cursor to a specified position. (See Cursor Position.) This function is identical to CUP.

Cursor Position Report (CPR)

ESC [<Pn> ; <Pn> R
Default Values: 1,1

When the host sends a device status report with a parameter of 6, the terminal sends a cursor position report.

The first parameter specifies the line number of the current cursor location. The second specifies the column. The current setting of the origin mode (DECOM) determines how lines are numbered.

If both parameters are 1 or if no parameters are sent, the cursor is at home position.

7.2.6 Save and Restore Cursor

Save Cursor (DECSC)

ESC 7

This sequence saves the cursor position, graphic rendition, and character set.

Restore Cursor (DECRC)

ESC 8

This sequence restores the cursor position, graphic rendition, and character set previously saved by DECSC.

7.2.7 Line Feed/New Line Mode, Next Line, Index, Reverse Index

Line Feed/New Line Mode (LNM)

ESC [20 h (New Line) (To set)
ESC [20 l (Line Feed) (To reset)

This parameter uses set mode (SM) and reset mode (RM) to specify the effect of a line feed (LF), which is defined in ANSI Standard X3.4-1977.

Reset Mode	
Line feed received	Cursor moves down one line, staying in same column
RETURN key	Sends single code: CR

Set Mode	
Line feed received	Cursor moves to first position of next line
RETURN key	Sends two codes: CR and LF

The setting of LNM does not affect the index (IND) or next line (NEL) functions.

Next Line (NEL)

ESC E

Moves the cursor to the first position on the following line. When the cursor is at the bottom margin, the display scrolls up. If the bottom margin is not the bottom of the window, the cursor may be positioned below the bottom margin. In such a case, if the cursor is at the bottom of the window, no scrolling occurs.

Index (IND)

ESC D

Moves the cursor down one line. The column position remains the same. When the cursor is at the bottom margin, the display scrolls up. If the bottom margin is not the bottom of the window, the cursor may be positioned below the bottom margin. In such a case, if the cursor is at the bottom of the window, no scrolling occurs.

Reverse Index (RI)

ESC M

Moves the cursor to the same position on the preceding line. If the cursor is at the top margin, the display scrolls down. If the top margin is not the top of the window, the cursor may be positioned above the top margin. In such a case, if the cursor is at the top of the window, no scrolling occurs.

7.2.8 Tab Functions

Horizontal Tabulation Set (HTS)

ESC H

Sets a horizontal tab stop at the current cursor position.

Tabulation Clear (TBC)

ESC [<Ps> g
Default Value: 0

Clears tab stops.

Parameter (<Ps>)	Meaning
0	Clear the horizontal tab stop at the active position
1	Clear the vertical tab stop at the active line
2	Clear all horizontal tab stops in the active line
3	Clear all horizontal tab stops
4	Clear all vertical tab stops

No other parameter values have meaning in this function.

Back Tab

ESC [Z
Back tab (⟵) key

This function is performed only when the back tab key is pressed. If the terminal is in local mode, the cursor moves to the previous tab stop on the current line. If there is no tab stop between the beginning of the line and the active position, the cursor moves to the first position on the current line. If the terminal is on-line, a CUB command is sent with the parameter being the number of character positions to the previous tab stop or the beginning of the line, whichever is less.

7.2.9 Scrolling Modes

Scrolling Mode (DECSCLM)

ESC [? 4 h (TCM selected rate) (To set)
ESC [? 4 l (Instantaneous) (To reset)

Determines scrolling speed and type. Set causes smooth scrolling to proceed at a rate selected by TCM. Reset causes instantaneous or jump scrolling.

7.2.10 Clear or Erase Commands

Erase in Display (ED)

ESC [<Ps> J
Default Value: 0

Erases some or all of the characters in the display.

Parameter (<Ps>)	Meaning
0	The screen will be erased from the cursor position to the end of the screen, inclusive (default). Row attributes are not cleared.
1	The screen will be erased from the beginning of the screen to the cursor position. Row attributes are not cleared.
2	The entire display is erased. All lines are changed to single-width. The cursor position remains unchanged. All row attributes are cleared.

Erase in Line (EL)

ESC [<Ps> K
Default Value: 0

Erases some or all of the characters in the current line. Row attributes are not cleared.

Parameter (<Ps>)	Meaning
0	The line will be erased from the cursor position to the end of the line (default).
1	The line will be erased from the beginning of the line to the cursor position.
2	The entire line will be erased.

7.2.11 Edit Commands

Insert/Replacement Mode (IRM)

ESC [4 h (Insert) (To set)
ESC [4 l (Replace) (To reset)

Uses set mode (SM) and reset mode (RM). SM activates insert mode, which causes a message to be displayed in the status line. Incoming characters are placed at the cursor position, with all other characters shifted to the right. Characters shifted past the right side of the screen are lost.

RM activates replacement mode; the message is removed from the status line. Incoming characters replace the character at the cursor position and the cursor advances one character position. Characters outside of the cursor position are not affected.

This mode is affected by the settings of ANSI modes SEM and HEM.

Delete Character (DCH)

ESC [<Pn> P
Default value: 1

Deletes <Pn> characters starting at the cursor position. Characters to the right of the cursor are shifted <Pn> spaces to the left. This command is affected by the settings of ANSI modes SEM and HEM.

Insert Line (IL)

ESC [<Pn> L
Default value: 1

Inserts <Pn> lines at the cursor location. These lines are filled with spaces which have the same graphic rendition as characters on the preceding line. Lines scrolled off the screen are lost. The cursor must be within the scrolling area when this command is executed. This command is affected by the setting of ANSI mode VEM.

Delete Line (DL)

ESC [<Pn> M
Default value: 1

Deletes <Pn> lines, including the current line. The rest of the screen is moved up <Pn> lines. New lines are moved up from the bottom margin of the scrolling area. New lines filled with spaces which have the same graphic rendition as characters on the preceding line are inserted at the bottom of the screen. The cursor must be within the scrolling area when this command is executed. This command is affected by the setting of ANSI mode VEM.

7.2.12 Reset Function

Reset to Initial State (RIS)

ESC c

Executes the power-up selftest and returns the terminal to its initial state, during which the screen is cleared.

7.2.13 Screen Control

Screen Alignment Display (DECALN)

ESC # 8

The screen, except the last three lines, will be filled with uppercase E's. Anything typed thereafter will overprint the E's.

Column Mode (DECCOLM)

ESC [? h	(132-column) (To set)
ESC [? 3 l	(80-column) (To reset)

Set mode (SM) and reset mode (RM) are used to determine the maximum number of columns on the screen. SM causes the display to be 132 columns wide. RM causes the screen display to be 80 columns wide.

Interlace Mode (DECINLM)

ESC [? 9 h	(Ignored) (To set)
ESC [? 9 l	(Ignored) (To reset)

This function is accepted but ignored. It is provided only for compatibility with software written for other systems.

Screen Mode (DECSCNM)

ESC [5 h	(White screen/black characters) (To set)
ESC [5 l	(Black screen/white characters) (To reset)

Set mode (SM) and reset mode (RM) are used to determine the type of display. RM causes a dark screen with light characters. SM state causes a light screen with dark characters.

7.2.14 Visual Attributes

Double-High/Double-Wide (DECDHL)

ESC # 3	(Top Half)
ESC # 4	(Bottom Half)

Double-high and double-wide rows are displayed by setting a double-high/ double-wide top row attribute on one line and a double-high/double-wide bottom row attribute on the next line down. Data on both lines must be the same.

Only half as many characters are available on a double-wide line as on a single-wide line. If a single-wide line that is already on the screen is converted to double-wide, all characters to the right of center screen are not displayed but are retained. The cursor remains over the same character unless the character is moved past the right side of the screen. In this case, the cursor moves to the extreme right side of the screen.

Double-Wide Line (DECDWL)

ESC # 6

This function causes the cursor line to become double-width/single-height. Only half as many characters are available on a double-wide line as on a single-wide line. If a single-wide line that is already on the screen is converted to double-wide, all characters to the right of center screen are not displayed but are retained. The cursor remains over the same character unless the character is moved past the right side of the screen. In this case, the cursor moves to the extreme right side of the screen.

Single-Wide Line (DECSWL)

ESC # 5

This function converts the cursor line to single-width/single-height, the default condition for all new lines on the screen. The cursor moves to stay on the same character position.

7.2.15 Display Locking

Set Top and Bottom Margins (DECSTBM)

ESC [<Pn> ; <Pn> r
Default values: No margins
(scrolling area is entire screen)

The top and bottom margins of the scrolling area are set.

Parameter (Pn)	Meaning
First parameter	Line number of top line in scrolling area
Second parameter	Line number of bottom line in scrolling area.

The top margin line number must be smaller than the bottom margin line number, making the minimum scrolling area two lines high. When this function is executed, the cursor goes to the home position (defined by origin mode DECOM).

7.2.16 Extended Character Set

Select Character Set (SCS)

The ATL-008 terminal can display up to 256 characters. However, in 100 mode only the following characters are available:

The 128 characters of the standard ASCII character set
Foreign characters (if a foreign character set has been enabled in TCM)
The Special Graphics characters listed in table 7-3.

There are two character sets available at any one time: the regular character set, called G0, and the alternate character set, called G1. At power-up, both of these are set to the character set defined by the Nationality selection in Initial Configuration in Installation TCM. Either G0 or G1 can be set as any of the three choices (ASCII, foreign, special graphics) by the Select Character Set (SCS) sequence. The specified character set remains in use until another SCS is executed. The sequences used to specify the character sets are listed in Table 7-3.

When entering data, G0 is the default character set. To access characters in G1, use a shift-in (SI, CTRL_O, 0FH) code. To return to accessing characters in G0, use a shift-out (SO, CTRL_N, 0EH) code.

When the special graphics character set is used, the displayable characters for the codes 6FH to 7EH are replaced with the special graphic characters in the Table 7-4.

To successfully print multiple character sets, the serial printer must be able to process the select character sequence (SCS) and the shift in (SI) and shift out (SI) characters.

Table 7-3
Character Set Sequences

G0 Sets Sequence	G1 Sets Sequence	Set Names
ESC (A	ESC) A	TCM selected foreign character set
ESC (B	ESC) B	ASCII set
ESC (0	ESC) 0	Special graphics

Table 7-4
Special Graphics Characters

HEX	REG	Graphics Character	CG LOC	HEX	REG	Graphics Character	CG LOC
5F	_	Blank	20	6F	o	⎺ Horizontal Line Scan 3	CC
60	`	◆ Diamond	DE	70	p	⎻ Horizontal Line Scan 5	CD
61	a	▒ Checkerboard	7F	71	q	─ Horizontal Line Scan 7	F8
62	b	␉ Horizontal Tab	09	72	r	⎽ Horizontal Line Scan 9	CE
63	c	␌ Form Feed	0C	73	s	_ Horizontal Line Scan 11	CF
64	d	␍ Carriage Return	0D	74	t	├ Left "T"	F6
65	e	␊ Line Feed	0A	75	u	┤ Right "T"	F5
66	f	° Degree Symbol	9A	76	v	┴ Bottom "T"	F7
67	g	± Plus/Minus	B8	77	w	┬ Top "T"	F4
68	h	␤ New Line	BC	78	x	│ Vertical Bar	F9
69	i	␋ Vertical Tab	0B	79	y	≤ Less Than or Equal	B9
6A	j	┘ Lower-right corner	F3	7A	z	≥ Greater Than or Equal	BB
6B	k	┐ Upper-right corner	F1	7B	{	π Pi	EF
6C	l	┌ Upper-left corner	F0	7C	|	≠ Not Equal To	BA
6D	m	└ Lower-left corner	F2	7D	}	£ UK Pound Sign	9D
6E	n	┼ Crossing Lines	FA	7E	~	· Centered Dot	D0

Select Graphic Rendition (SGR)

ESC [<Ps> ; ... ; <Ps> m
Default Value: 0

Defines the graphic rendition for the displayable characters which follow it. All characters sent to the terminal will appear as specified by the parameter(s) until the next SGR occurs.

Parameter	Meaning
0	Attributes off
1	Intensified (Bold) *
2	Half intensity *
4	Underscored
5	Blink
7	Reverse video

* These selections interact with the Video Display parameter in Operator TCM. For example, Bold only works if half intensity is selected in TCM. No other parameter values have meaning.

Existing attributes are affected by subsequent changes in TCM.

7.2.17 Selftest Diagnostics

Invoke Confidence Test (DECTST)

ESC [2 ; <Ps> y
Default value: 1

This sequence causes execution of the power-up selftest. The parameter is ignored.

7.2.18 Status Report Functions

Device Attributes (DA)

ESC [c
ESC [1 c

The host sends the terminal a DA control sequence without parameters or with a parameter of 1. This prompts the terminal to respond with a device attributes (DA) control sequence which will identify the terminal to the host.

The terminal responds to the above prompt by sending the following DA control sequence:

ESC [? 1 ; 11 c

7.2.19 Identify Terminal (DECID)

ESC Z

The DECID sequence has been used in the past to request a terminal to identify itself. In 100 mode, the ATL-008 will respond to a DECID in the same way as it responds to a DA request.

The DECID sequence is no longer used. All new software should use DA to request terminal identity.

7.2.20 Report Terminal Parameters (DECREPTPARM)

ESC [<request type> x	(Request)
ESC [<request type> <response string> x	(Response)

Sent upon receipt of a DECREQTPARM sequence from the host. It conveys the status of selected terminal parameters. Table 7-5 describes the various parameters.

Table 7-5
Report Terminal Parameters

Parameters	Value	Description
<request type>	0 or none	This is a request (DECREQTPARM). The terminal will be allowed to respond at will after the terminal exits the TCM SET-UP mode.
	1	This message is a request. The terminal must wait for a request before it responds.
	2	This message is a report.
	3	This message is a report. The terminal is responding to a request.
<Main parity>	1	No parity set for main port. Parity bit removed from character.
	2	Space parity set for main port.
	3	Mark parity set for main port.
	4	Odd parity set for main port.
	5	Even parity set for main port.
<Main bits>	1	8 data bits per character.
	2	7 data bits per character.

Parameters	Value	Description
<Main TX speed>	0	50
	8	75
	16	110
	24	134.5
	32	150 Bits per second
	40	200
	48	300
	56	600
	64	1200
	72	1800
	80	2000
	88	2400
	104	4800
	108	7200
	112	9600
	120	19200
<Main RX speed>	(Same values as main TX speed parameter)	
<Mult>	1	Bit rate multiplier is 16
<Reserved>	0	Always zero (0)
<Mode>	0	52 mode
	1	100 mode
<Aux parity>	1	No parity set for auxiliary port. Parity bit removed from character.
	2	Space parity set for auxiliary port.
	3	Mark parity set for auxiliary port.
	4	Odd parity set for auxiliary port.
	5	Even parity set for auxiliary port.
<Aux bits>	1	8 data bits per character.
	2	7 data bits per character.

Parameters	Value	Description
<Aux TX speed>	0	50
	8	75
	16	110
	24	134.5
	32	150 Bits per second
	40	200
	48	300
	56	600
	64	1200
	72	1800
	80	2000
	88	2400
	104	4800
	108	7200
	112	9600
	120	19200
<Aux RX speed>	(Split speeds not supported on the auxiliary port.)	

Device Status Report (DSR)

ESC [<Ps> n Default Value: 0

Terminal status may be requested by the host and reported by the terminal using the following parameters:

Parameter	Meaning
0	Ready response from terminal.
3	Not ready response. Possible malfunction. Try again.
5	Request for status report (using a DSR control sequence).
6	Request for active position report (using a CPR control sequence).

A requesting DSR with a parameter value of 5 will always receive a responding DSR with a parameter value of 0 or 3. A requesting DSR with a parameter value of 6 will always receive a cursor position report (CPR) in response.

7.2.21 Miscellaneous Functions

100/52 Mode (DECANM)

ESC [? 2 h (100 Mode) (To set) Not valid while in 52 mode.
ESC [? 2 l (52 Mode) (To reset)

This parameter uses set mode (SM) and reset mode (RM) to change between 100 mode and 52 mode. RM specifies 52 mode. SM specifies 100 mode.

Load LEDS (DECLL)

ESC [<Ps> q Default Value: 0

This sequence allows the host to turn the L indicators in the status line on or off. For example, if L1 is turned on, the message L1 appears in field H of the status line.

Indicator L3 is not available if the AC option is enabled.

Parameter	Meaning:
0	Clear all L indicators
1	Turn L1 indicator on
2	Turn L2 indicator on
3	Turn L3 indicator on
4	Turn L4 indicator on

Origin Mode (DECOM)

ESC [? 6 h (Relative to margins) (To set)
ESC [? 6 l (Independent of margins) (To reset)

This parameter determines cursor position and the relation of line and column numbers to the margin using set mode (SM) and reset mode (RM) control sequences.

RM puts the origin of the line and column numbers at the upper-left character position on the screen, causing line and column numbers to be independent of current margin settings. Cursor position (CUP) or horizontal and vertical position (HVP) control may be used to position the cursor outside the margins of the scrolling area.

SM state puts the origin of the line and column numbers at the upper-left character position inside the scrolling area, causing line and column numbers to be relative to the current margin settings. The cursor may not be positioned outside the margins.

When this mode is set or reset, the cursor returns to the new home position.

The origin is line 1, column 1. Lines and columns are numbered consecutively.

7.2.22 Keyboard Functions

Auto Repeat Mode (DECARM)

ESC [? 8 h	(Selected keys auto repeat) (To set)
ESC [? 8 l	(No keys auto repeat) (To reset)

This parameter uses set mode (SM) and reset mode (RM) control sequences to toggle the keyboard auto-repeat function. RM does not allow auto-repeat. SM allows certain keys to auto-repeat.

The following keys do not repeat: SELECT, all PF keys, shift, shift lock, CTRL, ESC, SOE, HOLD SCROLL, UNLOCK, CONFIG, SEND PAGE/LINE, keypad minus, keypad comma, keypad period, and ENTER.

Autowrap Mode (DECAWM)

ESC [? 7 h	(Wrap to new line) (To set)
ESC [? 7 l	(Write over last character) (To reset)

This parameter determines whether displayable characters sent when the cursor is on the right side of the screen will write over previous characters at the same position or will be written at the start of a new line, depending on the use of set mode (SM) and reset mode (RM) control sequences. RM results in writing over the previous character. SM results in the start of a new line.

Keypad Application Mode (DECKPAM)

ESC =

When this mode is enabled, the terminal sends special escape sequences when the keys on the keypad are struck rather than the code corresponding to the legend on the keycap.

Key	Code Sent	Key	Code Sent
0	ESC O p	7	ESC O w
1	ESC O q	8	ESC O x
2	ESC O r	9	ESC O y
3	ESC O s	-	ESC O m
4	ESC O t	,	ESC O l
5	ESC O u	.	ESC O n
6	ESC O v	ENTER	ESC O M

Keypad Numeric Mode (DECKPNM)

ESC >

When this mode is enabled, the terminal sends the code sequences corresponding to the legends on the keycaps.

Cursor Keys Mode (DECCKM)

ESC [? 1 h	(Application functions) (To set)
ESC [? 1 l	(Cursor control commands) (To reset)

This parameter determines whether the cursor function keys send cursor control commands or application functions, depending on the use of set mode (SM) and reset mode (RM) control sequences. When the cursor key mode is set, the four cursor function keys can send application functions. If the cursor key mode is reset, the four cursor function keys can send cursor control commands that apply in 100 mode .

The keyboard must be in keyboard application mode to perform this function.

Key	Reset Mode	Set Mode
Up arrow (↑)	ESC [A	ESC O A
Down arrow (↓)	ESC [B	ESC O B
Right arrow (→)	ESC [C	ESC O C
Left arrow (←)	ESC [D	ESC O D
Home (↖)	ESC [H	ESC O E

7.3 52 MODE OPERATION

In this mode the ATL-008 emulates a VT100 terminal emulating a VT52 terminal.

NOTE: If you use set mode (SM) or reset mode (RM) to set parameters in 100 mode, then use the DECANM command to enter 52 mode, the parameters set in 100 mode remain unchanged.

7.3.1 Cursor Movement Functions

Cursor Up

ESC A

The cursor position moves up one line while remaining in the same column. The cursor cannot move above the top margin.

Cursor Down

ESC B

The cursor position moves down one line while remaining in the same column. The cursor cannot move below the bottom margin.

Cursor Right

ESC C

The cursor is moved one position to the right. The cursor cannot move past the right side of the screen.

Cursor Left

ESC D

The cursor is moved one position to the left. The cursor cannot move past the left side of the screen.

Cursor to Home

ESC H

The cursor returns to the home position at the upper left corner of the screen.

Reverse Line Feed

ESC I

This results in cursor up unless the cursor is at the top margin, in which case a scroll down is performed.

Direct Cursor Address

ESC Y <line> <column>

The cursor moves to the line and column specified. See Table 7-6, Cursor Addressing, for the ASCII characters used to specify the line and column number.

Table 7-6
Absolute Cursor Addresses
Addressing Offset = 20H

Row or Column Number	ASCII Character	Row or Column Number	ASCII Character	Row or Column Number	ASCII Character
1	Space	28	;	55	V
2	!	29	<	56	W
3	"	30	=	57	X
4	#	31	>	58	Y
5	$	32	?	59	Z
6	%	33	@	60	[
7	&	34	A	61	
8	'	35	B	62	]
9	(	36	C	63	^
10	)	37	D	64	_
11	*	38	E	65	`
12	+	39	F	66	a
13	,	40	G	67	b
14	-	41	H	68	c
15	.	42	I	69	d
16	/	43	J	70	e
17	0	44	K	71	f
18	1	45	L	72	g
19	2	46	M	73	h
20	3	47	N	74	i
21	4	48	O	75	j
22	5	49	P	76	k
23	6	50	Q	77	l
24	7	51	R	78	m
25	8	52	S	79	n
26	9	53	T	80	o
27	:	54	U		

132-Column Cursor Addressing - If the terminal is in 132-column mode and the cursor needs to be addressed to a column beyond column 80, you should set bit 6 of the row parameter.

7.3.2 Clear or Erase Commands

Erase to End of Screen

ESC J

All characters right of the cursor to the end of the screen are deleted. Cursor position remains the same.

Erase to End of Line

ESC K

All characters right of the cursor to the end of the current line are deleted. Cursor position remains the same.

7.3.3 Extended Character Set

On entering 52 mode from 100 mode, the last character set selected in 100 mode remains valid until one of the commands in this section changes it.

Enter Graphics Mode

ESC F

The special graphics character set is specified.

Exit Graphics Mode

ESC G

The standard ASCII character set is specified.

7.3.4 Status Report Functions

Identify

ESC Z

To request terminal identity, the host sends an ESC Z.

The terminal responds by sending its identifier escape sequence to the host. This sequence is:

ESC / Z

7.3.5 Miscellaneous Functions

Enter 100 Mode

ESC <

Further 52 mode escape sequences are not recognized; all subsequent escape sequences will be interpreted as described in Section 6.

7.3.6 Keyboard Functions

Enter Alternate Keypad Mode

ESC =

Additional escape sequences (see Table 7-7) for use by applications programs are sent.

Table 7-7
Additional Escape Sequences

Key	Code Sent	Key	Code Sent
0	ESC ? p	7	ESC ? w
1	ESC ? q	8	ESC ? x
2	ESC ? r	9	ESC ? y
3	ESC ? s	-	ESC ? m
4	ESC ? t	,	ESC ? l
5	ESC ? u	.	ESC ? n
6	ESC ? v	ENTER	ESC ? M

Exit Alternate Keypad Mode

ESC >

The ASCII codes for the functions or characters engraved on the key are sent.

7.4 AC MODE PRINT OPERATIONS

The ATL-008 uses four print operations. Two of these (printer controller mode and the print cursor line function) may be selected by a host. The other two print operations (auto print mode and the print screen function) are available from either the keyboard or the host.

Only one print operation can be performed at a time. The current print operation must be terminated before a new one is selected.

Before selecting a print mode or print function, the status of the printer must be checked. In 100 mode, use the printer status report (DECPSR) sequence to check printer status. If you select a print operation while the serial printer is not ready, the terminal may hang until the print operation is completed. If the print operation cannot be completed, you must reset the terminal using either a RIS sequence or by issuing a reset from the keyboard.

To select a print operation:

1. Turn off auto print mode if it has been selected from the keyboard.
2. Select 100 mode and check to see if a printer is installed and operational by requesting a printer status report (DECPSR).
3. Return the terminal to 52 mode if it was operating in that mode.
4. Select the print operation.

7.4.1 Host-Selectable Print Operations

Printer Controller Mode - When printer controller mode is selected by the host computer, all characters received from the host computer are sent to the serial printer without being converted or altered in any way. This mode performs a transparent print. Communication from the terminal will be received by the host while printer controller mode is in operation.

The terminal does not automatically suppress or insert spaces, insert line-delimiter characters, or select alternate character sets while in printer controller mode. Printer tab stops must be properly set if tab control characters are used. The host can set the printer tab stops on some printers.

Always place the active column position of the printer at the left margin before leaving printer controller mode.

100 MODE	52 MODE	MEANING
ESC [5 i	ESC W	(select)
ESC [4 i	ESC X	(terminate)

Print Cursor Line Function - The host computer can select the print cursor line function. The screen line which contains the cursor is printed. The cursor position remains unchanged. Once the function is selected, the screen line which contains the cursor cannot be changed until it is printed.

The host cannot receive data from the terminal while a print cursor line function is in progress. Print cursor line automatically terminates and starts sending keyboard entries after the line is printed. Data entered at the terminal is stored in a buffer until the function is completed.

100 MODE	52 MODE
ESC [? 1 i	ESC V

7.4.2 Host or Keyboard Selectable Print Operations

Auto Print Mode - Auto print causes the screen display to be printed one line at a time. As the cursor moves off the line because of a line feed, form feed, or vertical tab character, the line is automatically printed.

Once the cursor moves off of a line, the line cannot be changed before it is printed. The host computer does not receive any terminal keyboard entries until the line is printed. Keyboard entries are stored in a buffer until the line is printed.

While the terminal is in edit mode, auto print is suspended. Auto print is the only print operation affected by edit mode.

100 MODE	52 MODE	MEANING
ESC [? 5 i	ESC ^	(select)
ESC [? 4 i	ESC -	(terminate)

Print Screen Function - This function prints either the entire screen or the current scrolling area, depending on the setting of the print extent selection. It may be selected by either the terminal or the host computer.

Once print screen is selected, the screen contents cannot be changed until printed . Keyboard entries are not received by the host computer while the screen display is being printed.

Print screen automatically terminates and starts sending keyboard entries after the screen is printed. Keyboard entries are stored in a buffer until the operation is complete.

100 MODE	52 MODE
ESC [i	ESC]

7.4.3 Character Conversions

In all modes/functions except printer controller mode, tabs are converted to spaces and trailing spaces are suppressed.

Double-high characters are printed as two identical lines of standard width characters. Double-wide characters are printed as one line of standard width characters.

7.4.4 Print Extent Mode (DECEXI)

During a print screen function, either all the characters on the screen or only the characters within the scrolling area will be sent to the printer. Set mode (SM) and reset mode (RM) are used to determine which option takes place. SM causes all characters on the screen to be sent to the printer. RM causes only the characters within the scrolling region to be sent. Use the DECSTBM sequence in 100 mode to select the top and bottom margins of the scrolling region.

To set:	ESC [19 h	(full screen)
To reset:	ESC [19 l	(scrolling region)

7.4.5 Form Feed Mode (DECFF)

A print termination character may be added to the characters printed during a print screen function. Set mode (SM) causes a form feed character (FF) to be added to the characters printed. Reset mode does not add a print termination character. The print screen function always sends carriage return (CR) and line feed (LF) characters to the printer.

To set:	ESC [18 h	(form feed)
To reset:	ESC [18 l	(none)

7.4.6 Printer Status Report (DECPSR)

The control sequences used by the host computer to determine printer status in response to a printer status report (DECPSR) request are shown below. Before any print mode or function is selected, printer status must be determined.

	Sequence	Description
Requests	ESC [? 15 n	Printer status requested
	ESC [? 13 n	No response (printer DTR was off during terminal power-up). Be sure the printer is on before turning the terminal on.
Responses		
	ESC [? 10 n	Ready response
	ESC [? 11 n	Not ready response (printer DTR went off after terminal power-up).

SECTION 8. BOOTLOADING

8.1 GENERAL

This section contains instruction and information neccessary to bootload code into an ATL-008 terminal. It is intended for users who are familiar with the terminal and the C programming language.

The APC sequnece is used to inform the terminal that subsequent code is to be loaded into RAM. The sequence has the following structure:

APC B <data> ST

APC is the ESC _ sequence (for additional information, see the APC sequence and the section on changing softkeys). B indicates bootloading. <data> consists of the data to be bootloaded and must consist of executable code for 68000-family microprocessors. This data must be in the form of Motorola (TM) S-Records. ST is the ANSI string terminator, ESC \ .

Bootloadable code must start at the terminal's execution address. This address is 84000H. With the exception of some pointers, other addresses given in this section are relative addresses based on this absolute.

8.2 S-RECORDS

8.2.1 Description

S-records are character strings made of five fields which identify the record type, record length, memory address, code or data, and checksum. Each byte of binary data is represented as a 2-character hex number. The first hex digit represents the high-order 4 bits of the byte; the second digit represents the low-order 4 bits of the byte.

TM Motorola is a trademark of Motorola.

The S-record fields are:

type record length address code/data checksum

where the fields are composed as follows:

FIELD	CHARACTERS	CONTENTS
type	2	S-record type--S0, S1, etc.
record length	2	The count of the character pairs in the record, excluding the type and record length.
address	4, 6, or 8	The 2-, 3-, or 4-byte address where the data is to be loaded into memory.
code/data	0-2n	From 0 to n bytes of executable code, memory-loadable data, or descriptive information.
checksum	2	The least significant byte of the one's complement is the sum of the values represented by the pairs of characters making up the record length, address, and the code/data fields.

Each record may be terminated with a CR/LF/NULL. Additionally, an S-record may have an initial field to accommodate other data such as line numbers generated by some time-sharing systems.

Accuracy of transmission is ensured by the record length (byte count) and checksum fields.

An S-record module may contain S-records of the following eight types:

S0 The header record for each block of S-records. The code/data field may contain any descriptive information identifying the following block of S-records. The ATL-008 will ignore this type, but it is acceptable.

S1 A record containing code/data and the 2-byte address at which the code/data is to reside.

S2 A record containing code/data and the 3-byte address at which the code/data is to reside.

S3 A record containing code/data and the 4-byte address at which the code/data is to reside.

S5 A record containing the number of S1, S2, and S3 records transmitted in a particular block. This count appears in the address field. There is no code/data field.

S7 A termination record for a block of S3 records. The address field may optionally contain the 4-byte address of the instruction to which control is to be passed. There is no code/data field.

S8 A termination record for a block of S2 records. The address field may optionally contain the 3-byte address of the instruction to which control is to be passed. There is no code/data field.

S9 A termination record for a block of S1 records. The address field may optionally contain the 2-byte address of the instruction to which control is to be passed. Under VERSAdos, the resident linker's ENTRY command can be used to specify this address. If not specified, the first entry point specification encountered in the object module input will be used. There is no code/data field.

Only one termination record is used for each block of S-records. S7 and S8 records are usually used only when control is to be passed to a 3- or 4-byte address after the bootload operation is complete. Normally, only one header record is used, although it is possible for multiple header records to occur.

NOTE: The ATL-008 recognizes only three of these type of records -- S0 (ignored, but acceptable), S2 (3-byte address), and S8 (termination block for S2 records). The other records are included in this discussion for illustrative purposes only.

8.2.2 Generating S-Records

S-record programs may be produced by several dump utilities, debuggers, linkage editors, or several cross assemblers or cross linkers. Several developement systems also provide S-record output capabilities. Several programs are available for downloading a file in S-record format from a host system to an 8-bit micropocessor-based or a 16-bit microprocessor-based system.

EXAMPLE

The following is a typical S-record-format module, as printed or displayed:

```
S0066000048444521B
S1130000285F245F2212226A00042429000823 7C2A
S1130010000200080008262900185381234100 1813
S113002041E900084E42234300182342000824A952
S107003000144ED492
S9030000FC
```

The module consists of one S0 record, four S1 records, and an S9 record.

The S0 record contains the following character pairs:

S0 S-record type S0, indicating that it is a header record.

06 Hexadecimal 06 (the record length) indicates that six character pairs (or ASCII bytes) follow.

00 Four-character 2-byte address field, zeroes in this example.
00

48
44 ASCII H, D, and R - "HDR".
52

1B The checksum.

The first S1 record contains the following character pairs:

S1 S-record type S1, indicating that it is a code/data record to be loaded/verified at a 2-byte address.

13 Hexadecimal 13 (decimal 19), indicating that 19 character pairs, representing 19 bytes of binary data, follow.

00 Four-character 2-byte address field; hexadecimal address 0000, where
00 the data which follows is to be loaded.

The next 16 character pairs of the first S1 contain the actual program code listed as ASCII pairs. In this assembly language example, the hex opcodes of the program are written in sequence in the code/data fields of the S1 records:

OPCODE	INSTRUCTION	
285F	MOVE.L	(A7)+,A4
245F	MOVE.L	(A7)+,A2
2212	MOVE.L	(A2),D1
226A0004	MOVE.L	$(A2),A1
237C	MOVE.L	#FORCEFUNC,FUNCTION(A1)

. (The balance of this code is continued in the code/data fields of
. remaining S1 records, and stored in memory location 0010, etc.)

2A The checksum of the first S1 record.

The second and third S1 records each also contain $13 (19) character pairs and are ended with checksums 13 and 52, respectively. The fourth S1 record contains 07 character pairs and has a checksum of 92.

The S9 record is explained as follows:

S9 S-record type S9, indicating that it is a termination record.

03 Hexadecimal 03, indicating that three character pairs (3 bytes) follow.

00 The address field, zeroes.
00

FC The checksum of the S9 record.

Each printable character in an S-record is encoded in hexadecimal (ASCII in this example) representation of the binary bits which are actually transmitted.

8.3 USING THE ATL-008 PRIMITIVES

A variety of primitive operations are available on the ATL-008. This subsection deals with the terminal algorithm access points, jump table pointers, and describes the available primitives.

8.3.1 ATL-008 Access Points

The ATL-008 provides three pointers to allow access to standard entry points. These pointers reside at the same addresses regardless of the firmware

version. Pointers, like integers, are 32 bits wide. The pointers, addresses, and their corresponding C defines are:

Pointer	Address
pointer to interrupt vector table	= 102H
pointer to table of routines	= 106H
pointer to parameter array	= 10AH
typedef int (*PFUNC)();	/* pointer to a function */
typedef PFUNC *PPF;	/* pointer to a ptr to a function */
typedef PPF *PPPF;	/* pointer to a ptr to a ptr to a function */
typedef int *PI;	/* pointer to an integer */
typedef PI *PPI;	/* pointer to a ptr to an integer */
	/* align pointer types */
# define RMPVT (PPPF)0x102	/* address of pointer to vector table */
# define RMPFT (PPPF)0x106	/* address of pointer to function table*/
# define RMPPA (PPI) 0x10a	/* address of pointer to int parameter array */

The vector table and function table used with these pointers are detailed in the following sections. The pointer to the parameter array points to a common RAM area used for passing parameters between routines.

Example

This example shows how to use the above pointers and pass parameters.

```
;*****************************************************************************
;       these are the user's local C variables and the basic method
;       for calling the C interface routines.
;*****************************************************************************


int     myCroutine();                   /* user's C routine */

PPPF    Ivect = RMPVT;                    /* local variable which points to */
                                          /* the ptr to table of interrupt */
                                          /* vectors */

PPPF    Cfunc = RMPFT;                    /* local variable which points to */
                                          /* the ptr to table of pointers to */
                                          /* functions  */

PPI     Cparm = RMPPA;                    /* local variable which points to */
                                          /* the ptr to an integer array used */
                                          /* to pass parameters */
```

```
(* Cparm)[0] = 0;                          /* pass parameters like this */
(* Cparm)[1] = 1;


( (* Cfunc) [ BLAWI ])();                  /* call a C interface routine */

x = (* Cparm)[0];                          /* return values like this */
y = (* Cparm)[1];

( (* Ivect) [2]) = myCroutine;            /* set aux port interrupt to */
                                           /* call a C routine */
```

8.3.2 Interrupt Vector Table

There are several ram areas that are used to provide access to the interrupt routines. The interrupt vector pointer points to these locations as listed in Table 8-1.

Table 8-1
Bootload Vector Table

Vector	Number (index)
Timer Interrupt	0
Keyboard	1
Video Interrupt	2
Aux Transmit Interrupt	3
Aux External Status Change	4
Aux Receive Interrupt	5
Aux Special Receive Interrupt	6
Main Transmit Interrupt	7
Main External Status Change	8
Main Receive Interrupt	9
Main Special Receive Interrupt	10
Keybd and I/O Received Char	11
Main Port Received Char	12
Aux Port Received Char	13
Main Port Output	14
Aux Port Output	15
Control Jump Table	16
Control Function Jump Table	17
Control Sequence without Intermediates JT	18
Control Sequence with Intermediates JT	19
User-defined Test Vectors	20-29

Interrupt Vectors

The first 15 entries in Table 8-1 are used as vectors. When an interrupt is received, execution will continue at the long word address contained in that interrupt's vector location. In effect, the C routine whose address is in the vector location will be called by the interrupt. If the downloaded code is written in AS68, then the address of the routine is still loaded in the vector. The routine should include an RTS at the end.

After power-up, all vector locations (1st 15 long words) are initialized to the standard ATL-008 algorithm.

Interrupt Vector Examples

These examples illustrate the use of the first 15 vectors in Table 8-1. Each starts with a title block and a vector number. These numbers correspond to the numbers in the table.

```
;*************************************************************************
;      interrupt routine interfaces
;*************************************************************************
```

VECTOR[n]

The user must load the address of the new interrupt routine into this location, where 'n' is the number of the interrupt to be changed. The routine should end with a standard return/rts statement. The calling routine will take care of the needed rte statement.

```
( (* Ivect) [1]) = myCroutine;        /* set keyboard interrupt to */
                                       /* call a C routine */
```

```
;*************************************************************************
;       keybd, main port and aux port received character
;       this routine receives control when a character has been received
;       from the respective source, but has not yet been interpreted.
;*************************************************************************
```

VECTOR[11-13]

The user must download the address of the routine to be executed into this location. The routine is called with the following parameters.

```
int char,src;
```

```
( (* Ivect) [11]) = myCroutine;        /* set received char to */
                                        /* call a C routine */

(* Cparm)[0] = [00,0char];        received character
```

The routine should end with a standard return/rts statement.

```
;*****************************************************************************
;       main i/o output
;       this routine receives control just prior to sending the character
;       out the main port. the character to be sent is passed to the routine.
;*****************************************************************************
```

VECTOR[14]

The user must download the address of the routine to be executed into this location. The routine is called with the following parameters:

```
int char;

( (* Ivect) [14]) = myCroutine;        /* set output char to */
                                        /* call a C routine */

/*      calling parameters      */
(* Cparm)[0] = [00,0char];      received character
```

char is character to be sent to main port. The routine must now assume all responsiblity for the character. The routine should end with a standard return/rts statement.

```
;*****************************************************************************
;       aux i/o output
;       this routine receives control just prior to sending the character
;       out the aux port. the character to be sent is passed to the routine
;*****************************************************************************
```

VECTOR[15]

The user must download the address of the routine to be executed into this location. The routine is called with the following parameters.

```
int char;

( (* Ivect) [15]) = myCroutine;        /* set output char to */
                                        /* call a C routine */

/*      calling parameters      */
(* Cparm)[0] = [00,0char];      received character
```

char is character to be sent to aux port. The routine must now assume all responsiblity for the character. The routine should end with a standard return/rts statement. The calling routine will take care of the needed rte statement.

Jump Table Pointers

Vectors 16 through 19 contain a pointer to the jump table used to interpret that level of ANSI X3.64 command. Each location is initialized at power-up to point to the ATL-008 algorithm jump tables. If the pointer is changed, the new jump table will be used instead.

ANSI Sequnce Type	Jump Table
control <char>	control jump table
esc <char>	control function jump table
esc [<char>	control sequence without intermediates jump
esc [space <char>	control sequence with intermediates jump table

User-defined Test Vectors

The user-defined test vectors determine the address of the routine to execute when the field type is set to a user-defined test. The vector routine is executed when an attempt is made to exit the field. The vectors are initialized to an error routine.

```
;*************************************************************************
;       user defined test vectors
;       these vectors contain pointers to the users downloaded code. when
;       a field that has been defined to use a user defined test is exited,
;       the address of the routine is loaded from this table.
;*************************************************************************
VECTOR[20]-VECTOR[29]
```

Vectors are initialized to the standard error routine.

on exit:

```
(* Cparm)[0] = char;     char that was entered in field.
(* Cparm)[1] = non0;     an error has been found, the terminal algorithm will
                         execute the field error exit routine.
(* Cparm)[1] = 0;        no error, no action is taken on exit.
```

8.3.3 Table of Routines

Several routines exist which may be used used for bootloading operations. Table 8-2 lists the C declarations which access the ATL-008 algorithm. These functions are detailed under the heading Available Primitives.

Table 8-2
Routine Definitions

```
# define   BLRNV  0     /* read EEPROM byte                          */
# define   BLCHP  1     /* set bytes out main or aux port            */
# define   BLGETB 2     /* get a dynamic buffer                      */
# define   BLRELB 3     /* release a dynamic buffer                  */
# define   BLICH  4     /* intercept byte from key or io             */
# define   BLCSP  5     /* char string at screen position            */
# define   BLCMDI 6     /* send byte to command interpreter */
# define   BLCUR  7     /* get cursor position                       */
# define   BLFAT  8     /* get field attr. at active pos             */
# define   BLEXC  9     /* examine char at active pos                */
# define   BLBAC  10    /* binary to ascii conversion                */
# define   BLABC  11    /* ascii to binary conversion                */
# define   BLGTD  12    /* get elapsed run time                      */
```

8.3.4 Available Primitives

This subsection illustrates the use of available primitives in the ATL-008.

Read EEPROM Byte

The following primitive reads a specific byte from EEPROM memory.

```
;*****************************************************************************
;                   read EEPROM byte
;*****************************************************************************

int offset;

/*      calling parameters      */
(* Cparm)[0] = offset;          offset into EEPROM where byte is to be read


        ( (* Cfunc) [ BLRNV ])();


/*      returned parameters     */
x[0] = [00,0byte]               low order byte from EEPROM
```

Send_a_Character_String

```
;*************************************************************************
;       send a character string out the main or aux port.
;       the characters are placed at the end of the output queue for
;       the specifed port. the routine will return after all chars
;       have been placed on the output queue. if the queue is full,
;       the routine will wait until space on the queue is available.
;*************************************************************************

int port;
char str[];

/*      calling parameters      */
(* Cparm)[0] = port;            port = 1 : main port
                                     = 2 : aux port
(* Cparm)[1] = str;             str is a pointer to a null terminated char
                                string to be sent to the indicated i/o port.

        ( (* Cfunc) [ BLCHP ])();


/*      returned parameters     */
<none>
```

Get_a_Dynamic_Buffer

```
;*************************************************************************
;       get a dynamically allocatable buffer
;       if ptr = 0 on return, no buffers are availiable
;*************************************************************************


/*      calling parameters      */
<none>


        ( (* Cfunc) [ BLGETB ])();


/*      returned parameters     */
(* Cparm)[0] = ptr to start of buffer;
```

Release a Dynamic Buffer

```
;*****************************************************************************
;       release a dynamically allocatable buffer
;*****************************************************************************

int *ptr;

/*      calling parameters      */
(* Cparm)[0] = ptr;             ptr points to the START of the buffer to be
                                returned

        ( (* Cfunc) [ BLRELB ])();


/*      returned parameters     */
<none>
```

Intercept a Character

```
;*****************************************************************************
;       intercept character from either the keyboard or i/o.
;       all i/o processing is done on the character. the character is
;       intercepted just before being passed to 'intchr'.
;*****************************************************************************


int src;

/*      calling parameters      */
(* Cparm)[0] = src;             src = 0 : keyboard
                                    = 1 : main port
                                    = 2 : aux port

        ( (* Cfunc) [ BLICH ])();


/*      returned parameters     */
(* Cparm)[0] = [00,0char];      high order word = 0,0
                                low  order word = 0,char
```

Place a Character on Screen

```
;*****************************************************************************
;       write a character string at the specified screen position without
;       changing the current position pointer
;
;*****************************************************************************
```

```
int row,col;
char str[];

/*      calling parameters      */
(* Cparm)[0] = row;              row,col is the position in the logical page
(* Cparm)[1] = col;              where the string is to be written
(* Cparm)[2] = str;              str is a pointer to a null terminated char
                                        string
(* Cparm)[3] = attribute        attribute to be written with char
(* Cparm)[4] = page#             logical page number

        ( (* Cfunc) [ BLCSP ])();


/*      returned parameters     */
<none>
```

Send_a_Byte_to_the_Command_Interpreter

```
;*****************************************************************************
;       send a command sequence or data char to the command interpreter
;       the character is sent to 'intchr', and from there diverted to
;       the correct jump table.
;       src = bootload
;*****************************************************************************

char cmd[];

/*      calling parameters      */
(* Cparm)[0] = cmd;              cmd is a pointer to a null terminated char
                                string to be sent to the command interpreter.
(* Cparm)[1] = src;              src to be used for char string (key,main,aux)

        ( (* Cfunc) [ BLCMDI ])();


/*      returned parameters     */
<none>
```

Check_Cursor_Position

```
;*****************************************************************************
;       interrogate the cursor position
;*****************************************************************************

int src;

/*      calling parameters      */
(* Cparm)[0] = src;              src = 0 : keyboard
                                    = 1 : main port
                                    = 2 : aux port
```

```
        ( (* Cfunc) [ BLCUR ])();


/*      returned parameters     */
(* Cparm)[0] = row;
(* Cparm)[1] = col;
(* Cparm)[2] = logical page # containing the src's active position (1-12)
```

Check_Attributes

```
;*****************************************************************************
;       interrogate the field attributes which are in effect for the
;       current position.
;*****************************************************************************

int src;

/*      calling parameters      */
(* Cparm)[0] = src;             src = 0 : keyboard
                                    = 1 : main port
                                    = 2 : aux port

        ( (* Cfunc) [ BLFAT ])();


/*      returned parameters     */
(* Cparm)[0] = start row of qualifed area if any
(* Cparm)[1] = start col        "
(* Cparm)[2] = end row          "
(* Cparm)[3] = end col          "
(* Cparm)[4] = flags

;
; bits within flag word...
;
spuser equ      15              ; special user defined qa type
modfid equ      14              ; this qa has been modified flag
modify equ      13              ; modify
autotx equ      12              ; auto-transmit
zfill  equ      11              ; zero fill (0 = space fill)
fill   equ      10              ; 1 = space or zero fill
rtjust equ      9               ; 1 = right justify (0 = left justify)
justfy equ      8               ; 1 = right or left justify
totfil equ      7               ; total fill
mstent equ      6               ; must enter
uprcas equ      5               ; upper-case
hortab equ      4               ; horizontal tab
qatype equ      0-3             ; qa type (bits 0,1,2,3)
;
```

```
; types of qa's...(bits 0,1,2,3)
;
progrd equ     0               ; protected and guarded (no entry, no xmit)
prougd equ     1               ; protected and unguarded (no entry, xmit)
unprot equ     2               ; accept all input
graqa  equ     3               ; accept graphics (all displayable)
alpha  equ     4               ; accept alphabetics
numsym equ     5               ; accept numerics (0-9 , . + -)
numbrs equ     6               ; numbers only (0-9)
```

NOTE: If the flags indicate, unprot then the current position is not part of a qualified area.

Check_Character

```
;*****************************************************************************
;       examine the char at the specified position without changing
;       the active position.
;*****************************************************************************

int row,col;

/*      calling parameters      */
(* Cparm)[0] = row;             row, col is the position in the logical page
(* Cparm)[1] = col;             which is examined
(* Cparm)[2] = src;             which logical page: key,main,aux


        ( (* Cfunc) [ BLEXC ])();


/*      returned parameters     */
x[0] = [00,attr,char]           high order word = 0,
                                lower order word = attribute,char
                                attribute is encoded as follows:
                                        $01     : half intensity bit
                                        $02     : blank video bit  security
                                        $04     : blinking video bit
                                        $08     : reverse video bit
                                        $10     : underline video bit
```

Binary_to_ASCII_Conversion

```
;*****************************************************************************
;       binary to ascii conversion routine.
;*****************************************************************************

int number;

/*      calling parameters      */
(* Cparm)[0] = number;          number = a binary number to be converted


        ( (* Cfunc) [ BLBAC ])();


/*      returned parameters     */
(* Cparm)[0] = 1st 4 digits in ascii
(* Cparm)[1] = 2nd 4 digits in ascii
(* Cparm)[2] = 0

        (* Cparm) is a char array (1 byte chars) containing an 8 byte
        null terminated character string. wow.
```

ASCII_to_Binary_Conversion

```
;*****************************************************************************
;       ascii to binary conversion routine.
;*****************************************************************************

char str[];

/*      calling parameters      */
(* Cparm)[0] = str;             str is a pointer to a null terminated ascii
                                string allowable digits: [-][0123456789]

        ( (* Cfunc) [ BLABC ])();


/*      returned parameters     */
(* Cparm)[0] = binary number;
```

Check_Time

```
;*****************************************************************************
;       get the elapsed run time.
;*****************************************************************************

int time;

/*      calling parameters      */
<none>


        ( (* Cfunc) [ BLGTD ])();


/*      returned parameters     */
(* Cparm)[0] = time;            time is number of 5 ms clock interrupts
                                since power-on
```

APPENDIX A. ASCII CHART

ASCII CHARACTER CHART

DEC	HEX	OCT	BINARY		CHR
000	00	000	00000000	NUL	*C/A @
001	01	001	00000001	SOH	C/A A
002	02	002	00000010	STX	C/A B
003	03	003	00000011	ETX	C/A C
004	04	004	00000100	EOT	C/A D
005	05	005	00000101	ENQ	C/A E
006	06	006	00000110	ACK	C/A F
007	07	007	00000111	BEL	C/A G
008	08	010	00001000	BS	C/A H
009	09	011	00001001	HT	C/A I
010	0A	012	00001010	LF	C/A J
011	0B	013	00001011	VT	C/A K
012	0C	014	00001100	FF	C/A L
013	0D	015	00001101	CR	C/A M
014	0E	016	00001110	SO	C/A N
015	0F	017	00001111	SI	C/A O
016	10	020	00010000	DLE	C/A P
017	11	021	00010001	DC1	C/A Q
018	12	022	00010010	DC2	C/A R
019	13	023	00010011	DC3	C/A S
020	14	024	00010100	DC4	C/A T
021	15	025	00010101	NAK	C/A U
022	16	026	00010110	SYN	C/A V
023	17	027	00010111	ETB	C/A W
024	18	030	00011000	CAN	C/A X
025	19	031	00011001	EM	C/A Y
026	1A	032	00011010	SUB	C/A Z
027	1B	033	00011011	ESC	C/A [
028	1C	034	00011100	FS	C/A \
029	1D	035	00011101	GS	C/A]
030	1E	036	00011110	RS	C/A ^
031	1F	037	00011111	US	C/A _

DEC	HEX	OCT	BINARY	CHR
032	20	040	00100000	SP
033	21	041	00100001	!
034	22	042	00100010	"
035	23	043	00100011	#
036	24	044	00100100	$
037	25	045	00100101	%
038	26	046	00100110	&
039	27	047	00100111	'
040	28	050	00101000	(
049	29	051	00101001	)
042	2A	052	00101010	*
043	2B	053	00101011	+
044	2C	054	00101100	,
045	2D	055	00101101	-
046	2E	056	00101110	.
047	2F	057	00101111	/
048	30	060	00110000	0
049	31	061	00110001	1
050	32	062	00110010	2
051	33	063	00110011	3
052	34	064	00110100	4
053	35	065	00110101	5
054	36	066	00110110	6
055	37	067	00110111	7
056	38	070	00111000	8
057	39	071	00111001	9
058	3A	072	00111010	:
059	3B	073	00111011	;
060	3C	074	00111100	<
061	3D	075	00111101	=
062	3E	076	00111110	>
063	3F	077	00111111	?

DEC	HEX	OCT	BINARY	CHR
064	40	100	01000000	@
065	41	101	01000001	A
066	42	102	01000010	B
067	43	103	01000011	C
068	44	104	01000100	D
069	45	105	01000101	E
070	46	106	01000110	F
071	47	107	01000111	G
072	48	110	01001000	H
073	49	111	01001001	I
074	4A	112	01001010	J
075	4B	113	01001011	K
076	4C	114	01001100	L
077	4D	115	01001101	M
078	4E	116	01001110	N
079	4F	117	01001111	O
080	50	120	01010000	P
081	51	121	01010001	Q
082	52	122	01010010	R
083	53	123	01010011	S
084	54	124	01010100	T
085	55	125	01010101	U
086	56	126	01010110	V
087	57	127	01010111	W
088	58	130	01011000	X
089	59	131	01011001	Y
090	5A	132	01011010	Z
091	5B	133	01011011	[
092	5C	134	01011100	\
093	5D	135	01011101	]
094	5E	136	01011110	^
095	5F	137	01011111	_

DEC	HEX	OCT	BINARY	CHR
096	60	140	01100000	`
097	61	141	01100001	a
098	62	142	01100010	b
099	63	143	01100011	c
100	64	144	01100100	d
101	65	145	01100101	e
102	66	146	01100110	f
103	67	147	01100111	g
104	68	150	01101000	h
105	69	151	01101001	i
106	6A	152	01101010	j
107	6B	153	01101011	k
108	6C	154	01101100	l
109	6D	155	01101101	m
110	6E	156	01101110	n
111	6F	157	01101111	o
112	70	160	01110000	p
113	71	161	01110001	q
114	72	162	01110010	r
115	73	163	01110011	s
116	74	164	01110100	t
117	75	165	01110101	u
118	76	166	01110110	v
119	77	167	01110111	w
120	78	170	01111000	x
121	79	171	01111001	y
122	7A	172	01111010	z
123	7B	173	01111011	{
124	7C	174	01111100	|
125	7D	175	01111101	}
126	7E	176	01111110	~
127	7F	177	01111111	DEL

*C/A = CONTROL/ALT

EXPLANATION OF CONTROL CODES

NUL = Null
SOH = Start of Heading
STX = Start of Text
ETX = End of Text
EOT = End of Transmission
ENQ = Enquiry
ACK = Acknowledge
BEL = Bell
BS = Backspace

HT = Horizontal Tab
LF = Line Feed
VT = Vertical Tab
FF = Form Feed
CR = Carriage Return
SO = Shift Out
SI = Shift In
DLE = Data Link Escape
DC1 = Device Control 1

DC2 = Device Control 2
DC3 = Device Control 3
DC4 = Device Control 4
NAK = Negative Acknowledge
SYN = Synchronous Idle
ETB = End of Transmission Block
CAN = Cancel
EM = End of Medium
SUB = Substitute

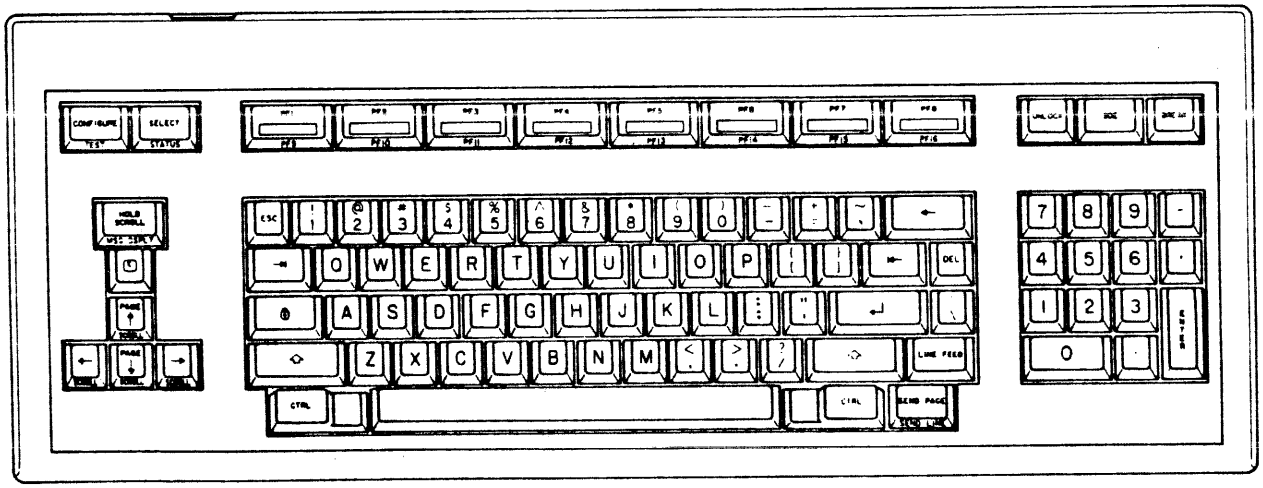
ESC = Escape
FS = File Separator
GS = Group Separator
RS = Record Separator
US = Unit Separator
SP = Space
DEL = Delete

APPENDIX B.

KEYBOARD LAYOUTS AND CODE CHARTS

BINARY VALUE b4 b3 b2 b1	HEX VALUE	CONTROL CHARACTERS b7=0 b6=0		CONTROL CHARACTERS b7=0 b6=1		DISPLAYABLE CHARACTERS b7=1 b6=0		DISPLAYABLE CHARACTERS b7=1 b6=1			
		0		1		2	3	4	5	6	7
0000	0	NUL	@	DLE	P	SP	0	@	P	`	p
0001	1	SOH	A	DC1 X-ON	Q	!	1	A	Q	a	q
0010	2	STX	B	DC2	R	"	2	B	R	b	r
0011	3	ETX	C	DC3 X-OFF	S	#	3	C	S	c	s
0100	4	EOT	D	DC4	T	$	4	D	T	d	t
0101	5	ENQ	E	NAK	U	%	5	E	U	e	u
0110	6	ACK	F	SYN	V	&	6	F	V	f	v
0111	7	BEL	G	ETB XMIT	W	'	7	G	W	g	w
1000	8	BS	H	CAN	X	(	8	H	X	h	x
1001	9	HT	I	EM	Y	)	9	I	Y	i	y
1010	A	LF	J	SUB	Z	*	:	J	Z	j	z
1011	B	VT	K	ESC	┘	+	;	K	[	k	{
1100	C	FF	L	FS	\	,	<	L	\	l	¦
1101	D	CR	M	GS	]	-	=	M	]	m	}
1110	E	SO ALT CHAR SET ON	N	RS	^	.	>	N	^	n	~
1111	F	SI ALT CHAR SET OFF	O	US	_	/	?	O	_	o	DEL

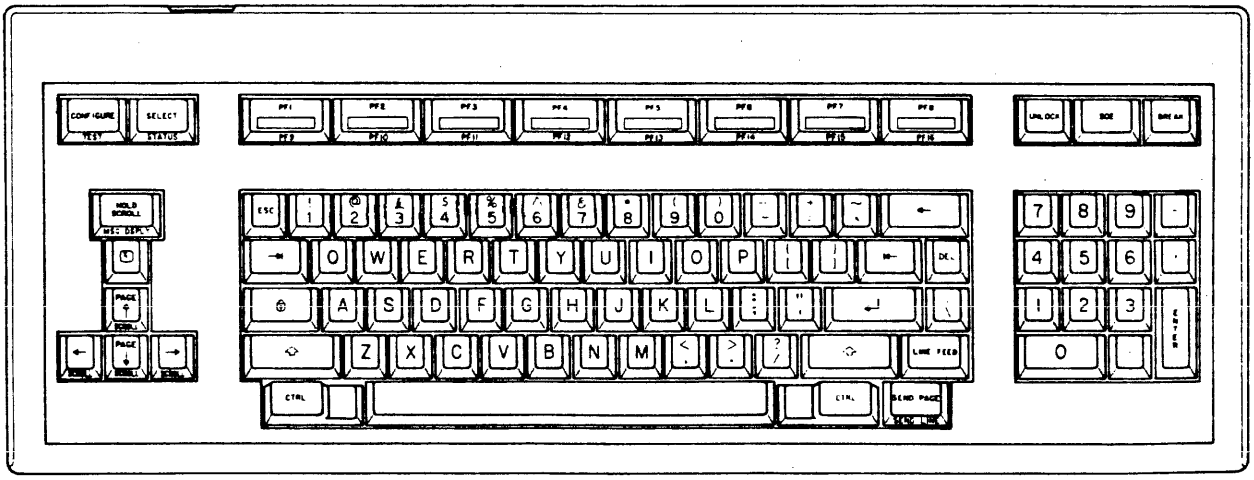
ATL-004 USASCII Code Chart



USASCII Keyboard Layout

	CONTROL CHARACTERS		DISPLAYABLE CHARACTERS					
BIT 7 6 / 4 3 2 1 0	0_0	0_1	0_{10}	0_{11}	1_{00}	1_{01}	1_{10}	1_{11}
0000	NUL @	DLE P	SP	0	@	P	`	p
0001	SOH A	DC1 X-ON Q	!	1	A	Q	a	q
0010	STX B	DC2 R	"	2	B	R	b	r
0011	ETX C	DC3 X-OFF S	£	3	C	S	c	s
0100	EOT D	DC4 T	$	4	D	T	d	t
0101	ENQ E	NAK U	%	5	E	U	e	u
0110	ACK F	SYN V	&	6	F	V	f	v
0111	BEL G	ETB XMIT W	'	7	G	W	g	w
1000	BS H	CAN X	(	8	H	X	h	x
1001	HT I	EM Y	)	9	I	Y	i	y
1010	LF J	SUB Z	*	:	J	Z	j	z
1011	VT K	ESC [	+	;	K	[	k	|
1100	FF L	FS \	,	<	L	\	l	¦
1101	CR M	GS]	-	=	M	]	m	}
1110	SO N	RS ^	.	>	N	^	n	~
1111	SI O	US _	/	?	O	_	o	DEL

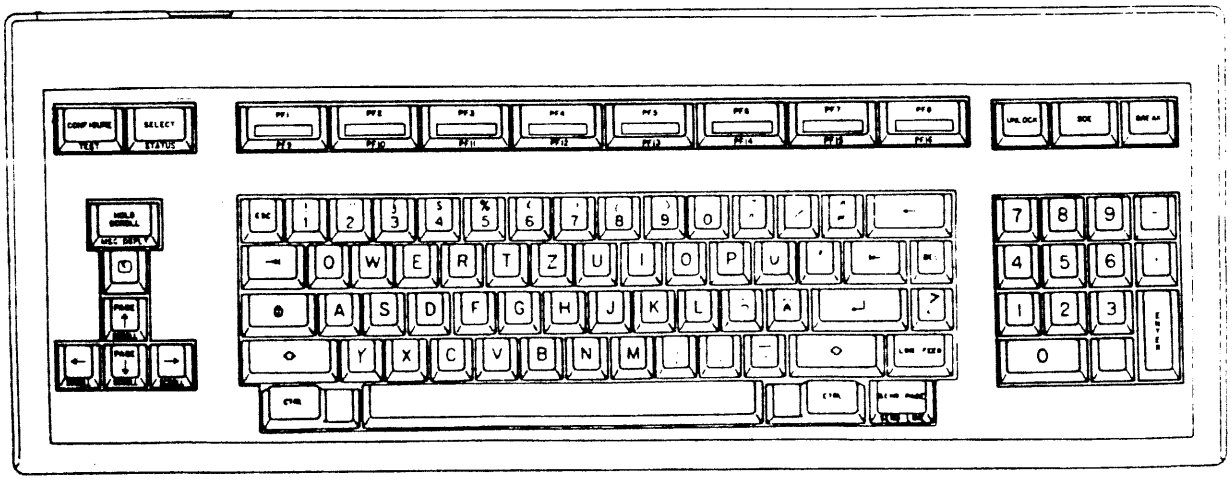
United Kingdom Code Chart



United Kingdom Keyboard Layout

BIT 7 6 / 4 3 2 1	CONTROL CHARACTERS		DISPLAYABLE CHARACTERS					
	0_0	0_1	0_1	0_1	1_0	1_0	1_1	1_1
0000	NUL @	DLE P	SP	0	@	P	`	p
0001	SOH A	DC1 X-ON Q	!	1	A	Q	a	q
0010	STX B	DC2 R	"	2	B	R	b	r
0011	ETX C	DC3 X-OFF S	#	3	C	S	c	s
0100	EOT D	DC4 T	$	4	D	T	d	t
0101	ENQ E	NAK U	%	5	E	U	e	u
0110	ACK F	SYN V	&	6	F	V	f	v
0111	BEL G	ETB XMIT W	'	7	G	W	g	w
1000	BS H	CAN X	(	8	H	X	h	x
1001	HT I	EM Y	)	9	I	Y	i	y
1010	LF J	SUB Z	*	:	J	Z	j	z
1011	VT K	ESC Ä	+	;	K	Ä	k	ä
1100	FF L	FS Ö	,	<	L	Ö	l	ö
1101	CR M	GS Ü	-	=	M	Ü	m	ü
1110	SO N	RS ^	.	>	N	^	n	ß
1111	SI O	US _	/	?	O	_	o	DEL

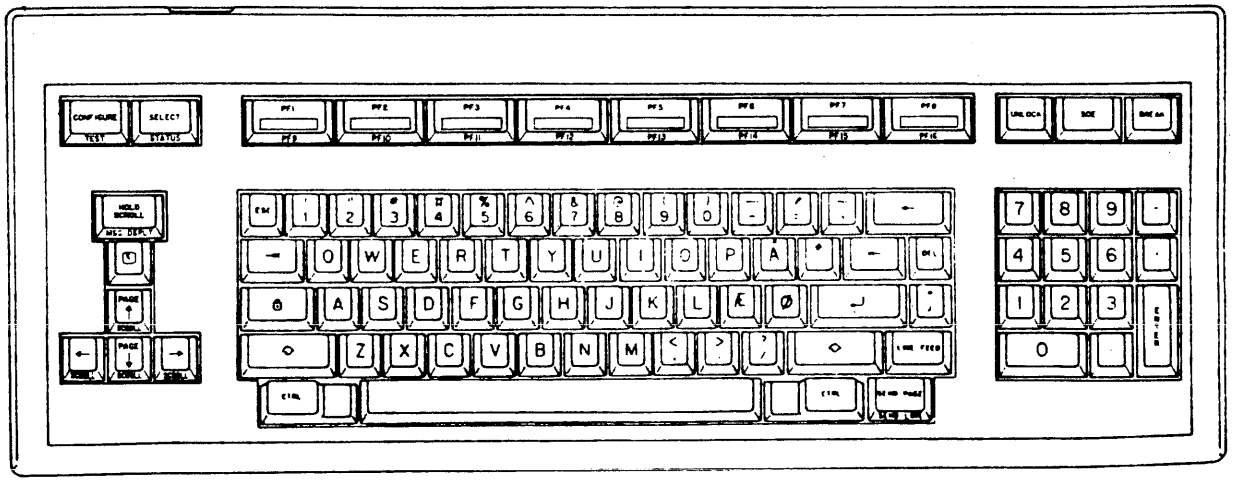
German Code Chart



German Keyboard Layout

	CONTROL CHARACTERS		DISPLAYABLE CHARACTERS					
BIT 7 6 4 3 2 1	0_0	0_1	0_{1_0}	0_{1_1}	1_{0_0}	1_{0_1}	1_{1_0}	1_{1_1}
0000	NUL @	DLE P	SP	0	@	P	`	p
0001	SOH A	DC1 X-ON Q	!	1	A	Q	a	q
0010	STX B	DC2 R	"	2	B	R	b	r
0011	ETX C	DC3 X-OFF S	#	3	C	S	c	s
0100	EOT D	DC4 T	¤	4	D	T	d	t
0101	ENQ E	NAK U	%	5	E	U	e	u
0110	ACK F	SYN V	&	6	F	V	f	v
0111	BEL G	ETB XMIT W	'	7	G	W	g	w
1000	BS H	CAN X	(	8	H	X	h	x
1001	HT I	EM Y	)	9	I	Y	i	y
1010	LF J	SUB Z	*	:	J	Z	j	z
1011	VT K	ESC Æ	+	;	K	Æ	k	æ
1100	FF L	FS Ø	,	<	L	Ø	l	ø
1101	CR M	GS Å	-	=	M	Å	m	å
1110	SO N	RS ^	.	>	N	^	n	~
1111	SI O	US -	/	?	O	_	o	DEL

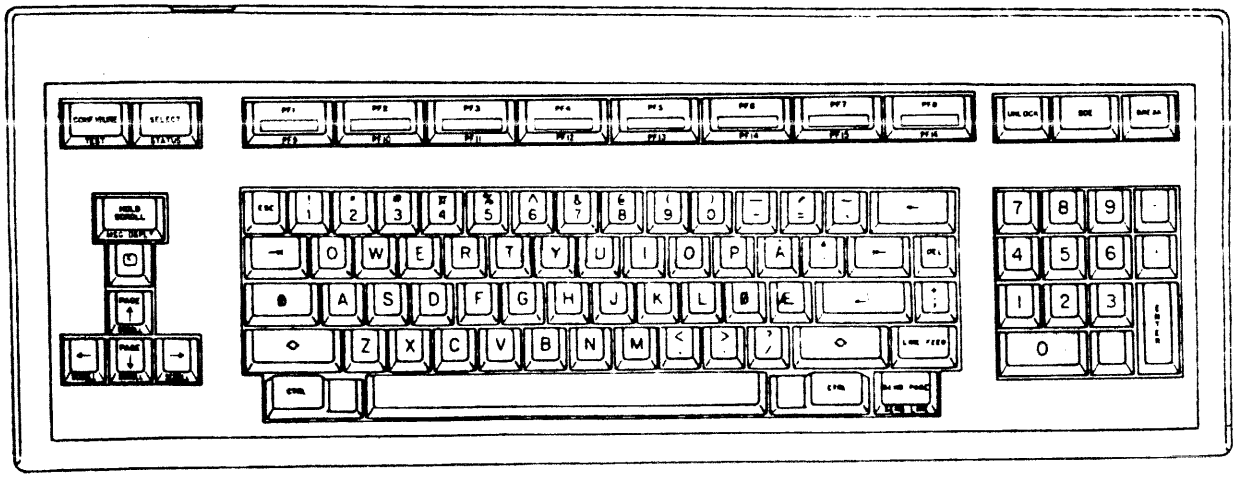
Danish Code Chart



Danish Keyboard Layout

BIT 7 6 / 4 3 2 1	CONTROL CHARACTERS 0_0	0_1	DISPLAYABLE CHARACTERS $0_1{}_0$	$0_1{}_1$	$1_0{}_0$	$1_0{}_1$	$1_1{}_0$	$1_1{}_1$
0000	NUL @	DLE P	SP	0	@	P	`	p
0001	SOH A	DC1 X-ON Q	!	1	A	Q	a	q
0010	STX B	DC2 R	"	2	B	R	b	r
0011	ETX C	DC3 X-OFF S	#	3	C	S	c	s
0100	EOT D	DC4 T	¤	4	D	T	d	t
0101	ENQ E	NAK U	%	5	E	U	e	u
0110	ACK F	SYN V	&	6	F	V	f	v
0111	BEL G	ETB XMIT W	'	7	G	W	g	w
1000	BS H	CAN X	(	8	H	X	h	x
1001	HT I	EM Y	)	9	I	Y	i	y
1010	LF J	SUB Z	*	:	J	Z	j	z
1011	VT K	ESC Æ	+	;	K	Æ	k	æ
1100	FF L	FS Ø	,	<	L	Ø	l	ø
1101	CR M	GS Å	-	=	M	Å	m	å
1110	SO N	RS ^	.	>	N	^	n	~
1111	SI O	US _	/	?	O	_	o	DEL

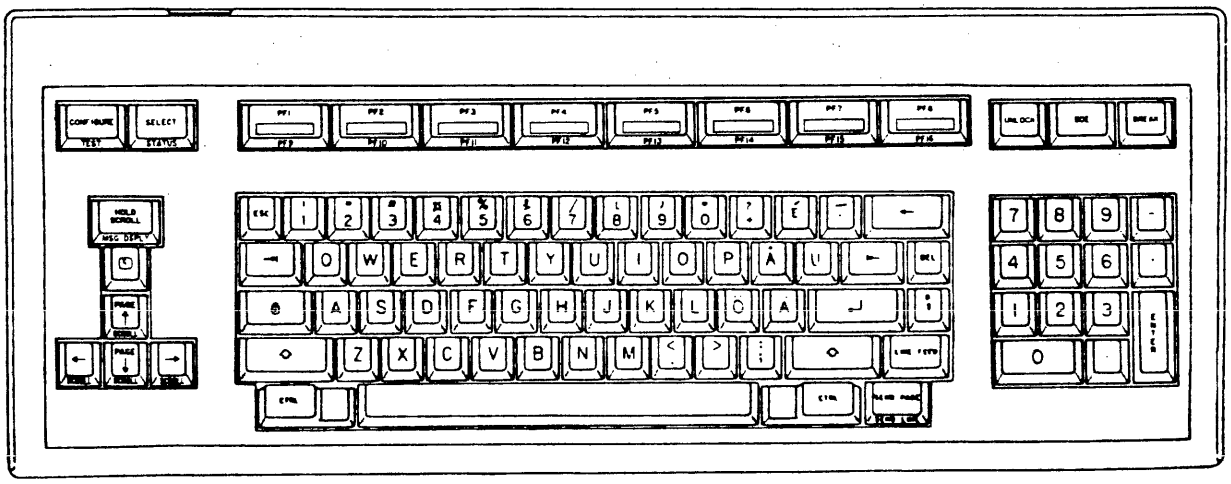
Norwegian Code Chart



Norwegian Keyboard Layout

	CONTROL CHARACTERS		DISPLAYABLE CHARACTERS					
BIT 7 6 / 4 3 2 1	0_0	0_1	0_{10}	0_{11}	1_{00}	1_{01}	1_{10}	1_{11}
0000	NUL @	DLE P	SP	0	@	P	`	p
0001	SOH A	DC1 X-ON Q	!	1	A	Q	a	q
0010	STX B	DC2 R	"	2	B	R	b	r
0011	ETX C	DC3 X-OFF S	#	3	C	S	c	s
0100	EOT D	DC4 T	¤	4	D	T	d	t
0101	ENQ E	NAK U	%	5	E	U	e	u
0110	ACK F	SYN V	&	6	F	V	f	v
0111	BEL G	ETB XMT W	'	7	G	W	g	w
1000	BS H	CAN X	(	8	H	X	h	x
1001	HT I	EM Y	)	9	I	Y	i	y
1010	LF J	SUB Z	*	:	J	Z	j	z
1011	VT K	ESC Ä	+	;	K	Ä	k	ä
1100	FF L	FS Ö	,	<	L	Ö	l	ö
1101	CR M	GS Å	-	=	M	Å	m	å
1110	SO N	RS Ü	.	>	N	Ü	n	ü
1111	SI O	US _	/	?	O	_	o	DEL

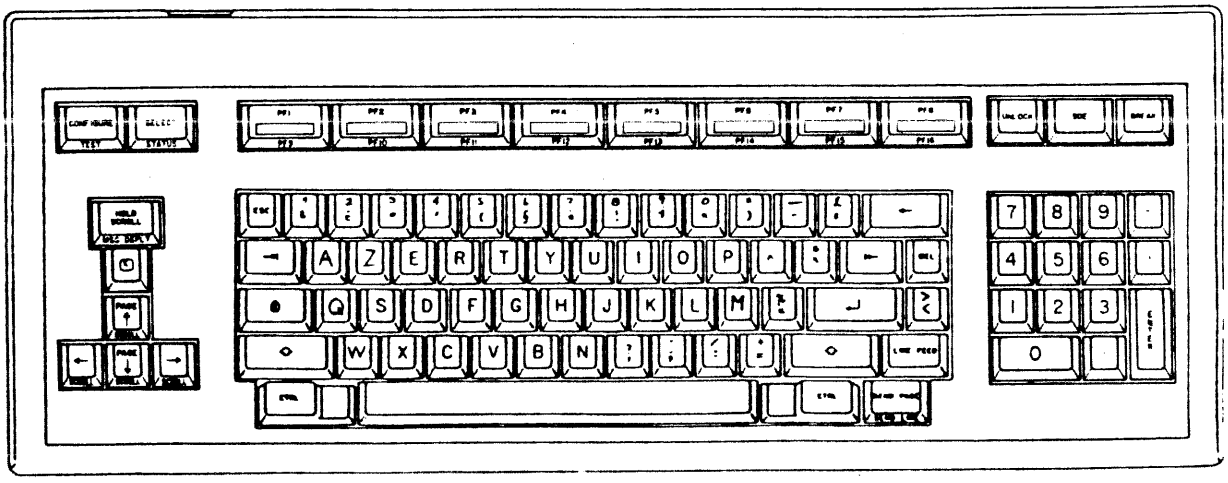
Swedish Code Chart



Swedish Keyboard Layout

BIT 7 6 / 4 3 2 1 5	CONTROL CHARACTERS		DISPLAYABLE CHARACTERS					
	0_0	0_1	$0_1 0$	$0_1 1$	$1_0 0$	$1_0 1$	$1_1 0$	$1_1 1$
0000	NUL @	DLE P	SP	0	à	P	^	p
0001	SOH A	DC1 Q X-ON	!	1	A	Q	a	q
0010	STX B	DC2 R	"	2	B	R	b	r
0011	ETX C	DC3 S X-OFF	£	3	C	S	c	s
0100	EOT D	DC4 T	$	4	D	T	d	t
0101	ENQ E	NAK U	%	5	E	U	e	u
0110	ACK F	SYN V	&	6	F	V	f	v
0111	BEL G	ETB W XMIT	'	7	G	W	g	w
1000	BS H	CAN X	(	8	H	X	h	x
1001	HT I	EM Y	)	9	I	Y	i	y
1010	LF J	SUB Z	*	:	J	Z	j	z
1011	VT K	ESC Ä	+	;	K	Ä	k	ä
1100	FF L	FS Ö	,	<	L	Ö	l	ö
1101	CR M	GS Ü	-	=	M	Ü	m	ü
1110	SO N	RS ^	.	>	N	^	n	ß
1111	SI O	US _	/	?	O	_	o	DEL

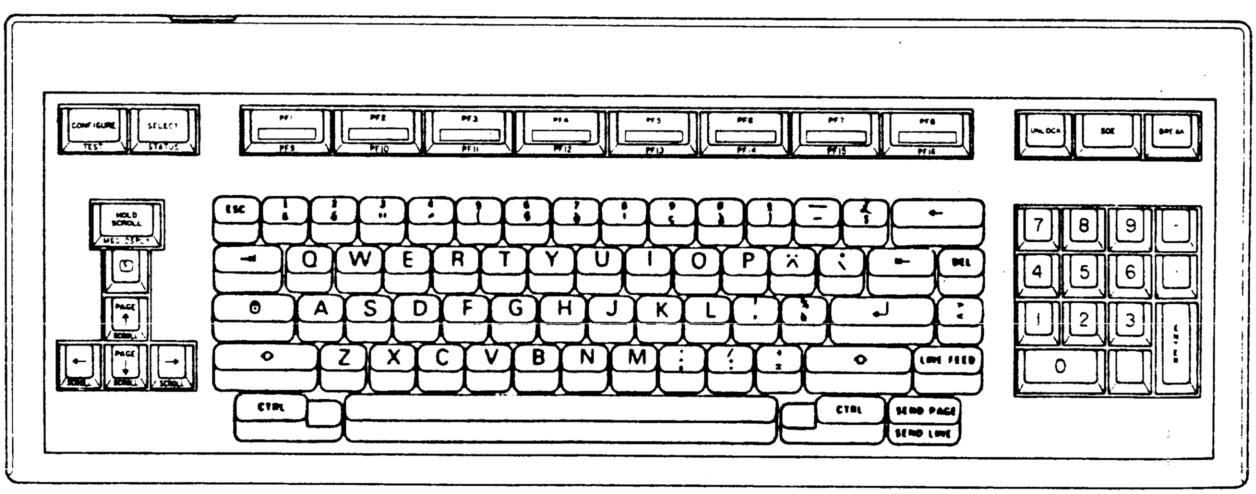
French Code Chart



French Keyboard Layout

BIT 7 6 5 / 4 3 2 1	000	001	010	011	100	101	110	111
	CONTROL CHARACTERS		DISPLAYABLE CHARACTERS					
0000	NUL @	DLE P	SP	0	à	P	`	p
0001	SOH A	DC1 X-ON Q	!	1	A	Q	a	q
0010	STX B	DC2 R	"	2	B	R	b	r
0011	ETX C	DC3 X-OFF S	£	3	C	S	c	s
0100	EOT D	DC4 T	$	4	D	T	d	t
0101	ENQ E	NAK U	%	5	E	U	e	u
0110	ACK F	SYN V	&	6	F	V	f	v
0111	BEL G	ETB XMIT W	'	7	G	W	g	w
1000	BS H	CAN X	(	8	H	X	h	x
1001	HT I	EM Y	)	9	I	Y	i	y
1010	LF J	SUB Z	*	:	J	Z	j	z
1011	VT K	ESC â	+	;	K	â	k	é
1100	FF L	FS ç	,	<	L	ç	l	ù
1101	CR M	GS §	-	=	M	§	m	è
1110	SO N	RS ^	.	>	N	^	n	¨
1111	SI O	US _	/	?	O	_	o	DEL

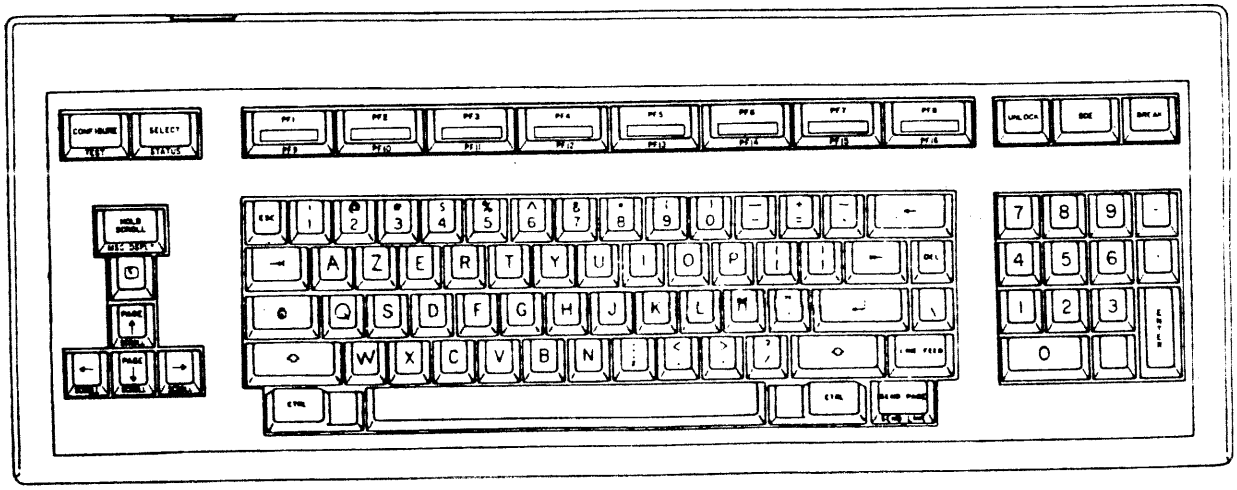
Canadian French Code Chart



Canadian French Keyboard Layout

BIT 7 6 / 4 3 2 1	CONTROL CHARACTERS		DISPLAYABLE CHARACTERS					
	0_0	0_1	$0_1 0$	$0_1 1$	$1_0 0$	$1_0 1$	$1_1 0$	$1_1 1$
0000	NUL @	DLE P	SP	0	@	P	`	p
0001	SOH A	DC1 Q X-ON	!	1	A	Q	a	q
0010	STX B	DC2 R	"	2	B	R	b	r
0011	ETX C	DC3 S X-OFF	#	3	C	S	c	s
0100	EOT D	DC4 T	$	4	D	T	d	t
0101	ENQ E	NAK U	%	5	E	U	e	u
0110	ACK F	SYN V	&	6	F	V	f	v
0111	BEL G	ETB W XMIT	'	7	G	W	g	w
1000	BS H	CAN X	(	8	H	X	h	x
1001	HT I	EM Y	)	9	I	Y	i	y
1010	LF J	SUB Z	*	:	J	Z	j	z
1011	VT K	ESC [	+	;	K	[	k	{
1100	FF L	FS \	,	<	L	\	l	¦
1101	CR M	GS]	-	=	M	]	m	}
1110	SO N	RS ^	.	>	N	^	n	~
1111	SI O	US _	/	?	O	_	o	DEL

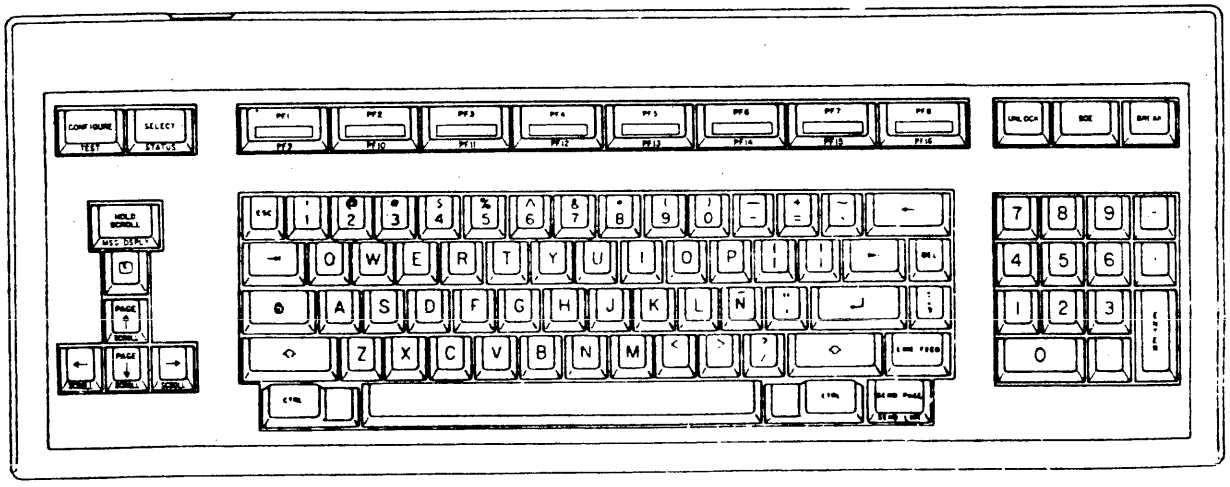
French Lower Level Code Chart



French Lower Level Keyboard Layout

BIT 7 6 5 4 3 2 1 0	CONTROL CHARACTERS		DISPLAYABLE CHARACTERS					
	0_0	0_1	$0_1 0$	$0_1 1$	$1_0 0$	$1_0 1$	$1_1 0$	$1_1 1$
0000	NUL @	DLE P	SP	0	@	P	`	p
0001	SOH A	DC1 X-ON Q	!	1	A	Q	a	q
0010	STX B	DC2 R	"	2	B	R	b	r
0011	ETX C	DC3 X-OFF S	#	3	C	S	c	s
0100	EOT D	DC4 T	$	4	D	T	d	t
0101	ENQ E	NAK U	%	5	E	U	e	u
0110	ACK F	SYN V	&	6	F	V	f	v
0111	BEL G	ETB XMIT W	'	7	G	W	g	w
1000	BS H	CAN X	(	8	H	X	h	x
1001	HT I	EM Y	)	9	I	Y	i	y
1010	LF J	SUB Z	*	:	J	Z	j	z
1011	VT K	ESC [	+	;	K	[	k	{
1100	FF L	FS $\tilde{N}$	,	<	L	$\tilde{N}$	l	$\tilde{n}$
1101	CR M	GS]	-	=	M	]	m	}
1110	SO N	RS ^	.	>	N	^	n	~
1111	SI O	US -	/	?	O	_	o	DEL

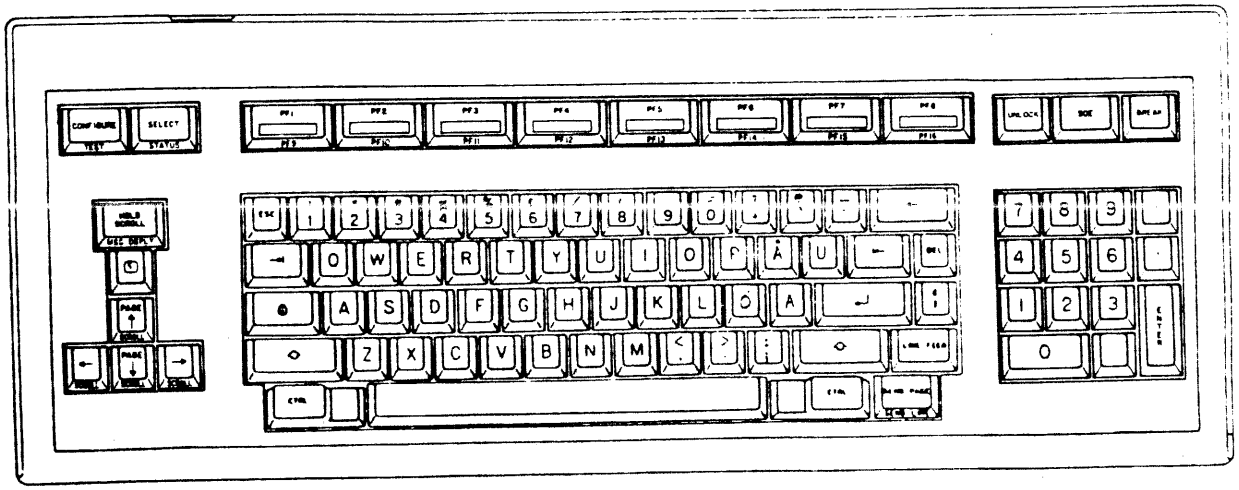
Spanish Code Chart



Spanish Keyboard Layout

BIT 7 6 5 4 3 2 1 0	CONTROL CHARACTERS 0_0	0_1	DISPLAYABLE CHARACTERS 0_{1_0}	0_{1_1}	1_{0_0}	1_{0_1}	1_{1_0}	1_{1_1}
0000	NUL @	DLE P	SP	0	@	P	`	p
0001	SOH A	DC1 X-ON Q	!	1	A	Q	a	q
0010	STX B	DC2 R	"	2	B	R	b	r
0011	ETX C	DC3 X-OFF S	#	3	C	S	c	s
0100	EOT D	DC4 T	X	4	D	T	d	t
0101	ENQ E	NAK U	%	5	E	U	e	u
0110	ACK F	SYN V	&	6	F	V	f	v
0111	BEL G	ETB XMIT W	'	7	G	W	g	w
1000	BS H	CAN X	(	8	H	X	h	x
1001	HT I	EM Y	)	9	I	Y	i	y
1010	LF J	SUB Z	*	:	J	Z	j	z
1011	VT K	ESC Ä	+	;	K	Ä	k	ä
1100	FF L	FS Ö	,	<	L	Ö	l	ö
1101	CR M	GS Å	-	=	M	Å	m	å
1110	SO N	RS Ü	.	>	N	Ü	n	ü
1111	SI O	US -	/	?	O	_	o	DEL

Finnish Code Chart



Finnish Keyboard Layout

APPENDIX C. CHARACTER GENERATOR

The character generator of the ATL-008 is capable of generating 256 characters. The characters are numbered from 0 to 255. Characters 33 through 255 are considered displayable characters. Characters 0 through 31 are displayable representations of control codes. The control codes can only be displayed using monitor mode (See the description of the MONITOR softkey in Section 5).

C.1 Displaying Special Characters

The first 128 characters in the ATL-008 character generator (00H - 7FH) are the standard ASCII character set (see also Appendix A). These characters can be generated by pressing single keys on the keyboard. (The terminal must be in the proper modes to display some characters.) The remaining characters (80H - FFH) can be displayed using escape sequences.

C.1.1 ANSI Mode

When the terminal is in ANSI mode, characters in the range 80H through FFH are accessed by the SS2 and SS3 control sequences (discussed in Section 6) Use the parameters in Table C-1. For example, to display the character with a hexadecimal value of EFH, send [ESC O 2] to the ATL-008.

C.1.2 100 Mode

Use the Select Character Set (SCS) control sequence (described in Section 7) to display the special graphics characters shown in Table 7-3.

NOTE: Except for the special graphics characters listed in Table 7-3 and any foreign letters that form part of the currently selected foreign character set (see the Nationality setting in Installation TCM in Section 3), characters in the range 80H to FFH cannot be accessed while the ATL-008 is in 100 mode.

C.1.3 52 Mode

On entering 52 mode from 100 mode, the last character set selected in 100 mode remains valid unless an Enter Graphics Mode or Exit Graphics Mode command is received. See Section 7.

Table C-1
Codes For Use With SS2 and SS3 Control Sequences

HEX - ASCII	SS2	SS3	HEX - ASCII	SS2	HEX - ASCII	SS2
21H - !	80H	DEH	43H - C	A2H	65H - e	C4H
22H - "	81H	DFH	44H - D	A3H	66H - f	C5H
23H - #	82H	E0H	45H - E	A4H	67H - g	C6H
24H - $	83H	E1H	46H - F	A5H	68H - h	C7H
25H - %	84H	E2H	47H - G	A6H	69H - i	C8H
26H - &	85H	E3H	48H - H	A7H	6AH - j	C9H
27H - '	86H	E4H	49H - I	A8H	6BH - k	CAH
28H - (	87H	E5H	4AH - J	A9H	6CH - l	CBH
29H -)	88H	E6H	4BH - K	AAH	6DH - m	CCH
2AH - *	89H	E7H	4CH - L	ABH	6EH - n	CDH
2BH - +	8AH	E8H	4DH - M	ACH	6FH - o	CEH
2CH - ,	8BH	E9H	4EH - N	ADH	70H - p	CFH
2DH - -	8CH	EAH	4FH - O	AEH	71H - q	D0H
2EH - .	8DH	EBH	50H - P	AFH	72H - r	D1H
2FH - /	8EH	ECH	51H - Q	B0H	73H - s	D2H
30H - 0	8FH	EDH	52H - R	B1H	74H - t	D3H
31H - 1	90H	EEH	53H - S	B2H	75H - u	D4H
32H - 2	91H	EFH	54H - T	B3H	76H - v	D5H
33H - 3	92H	F0H	55H - U	B4H	77H - w	D6H
34H - 4	93H	F1H	56H - V	B5H	78H - x	D7H
35H - 5	94H	F2H	57H - W	B6H	79H - y	D8H
36H - 6	95H	F3H	58H - X	B7H	7AH - z	D9H
37H - 7	96H	F4H	59H - Y	B8H	7BH - {	DAH
38H - 8	97H	F5H	5AH - Z	B9H	7CH - |	DBH
39H - 9	98H	F6H	5BH - [	BAH	7DH - }	DCH
3AH - :	99H	F7H	5CH - \	BBH	7EH - ~	DDH
3BH - ;	9AH	F8H	5DH -]	BCH		
3CH - <	9BH	F9H	5EH - ^	BDH		
3DH - =	9CH	FAH	5FH - _	BEH		
3EH - >	9DH	FBH	60H - `	BFH		
3FH - ?	9EH	FCH	61H - a	C0H		
40H - @	9FH	FDH	62H - b	C1H		
41H - A	A0H	FEH	63H - c	C2H		
42H - B	A1H	FFH	64H - d	C3H		

C.1.4 Printing Special Characters

To successfully print special characters, the serial printer must be able to process the appropriate control sequence(s) (SS2, SS3, SCS, SI, SO).

C.2 Description of Characters

The displayed shapes of the NUL and BEL codes differ from the shapes displayed by BEEHIVE™ DM series terminals. This was done to conform to the ANSI and ISO specifications describing their shapes.

The foreign characters are arranged to keep the same offset between upper- and lower-case characters to allow the shift code to perform normally. Further, certain word processing symbols are placed to allow the shift to select between two similar shapes.

The various arrow icons are arranged in the same order: up, down, right, and left.

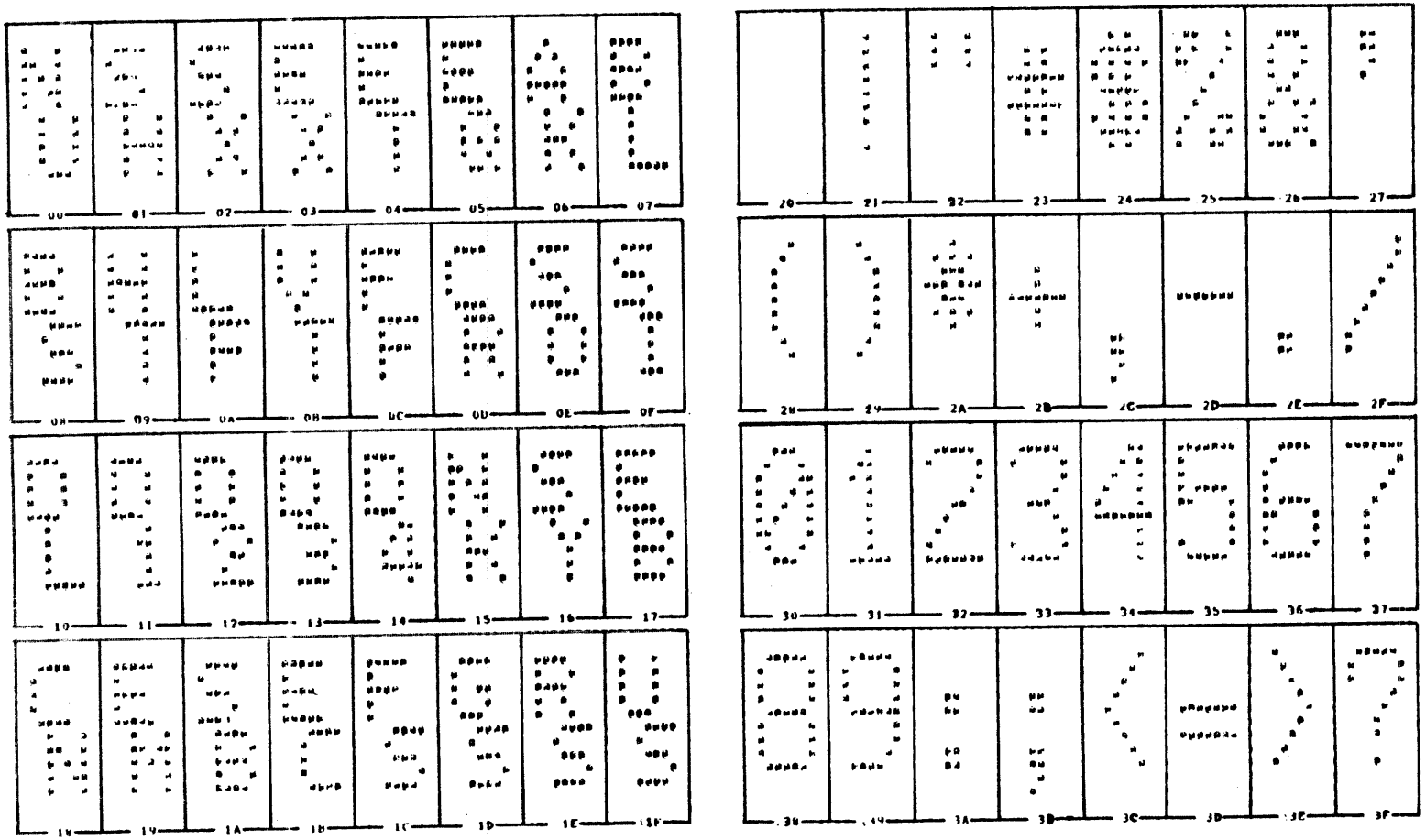
The histogram business graphics are placed to ease computation of the type of character displayed. For each character there is an open and a filled character. Their codes differ by a fixed constant.

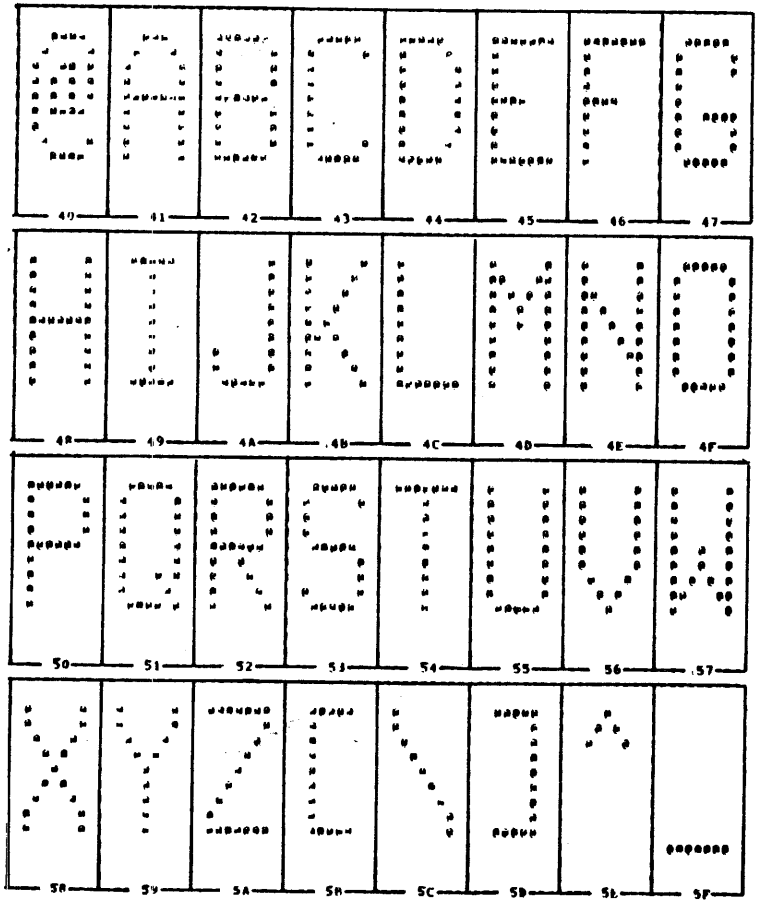
The bar code graphics allow the display of bar code on the terminal screen.

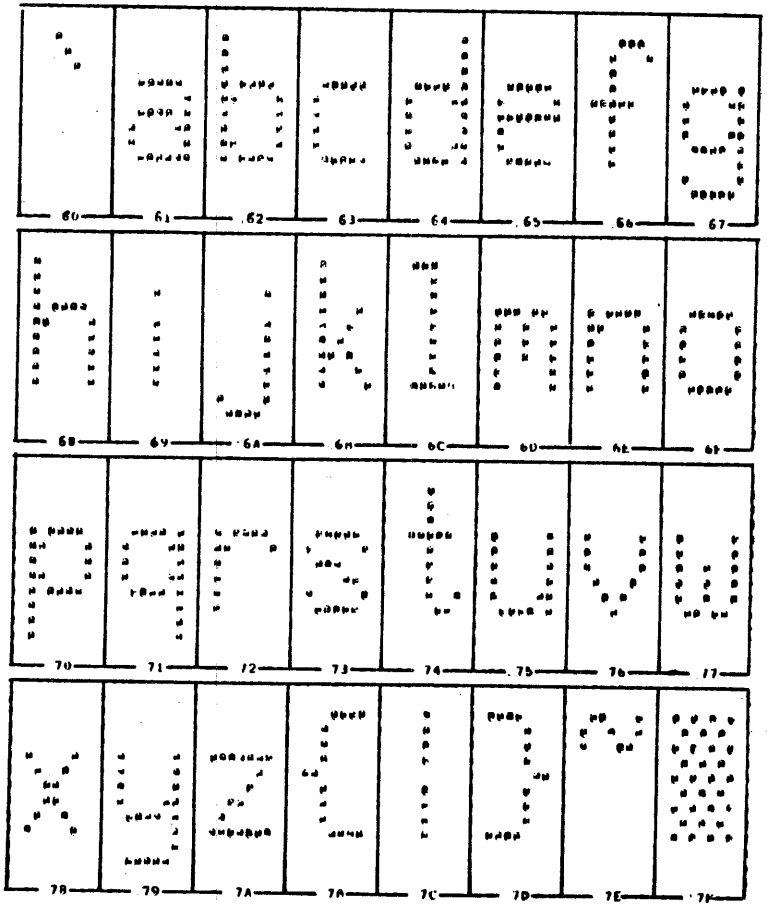
C.3 Character Generator Character Chart

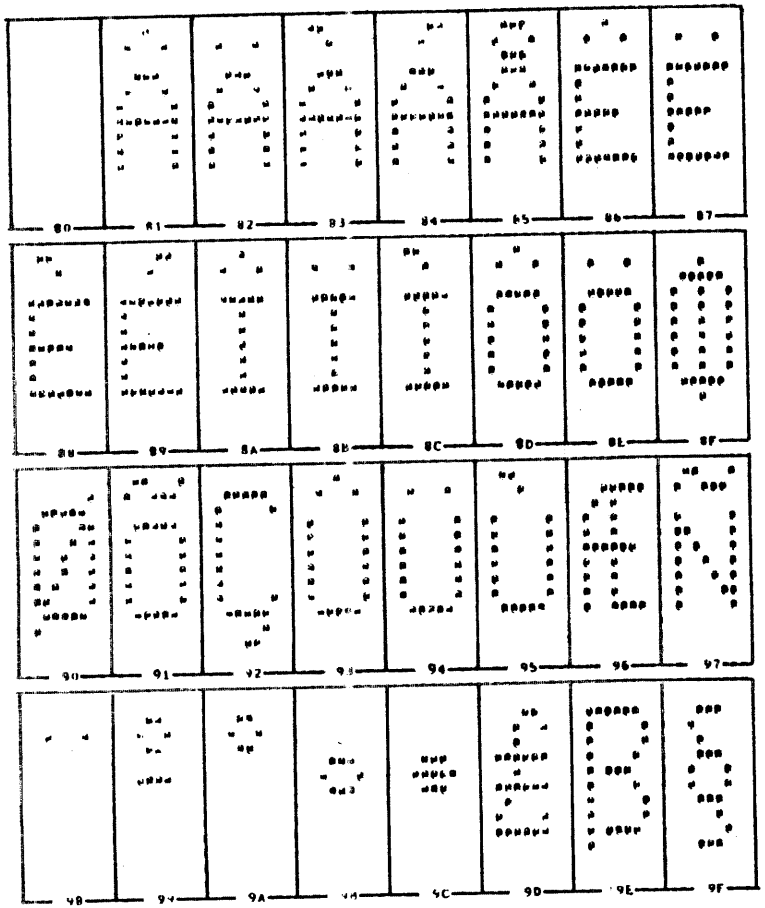
The following pages contain graphic representations of each character available in the ATL-008 character generator.

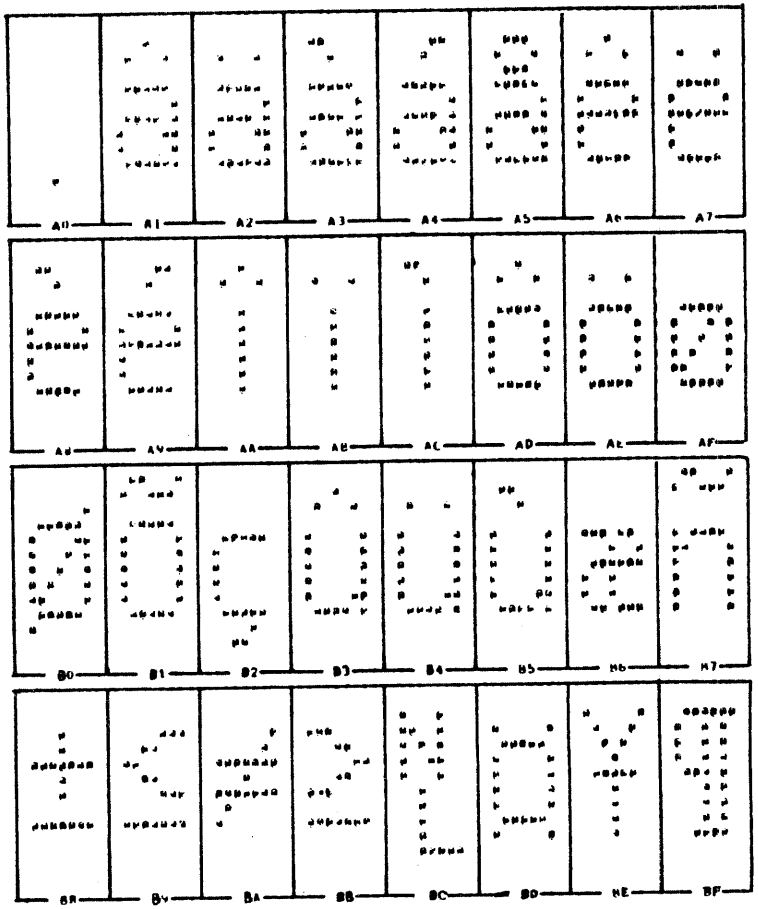
™ BEEHIVE is a trademark of Beehive International

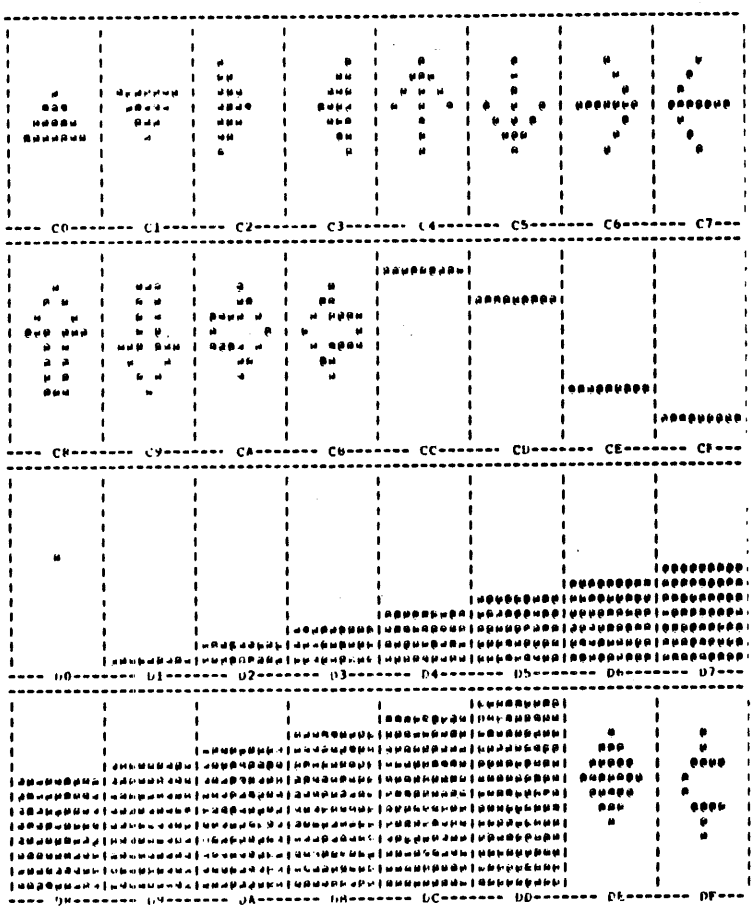


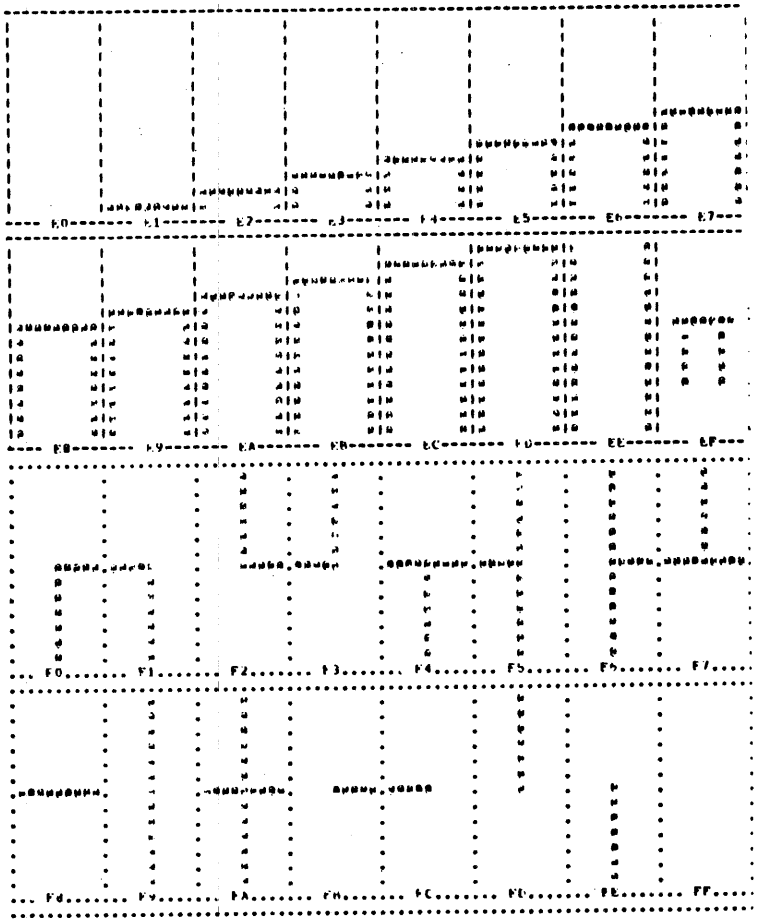

40
41
42
43
44
45
46
47
48
49
4A
4B
4C
4D
4E
4F
50
51
52
53
54
55
56
57
58
59
5A
5B
5C
5D
5E
5F


60
61
62
63
64
65
66
67
68
69
6A
6B
6C
6D
6E
6F
70
71
72
73
74
75
76
77
78
79
7A
7B
7C
7D
7E
7F









APPENDIX D. TCM MAP

This appendix gives the parameters which must be passed to the ATL-008 in a DCS sequence in order to change TCM parameters. See the description of the DCS sequence in Section 6 for details.

<Pn> Parameter number. Identifies the parameter to be downloaded or uploaded. 4 ASCII digits padded with leading digits if necessary.

NOTE:

- The parameter numbers <Pn> should not be considered addresses or offsets into memory.

reset An R in this column indicates that the terminal must be reset before the change takes effect.

data nvram data.

- Yes/no selections are specified with an ASCII 1 or 0.
- All numeric parameters (page, buffer sizes) are specifed using decimal numbers in ASCII.
- All character string parameters (passwords, answer-back) are specifed in hex format to allow the inclusion of control characters.
- The data may include an identifier (page number, buffer id) as well as the data to be saved.

parameter Identifies the TCM parameter.

value The value assigned to the parameter which corresponds to the selection entered for **data**.

<Pn>	reset	data	parameter	value
0001	R	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14	* main baud rate	50 baud 110 baud 134.5 baud 150 baud 200 baud 300 baud 600 baud 1200 baud 1800 baud 2000 baud 2400 baud 4800 baud 9600 baud 19200 baud external clock
0002	R	0 1 2 3 4	* main parity	even odd mark space none
0003	R	0 1 2	* main stop bits	1.0 stop bits 1.5 2.0
0004	R	0 1	* main data length	7 bit 8 bit
0005	R	0 1	* main duplex	full half
0006		0 1	* bit 19 is auto echo bit	((off?)) ((on?))
0007	R	0 1 2	* hardware interface type	RS-232 RS-422 current loop

<Pn>	reset	data	parameter	value
0008	R	0 1	* xon/xoff flag	no yes
0009		0 1	* space suppression	no yes
0010		0 1	* CR / CRLF to host	CR CRLF
0011	R	0 1	* parity error checking	no yes
0012		0 1 2 3 4	* main transmission mode	character line partial page page modified page
0013		0 1 2 3 4	* main termination character	ETX EOT CR CRLF none
0014	R	0 1	* modem type	standard 212a
0015	R	0 1	* main cts required	no yes
0016	R	0 1	* main dsr required	no yes
0017	R	0 1	* main dcd required	yes yes

<Pn>	reset	data	parameter	value
0018	R	0 1 2 3 4 5 6	* rts delay	0 msec 5 msec 10 msec 15 msec 20 msec 50 msec 100 msec
0019	R	00 01 02 03 04 05 06 07 08 09 10	* cts delay	0 msec 5 msec 10 msec 15 msec 20 msec 50 msec 100 msec 150 msec 200 msec 250 msec 300 msec
0020	R	00 01 02 03 04 05 06 07 08 09 10	* tx to rx delay	0 msec 5 msec 10 msec 15 msec 20 msec 50 msec 100 msec 150 msec 200 msec 250 msec 300 msec
0021		0 1	* auto answer back	no yes
0022		hex	* answer-back message	hex data terminated by ESC or ;

<Pn>	reset	data	parameter	value
0023	R	00	* aux baud rate	50 baud
		01		75 baud
		02		110 baud
		03		134.5 baud
		04		150 baud
		05		200 baud
		06		300 baud
		07		600 baud
		08		1200 baud
		09		1800 baud
		10		2000 baud
		11		2400 baud
		12		4800 baud
		13		9600 baud
		14		19200 baud
0024	R	0	* aux parity	even
		1		odd
		2		mark
		3		space
		4		none
0024	R	0	* aux stop bits	1.0 bits
		1		1.5
		2		2.0
0026	R	0	* aux data length	7 bits
		1		8 bits
0027	R	0	* printer protocol	none
		1		pin 4 rts
		2		pin 11/19 ready
		3		pin 11/19 busy
		4		pin 11/19 rev chan
		5		pin 20 dtr
		6		xon/xoff
		7		ack/nak
		8		etx/ack
		9		tty

<Pn>	reset	data	parameter	value
0028	R	0 1	* aux port external clock	no yes
0029		0 1	* space suppression	no yes
0030		0 1	* CR / CRLF to aux	CR CRLF
0031	R	0 1	* aux parity checking	no yes
0032	R	0 1	* single/double buffer	double single
0033		0 1	* send form feed after block	no yes
0034		0 1	* dc2/dc4 transparent printing	no yes
0035		dec	* number of pad chars	number of pad chars
0036		hex	* pad char	pad char
0037	R	0 1	* power line frequency bit	50hz 60hz
0038	R	0 1	* video full/half intensity	full half
0039	R	0 1 2 3	* cursor type bits	underline block underline, blink block, blink

<Pn>	reset	data	parameter	value
0040	R	0 . . . 15	* brightness level bits	dimmest . . . brightest
0041	R	0 1	* smooth scroll	jump smooth
0042		0 1 2 3	* smooth scroll rate	2 lines/sec 5 lines/sec 9 lines/sec 30 lines/sec
0043	R	0 1	* 132 col display	80 132
0044	R	0 1	* reverse video	normal reverse

<Pn>	reset	data	parameter	value
0045	R	hex	* x3.64 mode word	8 hex digits
			Mode Bit reserved bit 0 gatm bit 1 kam bit 2 crm bit 3 irm bit 4 srtm bit 5 erm bit 6 vem bit 7 reserved bit 8 reserved bit 9 hem bit 10 pum bit 11 srm bit 12 feam bit 13 fetm bit 14 matm bit 15 ttm bit 16 satm bit 17 tsm bit 18 ebm bit 19 lnm bit 20 reserved bit 21 reserved bit 22 reserved bit 23 reserved bit 24 reserved bit 25 reserved bit 26 reserved bit 27 reserved bit 28 reserved bit 29 reserved bit 30 reserved bit 31	
0046		0 1	* terminal mode	ansi vt100

<Pn>	reset	data	parameter	value
0047	R	0 1	* vt52 / vt100	vt52 vt100
0048	R	0 1	* on line	local online
0049	R	0 1 2 3	* SEM selection	edit in display edit in line edit in field edit in qualified area
0050	R	0 1	* qa auto-tab	no yes
0051	R	0 1	* destructive roll	no yes
0052	R	0 1	* VEM	no yes
0053	R	0 1	* HEM	no yes
0054		0 1	* ANSI status line	no yes
0055		0 1	* status reporting	no yes
0056		0 1 2 3 4 5 6 7	* auto repeat rate	50 cps 28 cps 20 cps 18 cps 14 cps 10 cps 6 cps ramp up

<Pn>	reset	data	parameter	value
0057		0 1 2 3 4 5	* hold down delay time	.50 sec .75 sec .80 sec 1.00 sec 1.50 sec 2.50 sec
0058		0 1	* lower case inhibit	no yes
0059		0 1	* keyboard clicker	no yes
0060		0 1	* keyboard sound volume	low high
0061		0 1	* shift lock key effect	all keys alpha keys
0062		0 1	* shift lock key off usage	shift shift lock
0063		0 1	* soft keys local/follow io	local follow io
0064		0 1	* repeated keys	typewriter IBM
0065		0 1	* new line key	CR CRLF
0066		0 1 2	* enter key definition	newline key send key termination char

<Pn>	reset	data	parameter	value
0067		0 1	* auto-wrap	no yes
0068		00 01 02 03 04 05 06 07 08 09 10	* nationality	ascii united kingdom german danish norwegian sweedish french canadian french french lower case spanish finnish
0069		0 1	* over strike	no yes
0070	R	dec	* bell col	columns from edge
0071		dec	* powerup soft key menu	menu #
0072		<key#>,0 <key#>,1 <key#>,2 <key#>,3 <key#>,4	* start of pf keys	APC OSC PM DCS SS3

Example:

```
ESC P  W  0072  01,0  ;  02,4  ;  16,2  ESC\
```

sets PF1 to send an APC, PF2 to send an SS3, and PF16 to send a PM.

<Pn>	reset	data	parameter	value
0073			<key#>,hex * graphic soft keys defn	hex value of char
Example:				
			ESC P W 0073 01,21 ; 02,40 ; 09,3F ESC\	
sets graphics softkey 1 to display an exclaimation point (!), graphics softkey 2 to display an at sign (@), and graphics softkey 9 to display a question mark (?).				
0074		hex	* start of tabs	hex bytes with bits set for tab stops
0075		0 1	* auto-sign off	no yes
0076		dec	* auto sign off time	# of minutes
0077		0 1	* screen saver yes/no	no yes
0078		dec	* screen saver time	# of minutes
0079	R	dec	* main receiver buffers	# of buffers
0080	R	dec	* main transmitter buffers	# of buffers
0081	R	dec	* aux receiver buffers	# of buffers
0082	R	dec	* aux transmitter buffers	# of buffers
0083	R	dec	* qualified area buffers	# of buffers
0084	R	dec	* selected area buffers	# of buffers

<Pn>	reset	data	parameter	value
0085	R	dec	* user keys buffers	# of buffers
0086	R	dec	* bootload	# of buffers
0087		0 1	* installation tcm access	free password
0088		hex	* installation tcm password	8 hex digits
0089		0 1 2	* program key access	free access control-shift password
0090		hex	* program key password	8 hex digits
0091		hex	* keyboard auto-repeat table	hex bytes
0092	R	dec	* page defn	page number 2 digits page length 3 digits page width (80 col disp) 3 digits page width (132 col disp) 3 digits window height 3 digits window width (80 col disp) 3 digits window width (132 col disp) 3 digits window home row 3 digits window home col (80 col disp) 3 digits window home col (132 col disp) 3 digits

APPENDIX E. COMMAND SUMMARY

E.1 ANSI AND 100 MODE COMMANDS

The ANSI and 100 mode control codes, control sequences, and mode changing parameters are divided into five tables as follows:

CONTROL CODES - These "sequences" consist of a single code - the command code as shown in the table.

INDEPENDENT CONTROL FUNCTIONS - These sequences are two code sequences. The first code is the 1BH or ESC code. The second code is the command code as shown in the table.

CONTROL SEQUENCES WITHOUT INTERMEDIATES - These sequences may be of varying length, but are characterized by the ESC [on the front end and the command code (as shown in the table) on the tail end.

CONTROL SEQUENCES WITH INTERMEDIATES - These sequences may be of varying length, but are characterized by the ESC [on the front end and a space followed by the command code (as shown in the table) on the tail end.

MODE CHANGING PARAMTERS - This is a table of parameters for the SM (set mode) and RM (reset mode) command sequences.

In each table, there are two columns, one for 100 mode and one for ANSI. An x in a column next to a command indicates that the command is implemented in that mode. A - means that the command is not implemented.

E.1.1 CONTROL CODES

HEX - ASCII	FUNCTION	100	ANSI
00H - NUL		-	-
01H - SOH		-	-
02H - STX		-	-
03H - ETX		-	-
04H - EOT		-	-
05H - ENQ	send answer-back	x	x
06H - ACK		-	-
07H - BEL	ring bell	x	x
08H - BS	backspace	x	x
09H - HT	horizontal tab	x	x
0AH - LF	line feed	x	x
0BH - VT	line feed vertical tab	-	x
0CH - FF	line feed form feed	-	x
0DH - CR	carriage return	x	x
0EH - SO	shift out	x	x
0FH - SI	shift in	x	x
10H - DLE		-	-
11H - DC1	XON	x	x
12H - DC2	printer control	x	x
13H - DC3	XOFF	x	x
14H - DC4	printer control	x	x
15H - NAK		-	-
16H - SYN		-	-
17H - ETB		-	-
18H - CAN	terminate control sequence	x	x
19H - EM		-	-
1AH - SUB	terminate control sequence	x	x
1BH - ESC	start escape sequence	x	x
1CH - FS		-	-
1DH - GS		-	-
1EH - RS		-	-
1FH - US		-	-

E.1.2 INDEPENDENT CONTROL FUNCTIONS ESC Fe

Fe HEX - ASCII	CONTROL SEQUENCE	100	ANSI
20H - SP		-	-
21H - !		-	-
22H - "		-	-
23H - #	set row visual attributes	x	-
24H - $		-	-
25H - %		-	-
26H - &		-	-
27H - '		-	-
28H - (	shift in character set	x	x
29H -)	shift out character set	x	x
2AH - *		-	-
2BH - +		-	-
2CH - ,		-	-
2DH - -		-	-
2EH - .	EXIT 100 MODE	x	-
2FH - /		-	-
30H - 0		-	-
31H - 1		-	-
32H - 2		-	-
33H - 3		-	-
34H - 4		-	-
35H - 5	DECXMIT transmit page	x	x
36H - 6		-	-
37H - 7	save cursor, cursor char and attribute	x	-
38H - 8	restore cursor, cursor char and attr	x	-
39H - 9		-	-
3AH - :		-	-
3BH - ;		-	-
3CH - <		-	-
3DH - =	keypad application mode	x	-
3EH - >	keypad numeric mode	x	-
3FH - ?		-	-
40H - @		-	-
41H - A		-	-
42H - B		-	-

Fe HEX - ASCII	CONTROL SEQUENCE	100	ANSI
43H - C		-	-
44H - D	index	x	x
45H - E	next line	x	x
46H - F	start of selected area	x	x
47H - G	end of slected area	x	x
48H - H	horizontal tab set	x	x
49H - I	horizontal tab with justification	x	x
4AH - J	vertical tab set	x	x
4BH - K	partial line down	-	-
4CH - L	partial line up	-	-
4DH - M	reverse index	x	x
4EH - N	single shift 2	x	x
4FH - O	single shift 3	x	x
50H - P	device control string	x	x
51H - Q	private use 1	x	x
52H - R	private use 2	x	x
53H - S	set transmit state	x	x
54H - T	cancel character	x	x
55H - U	message waiting	x	x
56H - V	start of protected area	x	x
57H - W	end of protected area	x	x
58H - X		-	-
59H - Y		-	-
5AH - Z	identify terminal	x	x
5BH - [	control sequence introducer	x	x
5CH - \	string terminator	x	x
5DH -]	operating system control	-	-
5EH - ^	privacy message	-	-
5FH - _	application program control	x	x
60H - `	disable manual input	x	x
61H - a	interrupt	x	x
62H - b	enable manual input	x	x
63H - c	reset to initial state	x	x
64H - d		-	-
65H - e		-	-
66H - f		-	-
67H - g		-	-
68H - h		-	-

Fe HEX - ASCII	CONTROL SEQUENCE	100	ANSI
69H - i		-	-
6AH - j		-	-
6BH - k		-	-
6CH - l		-	-
6DH - m		-	-
6EH - n		-	-
6FH - o		-	-
70H - p		-	-
71H - q		-	-
72H - r		-	-
73H - s		-	-
74H - t		-	-
75H - u		-	-
76H - v		-	-
77H - w		-	-
78H - x		-	-
79H - y		-	-
7AH - z		-	-
7BH - {		-	-
7CH - |		-	-
7DH - }		-	-
7EH - ~		-	-
7FH - DEL		-	-

E.1.3 CONTROL SEQUENCES WITHOUT INTERMEDIATES ESC [P ... P F

F HEX - ASCII	CONTROL SEQUENCE	100	ANSI
20H - SP	intermediate	x	x
21H - !		-	-
22H - "		-	-
23H - #		-	-
24H - $		-	-
25H - %		-	-
26H - &		-	-
27H - '		-	-
28H - (		-	-
29H -)		-	-
2AH - *		-	-
2BH - +		-	-
2CH - ,		-	-
2DH - -		-	-
2EH - .		-	-
2FH - /		-	-
30H - 0		-	-
31H - 1		-	-
32H - 2		-	-
33H - 3		-	-
34H - 4		-	-
35H - 5		-	-
36H - 6		-	-
37H - 7		-	-
38H - 8		-	-
39H - 9		-	-
3AH - :		-	-
3BH - ;		-	-
3CH - <		-	-
3DH - =		-	-
3EH - >		-	-
3FH - ?		-	-
40H - @	insert character	x	x
41H - A	cursor up	x	x
42H - B	cursor down	x	x
43H - C	cursor forward	x	x
44H - D	cursor backwards	x	x
45H - E	cursor next line	x	x
46H - F	cursor previous line	x	x

F HEX - ASCII	CONTROL SEQUENCE	100	ANSI
47H - G	cursor horizontal absolute	x	x
48H - H	cursor position	x	x
49H - I	cursor horiztonal tab	x	x
4AH - J	erase in display	x	x
4BH - K	erase in line	x	x
4CH - L	insert line	x	x
4DH - M	delete line	x	x
4EH - N	erase in field	x	x
4FH - O	erase in area	x	x
50H - P	delete character	x	x
51H - Q	select editing extent mode	x	x
52H - R	cursor position report	x	x
53H - S	scroll up	x	x
54H - T	scroll down	x	x
55H - U	next page	x	x
56H - V	preceding page	x	x
57H - W	cursor tab control	x	x
58H - X	erase character	x	x
59H - Y	cursor vertical tab	x	x
5AH - Z	cursor backtab	x	x
5BH - [		-	-
5CH - \		-	-
5DH -]		-	-
5EH - ^		-	-
5FH - _		-	-
60H - `	horizontal position absolute	x	x
61H - a	horizontal position relative	x	x
62H - b	repeat	x	x
63H - c	send device attributes	x	x
64H - d	vertical position absolute	x	x
65H - e	vertical position relative	x	x
66H - f	horizontal and vertical positioning	x	x
67H - g	tab clear	x	x

F HEX - ASCII	CONTROL SEQUENCE	100	ANSI
68H - h	set mode	x	x
69H - i	media copy	x	x
6AH - j		-	-
6BH - k		-	-
6CH - l	reset mode	x	x
6DH - m	select graphic rendition	x	x
6EH - n	device status report	x	x
6FH - o	define area qualification	x	x
70H - p	set soft key menu	x	x
71H - q	load LEDs	x	x
72H - r	set top and bottom margin	x	-
73H - s	align pointer	x	x
74H - t	set absolute page	x	x
75H - u	soft keyboard lock	x	x
76H - v	memory address pointer	x	x
77H - w	sound audible tone generator	x	x
78H - x	request terminal parameters	x	x
79H - y	invoke confidence test	x	x
7AH - z	blank data stream	x	x
7BH - {	communications mode	x	x
7CH - |	transparent user sequences	x	x
7DH - }		-	-
7EH - ~		-	-
7FH - DEL		-	-

E.1.4 CONTROL SEQUENCES WITH INTERMEDIATES [ESC I ... I F]

F HEX - ASCII	CONTROL SEQUENCE	100	ANSI
40H - @	scroll left	-	x
41H - A	scroll right	-	x
42H - B	graphic size modification	-	-
43H - C	graphic size selection	-	-
44H - D	font selection	-	-
45H - E	thin space selection	-	-
46H - F	justify	-	-
47H - G	spacing increment	-	-
48H - H	quad	-	-
49H - I		-	-
4AH - J		-	-
4BH - K		-	-
4CH - L		-	-
4DH - M		-	-
4EH - N		-	-
4FH - O		-	-
50H - P		-	-
51H - Q		-	-
52H - R		-	-
53H - S		-	-
54H - T		-	-
55H - U		-	-
56H - V		-	-
57H - W		-	-
58H - X		-	-
59H - Y		-	-
5AH - Z		-	-
5BH - [		-	-
5CH - \		-	-
5DH -]		-	-
5EH - ^		-	-
5FH - _		-	-
60H - `		-	-
61H - a		-	-
62H - b		-	-
63H - c		-	-

F HEX - ASCII	CONTROL SEQUENCE	100	ANSI
64H - d		-	-
65H - e		-	-
66H - f		-	-
67H - g		-	-
68H - h		-	-
69H - i		-	-
6AH - j		-	-
6BH - k		-	-
6CH - l		-	-
6DH - m		-	-
6EH - n		-	-
6FH - o		-	-
70H - p		-	-
71H - q		-	-
72H - r		-	-
73H - s		-	-
74H - t		-	-
75H - u		-	-
76H - v		-	-
77H - w		-	-
78H - x		-	-
79H - y		-	-
7AH - z		-	-
7BH - {		-	-
7CH - |		-	-
7DH - }		-	-
7EH - ~		-	-
7FH - DEL		-	-

E.1.5 MODE CHANGING PARAMETERS | ESC [<Ps> ... <Ps> h or l |

<Ps> HEX - ASCII	MODE	100	ANSI
30H - 0		x	-
31H - 1	guarded area transfer mode	x	x
32H - 2	keyboard action mode	x	x
33H - 3	control representation mode	x	x
34H - 4	insertion - replacement mode	x	x
35H - 5	status reporting transfer mode	x	x
36H - 6	erasure mode	x	x
37H - 7	vertical editing mode	x	x
38H - 8		-	-
39H - 9		-	-
3AH - :		-	-
		-	-
		-	-
		-	-
		-	-
		-	-
31H 30H - 10	horizontal editing mode	x	x
31H 31H - 11	positioning unit mode	x	-
31H 32H - 12	send-receive mode	x	x
31H 33H - 13	format effector action mode	x	x
31H 34H - 14	format effector transfer mode	x	x
31H 35H - 15	multiple area transfer mode	x	x
31H 36H - 16	transfer terminator mode	x	x
31H 37H - 17	selected area transfer mode	x	x
31H 38H - 18	tabulation stop mode/form feed mode in 100 mode	x	x

<Ps> HEX - ASCII	MODE	100	ANSI
31H 39H - 19	editing boundary mode print extent	x	x
	in 100 mode	-	-
		-	-
		-	-
		-	-
		-	-
		-	-
		-	-
32H 30H - 20	line feed new line mode	x	-
32H 31H - 21		-	-
	.	.	.
	.	.	.
	.	.	.
		-	-
		-	-
3FH 30H - ?0			
3FH 31H - ?1	cursor keys application mode	x	-
3FH 32H - ?2	reset VT52 mode	x	-
3FH 33H - ?3	132 column mode	x	-
3FH 34H - ?4	smooth scrolling on	x	-
3FH 35H - ?5	white background	x	-
3FH 36H - ?6	relative cursor addresses	x	-
3FH 37H - ?7	wrap around	x	-
3FH 38H - ?8	auto repeat on	x	-
3FH 39H - ?9	interlaced on	x	-
		-	-
		-	-
		-	-
		-	-
		-	-
		-	-
		-	-
3FH 31H 30H - ?10		-	-
3FH 31H 31H - ?11		-	-
3FH 31H 32H - ?12		-	-
3FH 31H 33H - ?13		-	-
3FH 31H 34H - ?14		-	-

<Ps> HEX - ASCII	MODE	100	ANSI
3FH 31H 35H - ?15		-	-
3FH 31H 36H - ?16		-	-
?17	autotab	x	x
3FH 32H 30H - ?20		-	-
3FH 32H 31H - ?21		-	-
3FH 32H 32H - ?22		-	-
?24	form feed to col 0	x	x
?25	form feed to next page	x	x
3FH 32H 36H - ?26	destructive Roll	x	x
3FH 32H 37H - ?27	vertical auto scroll	x	x
3FH 32H 38H - ?28	horizontal auto scroll	x	x
?30	vertical tab to col 0	x	x
?31	insert delete line to col 0	x	x

Only an authorized service representative should remove the equipment cover.

Beehive equipment is in compliance with the following standards:

1. Underwriter's Laboratories (UL)
   UL478 Electronic Data Processing Units and Systems
   UL114 Office Appliances and Business Equipment

2. Canadian Standards Association (CSA)
   CSA 22.2 #154 Data Processing Equipment
   CSA 22.2 #143 Office Equipment

3. International Electronics Commission
   IEC 380 Safety of Electrically Energized Office Machines

4. Federal Communications Commission (FCC)
   FCC Part 15 For Class A Computing Devices