SOROC

IQ135

Operator's guide

Table of contents

General specifications

Character size	5x9 character in a 7x10 field
Display	96—ASCII alphanumeric characters. 24 special control and graphics characters 8 video atttribute codes 24 lines of 80 characters 25th line of status, mode, and user message dual intensity reverse, blink, underline fields combination video fields
CRT	12 in, non-glare, P4 phosphor
Keyboard	55-key main module 8 key cursor control pad 14 key numeric entry pad 8 key edit and clear pad (IQ135) 4 general purpose function keys (IQ135) 14 dual purpose function keys (reprogrammable) 4 key block transmission and print (IQ135) auto repeat after 0.5 sec delay programmable repeat rate
Edit features	clearing to space or null clear page all or foreground only clear to end of page clear to end of line char insert and delete line insert and delete insert entry mode
Cursor motion	incremental—forward, back, up, down home, return, newline cursor addressing & read cursor formatted tab (forward and backward) typewriter tabs (forward and backward)
Word structure	7 or 8 data bits with 1 or 2 stop bits. complete parity selection
Transmission	conversational mode—full or half duplex block mode—transmit by line, page, or message
Interfaces	RS232—main port, auxiliary port, printer port 20 ma current loop—main port
Formatting	set format or typewriter tab all video fields high or low intensity reverse video field underline video field blinking video field reverse-underline field reverse-blinking field blinking-underline field

	reverse-blinking-underline field low intensity field LOW and HIGH intensity markers protected fields adjustable right margin line graphics
Data rates	110, 150, 300, 600, 1200, 1800, 2400 4800, 9600, 19200. separately selectable for main and printer ports
Weight	45 LBS. approx.
Environmental	altitude–sea level to 10,000 feet operational temperature—+5 C to +40 C humidity—5% to 90% non-condensing vibration—10 hz to 55 hz (.01 in peak-to-peak)

Escape functions

Escape functions from the computer	When executing an escape sequence from the remote computer, the computer must transmit the *ASCII ESC* code *(IB hex)* followed by one or more characters identifying the function to be performed by the IQ135. Note that the *ESC* code may be reprogrammed *(intelligent flexibility)* to be any code and the user must take this into account in reading this manual if the *leadin* is reprogrammed by the user.
Escape functions from the keyboard	When executing an escape sequence from the keyboard, the state of the *shifted leadin* switch *(K8-SW6)* is important. If this switch is up *(shifted leadin)*, then performing a shift/*ESC* operation sets the unit up for an internal *ESC* sequence. An unshifted/*ESC* operation transmits an *ESC* code to the line. If the *shifted leadin* switch is down, the functions are reversed. Therefore, this switch should be positioned to accomplish the most frequently used operation in the unshifted mode. Once the unit is set up for an internal *ESC* sequence, it should be immediately followed by the rest of the sequence described in the IQ135 *operations* section.

Video fields

Video attributes	A video attribute code simply specifies what field attributes will become active at that point. Therefore, 'housekeeping' of the screen format is simple since every attribute stands alone without regard to prior video conditions.
Writing video attribute codes to the screen	The method for writing the video attribute codes to the screen is the same as writing control codes to the screen. The user must first transmit an *ESC* code. The next control code that follows is written to the screen at the current cursor location.

Video attributes codes

Field type/control sequence		*ASCII codes*
Rev/Blink/Underline	(Ctrl P)	0001 0000
Blink/Underline	(Ctrl Q)	0001 0001
Reverse/Underline	(Ctrl R)	0001 0010
Reverse/Blink	(Ctrl S)	0001 0011
Underline	(Ctrl U)	0001 0101
Blink	(Ctrl V)	0001 0110
Reverse	(Ctrl W)	0001 0111
All fields off	(Ctrl D)	0000 0100

Protected formats

The IQ135 has the very powerful capability of protecting portions of the screen from data manipulation. This allows the display and use of formats such as forms where large portions of the information remains unchanged. In this case, the unchanging locations are written protected and the variable information unprotected.

Creating protected formats

Protected data on the screen depends on two things. The data must be written in 'low intensity' and *protect* mode must be on (indicator in reverse video). Writing low intensity data to the screen is normally done by setting *write low* mode. Any subsequent characters are then written in low intensity. The transition to high intensity data is then accomplished by resetting *write low* mode. Another method of creating low intensity fields is to first write all of the data in normal high intensity and then invoking the command *low markers*. This command changes the data between two pointers to low intensity. The command *high markers* changes the specified block of data to high intensity. Then when *protect* mode is set, the low intensity data becomes protected.

Movement through the protected form

Once the form is created and *protect* mode is set, the cursor can only reside in the unprotected (high intensity) areas. If a protected field is encountered in the course of incrementally moving the cursor, it is automatically moved out of the field. The direction of movement is determined by the type of cursor operation being performed. Any movement in the forward direction such as *forespace, return, linefeed,* and *newline* skips out of the field forward. *Backspace* and *upline* skips in the backward direction.
When *protect* mode is on, *tab* and *backtab* can be used to move between unprotected fields. Please see the *tabbing modes & functions* section.

Effects on transmission time

An advantage of this is the minimization of transmission time between the computer and the IQ135. The protected form is written to the screen and the operator fills in the smaller unprotected areas. The cursor operations facilitate forward and back movement between unprotected fields. Then the operator initiates a transmission to the computer of unprotected *(foreground)* data only. This type of transmission inserts special codes into the unprotected data stream to indicate to the computer that a protected field has been skipped or an end of line has been detected. The computer then merely issues a 'clear' command that erases only unprotected *(foreground)* data at which time the form is again ready for new entries. This operation avoids transmission from the terminal of the protected data (written by the computer anyway) and the rewriting of the protected form.

Intelligent flexibility

This will describe the capabilities of the IQ135 *intelligent flexibility.* These allow operational characteristics of the terminal to change to accommodate non standard usage. The table of variable parameters is shown below.

Programming—general

The flexibility is programmed by sending a multiple character sequence consisting of the following:

⟨ESC⟩ • ⟨nn⟩⟨mm⟩⟨SO⟩ or ⟨ESC⟩ • ⟨nn⟩⟨c⟩⟨SO⟩

⟨ESC⟩	=	ASCII escape code
•	=	ASCII period
⟨nn⟩	=	item no. (Flexibility table) (2 digit decimal number)
⟨mm⟩	=	new value (2 digit decimal number)
⟨c⟩	=	ASCII character used in place of ⟨mm⟩
⟨SC⟩	=	ASCII SO code (terminates loading if from line)

If a NULL code is programmed as a transmission delimiter (computer or printer) then it is ignored and nothing is sent.

Programming from the computer

When the sequence is executed from the line, loading of a contiguous section of the table is allowed with one *ESC* sequence. The ⟨nn⟩ in the sequence points at the first item number to change and the new values for it and the following sequential items are strung before the terminating SO code.

Programming from the keyboard

Characteristics may also be changed from the keyboard with two variations from the sequence from the line. First the ⟨SC⟩ code on the end of the sequence is not required. Second the multiple loading feature does not apply to the keyboard programming and thus items may be changed only by individual escape sequences.
For example: ⟨ESC⟩ • 00& would set to leadin character to &

Flexibility table

Notes: ⟨ ⟩ indicate default values
NULLS are not sent to the computer or printer

00. **Leadin ⟨ESC⟩**
Indicates which ASCII character will be recognized as the *leadin* (action) code from the line.

01. **End block 1 ⟨CR⟩**
02. **End block 2 ⟨NULL⟩**
Appended to the text to indicate the end of a *send line/page/message* operation.

03. **End line 1** ⟨**US**⟩
04. **End line 2** ⟨**NULL**⟩
Inserted in the text to indicate 'end of line' during a *send page/msg* operation.

05. **Protect field skip 1** ⟨**FS**⟩
06. **Protect field skip 2** ⟨**NULL**⟩
Inserted in the text when a protected field is skipped during a *send foreground* operation. It is transmitted when the end of the protected field is detected.

07. **Protected field start 1** ⟨**ESC**⟩
08. **Protected field start 2** ⟨**(**⟩
Inserted into the text when the start of a protected field is encountered during a *send all* operation.

09. **Protected field end 1** ⟨**ESC**⟩
10. **Protected field end 2** ⟨**(**⟩
Inserted into the text when the end of a protected field is encountered during a *send all* operation.

11. **Start of message marker** ⟨**STX**⟩
Determines which *ASCII* code is to be detected as the start of message code for *send message* operations.

12. **End of message marker** ⟨**ETX**⟩
Determines which *ASCII* is to be detected as the end of message code for *send message* operations.

13. **Pad character** ⟨**NULL**⟩
Determines the codes that will be transmitted between the text and the *end block* (see item #1) delimiter if *block mode pad* is enabled.

14. **Print delimiter 1** ⟨**CR**⟩
15. **Print delimiter 2** ⟨**LF**⟩
Inserted into the print message before the text, at the end of every line, and after the end of the text for every *formatted print operation*.

16. **Number of nulls** ⟨**01**⟩
Determines the number of null codes sent to the printer after items 14 and 15 are transmitted.

17. **Printer delay internal time** ⟨**30**⟩
Determines the delay time between the transmission of items 14 and 15 and the next text character if *printer delay internal* is enabled. The value to be stored is calculated by dividig the desired time by 16.667 msec. (500 msec/16.667 msec = 29.99 or 30).

18. **Auxilliary port print** ⟨**NULL**⟩
Determines whether the printer output will go to the printer port or the auxiliary port. (LSB= 1 then *aux* port, LSB= 0 then *printer* port).

19. **Right hand margin** ⟨**79**⟩
Determines the usable line length. Legal values are between 00 and 79.

20. **Blink rate 'on' time** ⟨**45**⟩
21. **Blink rate 'off' time** ⟨**10**⟩
Determines the time that a blink field will be 'on' (visible) and 'off'. See item 17 for calculations. (750 msec on/167 msec off)

22. **Keyboard repeat rate** ⟨03⟩
Determines the repeat rate of the keyboard. Calculated by dividing 60 by the stored value (60/3= 20 chars/sec).

23. **Break time** ⟨16⟩
Determines the duration of the *break* signal to the line. See item 17 for calculations. (260 msec/16.667 msec= 15.6 or 16)

24. **Request to send delay** ⟨17⟩
Determines the time *RTS* is held up after the end of the transmission on the *send* operations. See item 17 for calculations. The initial value is about 266 milliseconds.

25. **Reserved for future use**

26. **Enable cursor** ⟨01⟩
Enables or disables cursor display. Cursor off is 00, cursor displayed is 01.

27. **Program switches**
This is where the program switches for the IQ135 are stored upon power up or whenever a switch value changes. By loading this byte the user may set the up the switch configuration from the remote computer or from the keyboard. Each of the low six bits is a program switch and all switches must be loaded at the same time.

bit 0 is *print all*
bit 1 is *suppress echoes ESC*
bit 2 is *block mode pad enable*
bit 3 is *switch 135*
bit 4 is *conversational FDX*
bit 5 is *shifted leadin*
bit 6 is *unused*
bit 7 is *unused*

Function keys — (general purpose)

The IQ135 has 14 special functions that cause data to be sent to the remote computer upon activation. These functions are activated by depressing the *control* key along with one of the keys on the 14 key numeric pad consisting of the numbers 0-9, period, return, comma, and minus. Each of these will transmit a three character sequence to the computer consisting of the *SOH* code *(control A)*, than a character from (thru M, followed by a
CR code. The codes generated by each key are listed over. The IQ135 has four dedicated function keys labeled f1 thru f4 whose messages duplicate *control 1* thru *control 4*. Since these keys require the depression of only one key, they should be assigned to perform the most frequently used functions.

Key (s) depressed	*Codes transmitted*
Control/NP1 or f1	⟨SOH⟩ @ ⟨CR⟩
Control/NP2 or f2	⟨SOH⟩A⟨CR⟩
Control/NP3 or f3	⟨SOH⟩B⟨CR⟩
Control/NP4 or f4	⟨SOH⟩C⟨CR⟩
Control/NP5	⟨SOH⟩D⟨CR⟩
Control/NP6	⟨SOH⟩E⟨CR⟩
Control/NP7	⟨SOH⟩F⟨CR⟩
Control/NP8	⟨SOH⟩G⟨CR⟩
Control/NP9	⟨SOH⟩H⟨CR⟩
Control/NP0	⟨SOH⟩I⟨CR⟩
Control/NP•	⟨SOH⟩J⟨CR⟩
Control/NP RTN	⟨SOH⟩K⟨CR⟩
Control/NP,	⟨SOH⟩L⟨CR⟩
Control/NP-	⟨SOH⟩M⟨CR⟩

Note: 'NP' means numeric pad

Programmable key display

To display the key messages, the user (or remote computer) issues ⟨ESC⟩/ which causes the unit to display the 14 *function* key messages on the first 14 lines of the display. The rest of the lines are filled with protected '-' characters.

Function key programming

To reprogram these *function* key messages the operator (or computer) simply puts the desired message for the first *function* key *(CTRL/NP1)* on the first line of the display, the message for the second key *(CTRL/NP2)* on the second line, etc. When all the messages are as desired, the operator (or computer) then issues an ⟨ESC⟩! to reprogram the *function* keys. The message programmed starts at the leftmost column and extends to the last non-null or non-space character on the line. Note that, upon activation, the terminal only transmits to the remote computer the codes that are programmed. No internal operations are performed unless the computer echoes the codes from the *function* keys. The maximum length message for any individual key is 80 characters. The total length for all 14 messages cannot exceed 128 bytes.

A useful hint when reprogramming from the keyboard is to display the message (see above), making the alteration, and then reprogramming the keys. This is particularly useful when a slight error (typo, etc.) has been made. Remember, all editing functions are active!

Graphics mode

The IQ135 *graphics* option uses eleven selected control codes to create a line graphic set. These codes are shown on the *control & action* code chart in the *quick reference* section at the back of this manual.

This mode, set with ⟨ESC⟩-4 and cleared with ⟨ESC⟩-5, is used to allow the writing of these control codes to the screen without preceeding them with an *ESC* code.

These eleven characters allow the terminal to outline, with straight lines, the different sections of a form.

Block transmit operations

Selecting the data block to transmit

Send page

In this type of send, the data block starts at the *home* position of the page and extends up to and including the cursor location. This operation is available in *send page all* and *send page foreground*.

Send line

The data block starts in the leftmost column of the current cursor line and extends up to and including the cursor location. Available in *send line all* and *send line foreground*.

Send message

In this type of operation, the terminal starts a search backward starting one position before the cursor location for a *start of message marker* (normally an *ASCII STX*). When one is found, the character immediately to the right is the start of the data block. If not found, the *home* position becomes the start of the data block. After determining the starting position, a search forward is started looking for an *end of message marker* (normally an *ASCII ETX*) or the end of the page. If one is found, the character immediately preceeding it is considered the end of the transmission block. Available in *send message all* and *send message foreground*.

Selecting what data is transmitted

Send all

This type of send transmits all data on the screen whether protected or unprotected. When a protected field is encountered, the *protected field start* codes are transmitted to inform the computer that a protected field is starting. The characters in the protected field are then sent followed immediately by the *protected field end* codes which indicate that unprotected characters follow. When the end of a line is detected, the *end line* codes are inserted into the data stream. *End block* codes are sent when the entire message has been completed. All of the codes indicated above are reprogrammable (see *intelligent flexibility* section). This type of send is available in *send page all*, *send line all*, and *send message all*.

Send foreground

Only unprotected data is transmitted with a *send foreground* type operation. When the end of a protected field is detected, the *protect field skip* codes are inserted into the data stream to indicate that a protected field has been skipped. The *end line* codes are again transmitted when an end of line is encountered and the message is terminated with the *end block* codes. All transmission delimiters are reprogrammable (see *intelligent flexibility* section).
This type of send is available in *send page foreground*, *send line foreground* and *send message foreground*.

Other transmission features

Pad character

This feature is used to allow proper block mode transmissions on full duplex systems that normally echo data. If all echoed data was processed normally once the block send was completed, the screen data would be destroyed. To prevent this from happening, three pad characters (normally *NULL* codes) are inserted into the data stream

after the last character of the text and the *end block* codes. This allows time for the last echoed character to reach the IQ135. When the third pad character has cleared the transmitter, the IQ135 initializes its fifo to get rid of the echoed data.

Print operations

Print formatted

In this method of printing, the cursor is placed one position beyond the last character to be printed. When the *print* key is depressed or an ⟨ESC⟩P sequence is received, an *EM* (ctrl/Y) code is written at the cursor location, Then all data from the home position to the *EM* code (not included) is sent to the printer. *Print delimiters* are sent before the text, at the end of every line, and after the text. These *print delimiters* can be reprogrammed. When the operation is complete, the *EM* code is overwritten with a space code. Any control codes embedded on the screen, such as video attributes, are changed to space codes before sending it to the printer if operating in 135 mode. If in 120 mode, the control codes are sent to the printer unchanged. The unit normally prints only unprotected data unless the *print all* switch is active (up) in which case all characters, protected and unprotected, are printed.

Print unformatted

This method of printing is activated with a shifted *print* key or with an ⟨ESC⟩p sequence. This operation starts at the cursor location and ends at the next *EM* (ctrl/Y) code or the end of the page without changing or inserting any codes in the data stream. All printer control codes (such as *CR, LE,* etc.) must be written on the screen. When the *EM* code is reached, the printing stops but the *EM* code is not overwritten.

End of line control

The IQ135 has three methods of controlling the transmission of data to the printer. The first is called *printer busy* control (see front panel switch *K12-SW4*). If this switch is up, then the printer controls the transmission be raising *data terminal ready* (pin 20) when it wants to stop the transmission and lowering it to start. The second method is *printer ready* (see front panel switch *K12-SW3*) which functions like *printer busy* except that the sense of *data terminal ready* is reversed (high to transmit and low to stop). Only one of these switches should be up!
When both of these switches are down, *printer delay internal* is selected. This type of control ignores signals from the printer and inserts a delay after every line is transmitted to give the printer time to accomplish the *CR/LF* operations. This delay, normally 500 msec, can be reprogrammed in *flexibility.*

Tabbing modes and functions

Formatted tab

When the *tab* or *backtab* function is initiated with *protect* mode active, the tabbing function is done between unprotected fields. This means that a *tab* would move the cursor to the first position of the next unprotected field. A *backtab* would move the cursor to the first position of the current field or, if already there, to the first position of the previous unprotected field. One method of creating forms with column-type data is to use the *protected tab set* function.

Typewriter tabs

The *typewriter tabs* are much more flexible than *formatted tabs* because no character position is used to mark the stops and *protect* mode is not set which allows scrolling and the use of any of the editing functions. Remember that when *protect* mode is active, the typewriter tab function is disabled.

Keyboard

To set a *typewriter* tab stop, simply position the cursor in the proper column and depress the *tab* key while holding down the *control* key. If the *tab* key is depressed while holding down both the *control* and *shift* keys then a tab stop programmed at the current cursor column will be cleared. Using the *tab* key then moves from stop to stop in the forward direction. The *backtab* key (*tab* key shifted) does the same function in the backward direction. In either direction, the leftmost column of every line is considered a typewriter tab stop.

Computer

To set typewriter tab stops from the computer, send the sequence
⟨ESC⟩ l ⟨x1⟩ ⟨x2⟩ ⟨x3⟩ . . . ⟨xn⟩ ⟨RUB⟩
When the 'l' is received, all current tab stops are cleared. The x1, x2, x3, xn are the coordinate positions where the new tab stops are desired. These coordinates are the same as those of the *load cursor* and can be found in *table 1*. The *ASCII rub* code (7F hex) indicates the end of the the operation.
All typewriter tab stops are cleared when an ⟨ESC⟩ sequence is issued. Therefore, when any change is to be made to the tab stops, all of the positions must be rewritten.

Switches and controls

Back panel

On/off switch

This two position switch controls AC power to the terminal. Setting the switch to the *on* position will reset the circuitry, position the cursor to home and clear the display to unprotected nulls, and reset all modes.

Brightness control

This potentiometer controls the brightness of the *CRT display.* The brightness is usually adjusted so that the raster (background) is just below the level of visibility.

Contrast control

This potentiometer controls the character brightness relative to the background. Contrast adjustment is usually made just after brightness adjustment.

Baud rate switch
—Main communication port

This rotary switch selects one of ten baud rates for *main port* transmission. The following is a list of the switch positions and the associated baud rates.
(switch position = baud rate)

0=110	1=150	2=300	3=600	4=1200
5=1800	6=2400	7=4800	8=9600	9=19200

Interface select switch
—Main I/O port

This three position slide switch is located to the far right as viewed from the back of the unit. When the switch is in the RS position, then the *RS232* interface is chosen. *CL* selects the current loop interface. The center position (not labeled) is a local loopback used mainly for testing.

Transmission mode select switch

This three position slide switch, located just inside of the *interface select switch*, is used to select the transmission mode for the main I/O port. The position labeled 'F' selects *conversation full duplex*, 'B' selects *block mode*, and 'H' selects *conversation half duplex*. This switch selects the mode that is used on initialization and the computer (or operator) can still change modes with *block mode set* and *conversational mode set*. However, a change on this switch always overrides any software initiated mode selection.

Main port connector

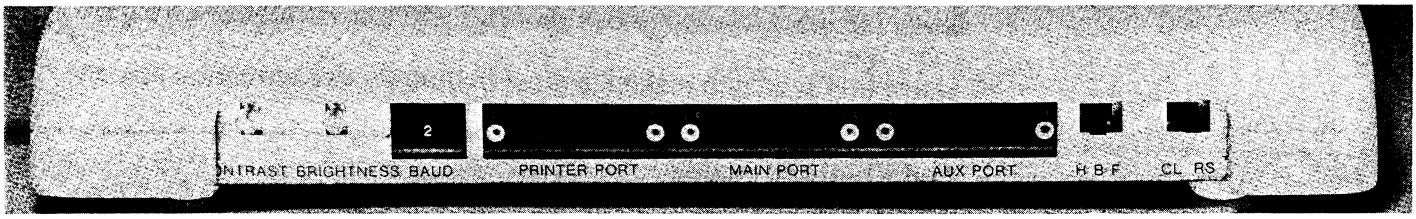
The *main communications port* is the center female *DB25 connector* as viewed from the rear of the unit.
The pinouts for the *main port* are listed below:

Pin no	*Signal name*
1	Frame ground
2	Transmit data (to the computer)
3	Receive data (from the computer)
4	Request to send
5	Clear to send
6	Data set ready
7	Signal ground
8	Carrier detect
9	20 milliamp source
10	Current loop *Receive*
11	Current loop *Receive*
12	Current loop *Transmit*
13	Current loop *Transmit*
20	Data terminal ready

Aux port connector

The *aux port* is the rightmost female *DB25 connector* as viewed from the rear of the unit. This port is essentially a pass thru from the main communications port and can be used to connect to any *RS232 serial device* to the computer. The data path from the *main port* to the *aux port* is controllable and is enabled on initialization. It may be turned on or off under software control (see *auxiliary port on/off*). If the *main port* is running a current loop interface, the IQ 135 converts the data lines in both directions.
The pinouts for the *aux port* are as follows.
Note: All signals are referenced to the device attached to the auxiliary port. A terminal would use a straight-thru cable.

Pin	*Signal*
1	Frame ground
2	Transmit data
3	Receive data
4	Request to send
5	Clear to send
6	Data set ready
7	Signal ground
8	Carrier detect
20	Data terminal ready



Printer port connector

The *printer port* is the leftmost female *DB25 connector* as viewed from the rear of the unit. This port allows the IQ 135 to interface to most serial printers. See the section on *print operations* for the many controls and capabilities. The *printer port* baud rate is selected separately from the main port. The pinouts for the *printer port* follow.

Pin	*Signal*
1	Ground
3	*Serial data*
6	IQ 135 ready
7	Gnd
8	IQ 135 ready
20	*Printer busy/ready*

Front panel switches

Switch bank K8

Switch 1

Print all
If up during a *print formatted* operation, the terminal will send both unprotected and protected data. If down, only unprotected data will be printed.

Switch 2

Suppress echoed ESC mode
If up, causes the terminal to ignore an *ESC* code transmitted by the computer immediately after the user has transmitted an *ESC* code. This is to prevent an echoed *ESC* code from locking up the IQ 135. If down, this mode is disabled.

Switch 3

Block mode pad enable
If up, enables the user to run in *block* mode on full duplex systems that normally echo data. Three *pad characters* (Intelligent flexibility), are sent between the last data character and the *end of block* code(s) as a delay to insure that the system has echoed everything. After sending these pad characters, the input buffer is cleared to erase the echoed data. The *end of block* code is then sent. The pad character should be an *ASCII* code that will be ignored by the computer (such as *null*).
If down, disables the *block mode pad* feature.

Switch 4

135 Mode
If this switch is down, the terminal operates in IQ 120 emulation mode.

Switch 5

Full/half duplex select
If up, causes *full duplex* to be selected when entering conversation mode. If down, *half duplex* is selected.

Switch 6

Shifted leadin—keyboard
If up, the *leadin* (action) code will be generated on the keyboard by depressing the *ESC* key while holding down the *shift* key. An unshifted *ESC* key will transmit the *ASCII ESC* code to the computer.
If down, the functions are reversed. The *ESC* key alone will be the

	leadin (action) code and the *shifted/ESC* will transmit the *ESC* to the computer. Select this switch to make the most frequently used function activate when unshifted.
Switch 7	**Audible alarm enabled** If up, the audible alarm will sound upon receipt of a *ctrl/G*. If down, the alarm is disabled.
Switch 8	**Break key disabled** If up, the activation of the *break* key will not 'space' the main transmit data line. If down, normal *break* operation.
	Switch bank K10
Switch 1	**50 Hz refresh** If up, 50 hz refresh selected. 60 hz otherwise.
Switch 2	**Line 25 display enabled** Line 25 (indicators) is displayed when this switch is up. Blanked if down.
Switch 3	**Request to send 'on'** If up, *RTS* on the main port is held 'spacing' (high voltage). If down, *RTS* is controlled by the IQ 13x depending on the transmission mode.
Switch 4	**Main bit 8 one** Active only if *SW5* is up which enables transmission of an 8 bit data word. Normally used with parity disabled to hold the parity bit position constant. If up, a 'one' bit is sent in the parity bit position of the word. If the switch is down, a 'zero' bit is inserted.
Switch 5	**Main eight bit data** Normally used with parity disabled to hold the parity bit position constant. If up, *SW4* determines the sense of the parity bit position in the word transmitted out the main I/O port. If down, normal 7 bit data is sent.
Switch 6	**Main even parity** Active only if *SW7* is down. If up, an *even* parity bit is added to each character transmitted out the main *I/O* port. *Odd* parity if down.
Switch 7	**Main no parity** If up, no parity bit is added to characters transmitted out the main *I/O* port. If down, *SW6* determines the type of parity to be used.
Switch 8	**Main port two stop bits** If up, two stop bits are appended to each character transmitted out the main *I/O* port. One stop bit if down.
	Switch bank K12
Switch 1	**Non-blinking cursor** If up, non-blinking cursor. Blinking if down.
Switch 2	**Block cursor** If up, reverse video, block cursor. Underline if down.

Switch 3	**Printer ready control** Used if the printer is to control data transmission from the IQ 135. If up, transmission to the printer will occur only if *PIN 20* of the *printer* port is 'spacing' (high voltage). No effect if switch is down. If this mode is selected, *SW4 (printer busy)* must be down.
Switch 4	**Printer busy control** Used if the printer is to control data transmission from the IQ 135. If up, transmission to the printer will occur only if *PIN 20* of the *printer* port is 'marking' (low voltage). No effect if switch is down. If this mode is selected, *SW3 (printer ready)* must be down.
Switch 5	**Power on auxiliary port print** If up, selects the *AUX* port as the printer output port. *Printer* port otherwise. Read only on initialization.
Switch 6	**Printer even parity** Active only if *SW7* is down. Selects *even* parity on the printer port if up. *Odd* parity if down.
Switch 7	**Printer no parity** If up, no parity bit is added to the printer output byte. If down, then *SW6* is followed to add the parity bit.
Switch 8	**Printer two stop bits** If up, two stop bits are added to the printer output byte. One stop bit if down.

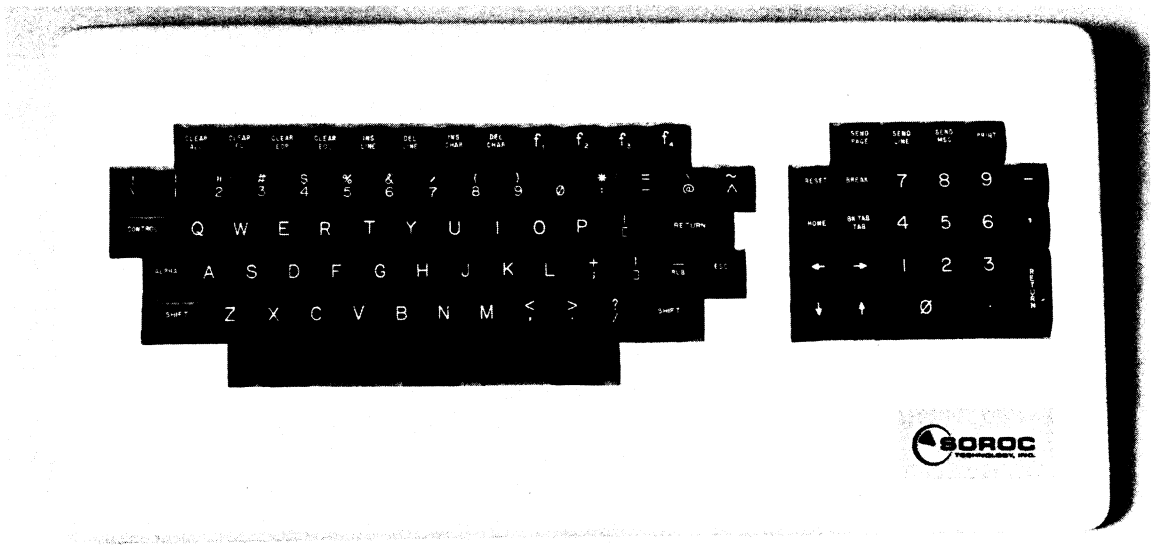
Turn on procedure

1.	Ensure the IQ 135 is plugged into a grounded outlet of the proper voltage.
2.	Set the *on/off* switch on the IQ 135 to the *on* position.
3.	Wait approximately 20 seconds for the *CRT* to warm up. The cursor should then appear at the home position with the rest of the screen clear. *Note: Turning the power switch on clears the display memory. Therefore, if the display contains information which should be saved, the information should be transmitted to the remote computer or otherwise saved before turning the terminal Off.*
4.	If the cursor does not appear after the warm-up period, hit the *reset* key while depressing *shift*. a) Set the contrast control to the middle of it's range. b) Turn the brightness control clockwise until the screen is bright, then reduce brightness slowly until the background is barely visible. c) Adjust brightness and contrast for desired presentation. d) If the cursor does not appear, contact an authorized service representative.

IQ135 operations

Answerback	⟨**ESC**⟩ **-0** Activates the transmission of the canned *answerback* message.
Auxiliary port off	⟨**ESC**⟩ **space of** ⟨**ESC**⟩ **A** Stops the data sent from the computer to the main port from reaching the device on the *auxiliary I/O* port. The computer can still receive data from auxiliary I/O device.
Auxiliary port on	⟨**ESC**⟩ This command allows data from the computer to reach the device on the *auxiliary I/O* port. The computer can then transmit as well as receive data from the auxiliary *I/O* device.
Auxiliary port print	(See *Flexibility* table item 18)
Backspace	⟨**BS**⟩ **or ← key** Moves the cursor left one position. Movement off the first column places the cursor on the last position of the line above. Movement off the *home* position causes the cursor to be positioned on the last line and column of the page. Protected characters are skipped.
Backtab	⟨**ESC**⟩ **I or shift** ***tab/bk tab*** **key** *Protect mode* — causes the cursor to skip backwards to the first position of the current unprotected field or, if already in the first position, to the previous unprotected field (the last position of the page is a legal stopping point). *Not protect mode* — moves the cursor to the last position of the page if no *typewriter* tab stops exist. If tab stops are set, the cursor will move backwards to the previous stop.
Blink rate on/off times	(See *Flexibility* table items 20,21)
Block mode set	⟨**ESC**⟩ **B** Sets the terminal into *block* mode. In *block* mode any characters typed on the keyboard are simply written to the display. Nothing is transmitted unless one of the block send operations is utilized such as *send line, send page,* or *send message.*
Break time	(See *Flexibility* table item 23)
Clear all nulls	⟨**ESC**⟩ *** or shift/*clear all* key** Clears the entire page (including protected data) to unprotected null codes. *Protect* and *write low* modes are cleared and the cursor is moved to the *home* position.
Clear all spaces	⟨**ESC**⟩ **+ or *clear all* key** identical to *clear all nulls* described above except that unprotected space codes are written to the memory.
Clear EOL nulls	⟨**ESC**⟩ **t or shift *clear EOL* key** clears all unprotected data from the cursor location to the end of the line to null codes. A protected field encountered before the end of the line terminates the operation. If *write low* mode is on then the null codes are written as low intensity and will be protected if *protect* mode is on. In this case, the cursor will skip out of the newly created protected field.

Clear EOL spaces	**⟨ESC⟩ T or *clear EOL* key** identical to *clear EOL nulls* described above except that space codes are written to memory.
Clear EOP nulls	**⟨ESC⟩ y or shift/*clear EOP* key** clears all unprotected data from the cursor location to the end of the screen to unprotected null codes. The cursor location is not modified.
Clear EOP spaces	**⟨ESC⟩ Y or *clear EOP* key** identical to *clear EOP nulls* described above except that space codes are written to memory.
Clear foreground nulls	**⟨ESC⟩: or shift/*clear FG* key** clears all the unprotected (foreground) data on the page to unprotected null codes. The cursor is positioned to the first unprotected location on the page.
Clear foreground spaces	**⟨ESC⟩; or *clear FG* key** identical to *clear foreground nulls* described above except that unprotected space codes are written to the memory.
Conversational mode set	**⟨ESC⟩ C** Sets the terminal into *full duplex (FDX)* if switch 6 on switch bank *K20* is up or *half duplex (HDX)* if it is down. Characters typed are sent to the computer in either mode but are only written to the screen in *HDX*.
Cursor enable	(See *flexibility* table item 26)
Delete char nulls	**⟨ESC⟩ w or shift *del char* key** Deletes the character under the cursor and moves all data following one position to the left writing a null code at the end of the line A protected field encountered will always terminate the operation and the null code is written in the last unprotected position before the protected field.
Delete char spaces	**⟨ESC⟩ W or *del char* key** Identical to *delete char nulls* described above except that a space code is used as the edit character.
Delete line nulls	**⟨ESC⟩ r or shift *del line* key** Causes all lines starting from the cursor line to the end of the page to be shifted up one line. The line the cursor is on is lost. The bottom line of the page is cleared to unprotected nulls. The cursor is moved to the first column of the same line. *Protect* mode disables this operation.



Delete line spaces	**⟨ESC⟩ R or *del line* key** Identical to *delete line nulls* described above except that the bottom line is cleared to spaces.
End of block ***transmission delimiter***	(See *flexibility* table items 01,02)
End of line ***transmission delimiter***	(See *flexibility* table items 03,04)
END message marker	**⟨ESC⟩ d** Used with *start message marker.* (See *flexibility* item 12)
Forespace	**⟨FF⟩ or → key** Moves the cursor right one position. Movement off the last column performs a *newline* operation. Protected characters are skipped.
Function keys initialization	**⟨ESC⟩, 0** Causes the function keys to be reset to the power-on state. No other modes or operating characteristics are modified.
Graphics mode off	**⟨ESC⟩ -5** Turns off the mode that allows the writing of the 11 line graphics control codes.
Graphics mode on	**⟨ESC⟩ -4** Turns on the mode that allows the writing of the 11 line graphics control codes to the screen without preceeding them with *ESC* codes.
High markers	**⟨ESC⟩ -2 ⟨y1⟩ ⟨x1⟩ ⟨y2⟩ ⟨x2⟩** This command will cause all characters delimited by the positions specified by the "y" and "x" coordinates to become high intensity. The (y1,x1) specifies the starting page coordinate, and the (y2,x2) specifies the ending page coordinate. Both of these coordinates are two *ASCII* characters each thus the total command length is seven (7) characters. The coordinates are the same as for the *load cursor* command and the values they may have are listed in *table 1.*
Home	**⟨RS⟩ or *home* key** Causes the cursor to be positioned to the first line and column of the current page. Movement out of a protected field is forward.
Insert character nulls	**⟨ESC⟩ q or shift *ins char* key** Inserts a null code into the text at the cursor position and moves all data following to the right one position. A protected field always terminates the operation and the last unprotected byte is lost.
Insert character spaces	**⟨ESC⟩ Q or *ins char* key** Identical to *insert character spaces* described above except that a space code is inserted into the text at the cursor position.
Insert line nulls	**⟨ESC⟩ e or shift/*ins line* key** Identical to *insert line spaces* described below except that the new line consists of unprotected null codes. When *AMD-2* mode is enabled the escape sequence (⟨ESC⟩ e) for *insert line nulls* becomes the escape sequence for *status two on*. This is for full *ADM-2* compatibility.
Insert line spaces	**⟨ESC⟩ F or *ins line* key** Inserts an entire line of unprotected space codes at the current cursor line. All data at and below the current cursor line is shifted down and the last line of the page is lost. The cursor is moved to the start of the cleared line. *Protect* mode disables this command.

Insert mode off	⟨**ESC**⟩ **8** Disables *insert mode*. Sets to the standard mode of causing any characters received from the computer or keyboard to overwrite any character in the cursor location.
Insert mode on	⟨**ESC**⟩ **9** Enables *insert mode*. All characters received from the computer or keyboard are inserted into the text. This is accomplished by performing an *insert character* operation (described above) before each character is written to the screen.
Intelligent flexibility initialization	⟨**ESC**⟩ **, 1** Causes the *intelligent flexibility* to be reset to the power-on state. No other non-related modes are modified.
Keyboard lock	⟨**ESC**⟩ **#** Causes the keyboard to be locked. This prevents the user from using any key on the keyboard except the *reset* key which will reset the *keyboard lock* mode.
Keyboard repeat rate	(See *flexibility* table item 22)
Keyboard unlock	⟨**ESC**⟩ **' or *reset* key** Resets the *keyboard lock* function allowing operator input.
Leadin code	(See *flexibility* table, item #00)
Linefeed	⟨**LF**⟩ **or ↓ key** Causes the cursor to move down one line while remaining in the same column. Movement out of a protected field is forward. When moving off the last line of the page, the operation depends on the condition of *protect* mode. If *protect* mode is on, the cursor moves to the top line of the page. If *protect* mode is off, the page is scrolled and the top line is lost.
Load cursor	⟨**ESC**⟩ = ⟨**y**⟩ ⟨**x**⟩ This command gives the user the capability of moving the cursor to any line and column of the page by its address. The address consists of two *ASCII* characters describing the line and column. The format of this command is: ⟨ESC⟩ = ⟨y⟩ ⟨x⟩ where ⟨y⟩ is the character describing the line number, and ⟨x⟩ is the character describing the column number. For information of the values of ⟨y⟩ and ⟨x⟩ refer to *table 1* in the back of this manual.
Low markers	⟨**ESC**⟩ **-1** ⟨**y1**⟩ ⟨**x1**⟩ ⟨**y2**⟩ ⟨**x2**⟩ This command will cause all characters delimited by the positions specified by the "y" and "x" coordinates to become low intensity. The (y1,x1) specifies the starting page coordinate, and the (y2,x2) specifies the ending page coordinate. Both of these coordinates are two *ASCII* characters each thus the total command length is seven (7) characters. The coordinates are the same as for the *load cursor* command and the values they may have are listed in *table 1*.
Multiple character write	⟨**ESC**⟩ **-3** ⟨**m**⟩ ⟨**n**⟩ Allows the user to write from 1 to 96 characters repetitively on the screen. This command has two parameters where ⟨m⟩ is the number of times to write the character ⟨n⟩. Both of these parameters are *ASCII* characters. For example referring to *table 1* in the back of this manual we see that *ASCII* character * has a Y or X value of 11. Thus the command ⟨ESC⟩ -3 * A would cause eleven (11) A's to be written to the screen.

Newline	**⟨US⟩ or *new line* key** This command has the same effect as sequentially performing a *return* and *linefeed* with one exception. The test to see if the cursor is residing in a protected field is done after both operations have been completed. The movement out of a protected field is forward.
Number of nulls	(See *flexibility* table item 16)
Pad character	(See *flexibility* table item 13)
Print delimiters	(See *flexibility* table items 14,15)
Print formatted	**⟨ESC⟩ P or *print* key** Causes an *EM* code (ctrl/Y) to be written to the screen at the cursor location. The cursor is then moved to the 'home' position. All data from the beginning of each line to the last non-space character is transmitted to the printer with the exception of the last line on which all data is printed. The *EM* code is then overwritten with a space code. *Print delimiters* are transmitted to the printer at the start of the operation, at the end of every line, and the end of the print message.
Print unformatted	**⟨ESC⟩ p or shift *print* key** Prints all data from the cursor location to an *EM* code (or end of page) without any additional codes inserted in the data stream. When a stop code is reached, the printing stops but the stop code is not overwritten. The *unformatted print* allows greater flexibility in preparing copy for printers. Some examples of this is line length greater than 80 characters and embedded control codes for special print operations.
Printer delay internal	(See *flexibility* table item 17)
Protect field end ***transmission delimiter***	(See *flexibility* table items 9,10)
Protect field skip ***transmission delimiter***	(See *flexibility* table items 5,6)
Protect field start ***transmission delimiter***	(See *flexibility* table items 7,8)
Protect mode off	**⟨ESC⟩ /** Causes the *protect* mode to be turned off. Low intensity data not protected.
Protect mode on	**⟨ESC⟩ &** Causes the *protect* mode to be set. In this mode, certain operational characteristics are changed. The major difference is that low intensity data becomes protected. This means that the cursor may not be positioned into low intensity fields (now called protected fields). Many commands operate differently or, as in *insert line* and *delete line*, not at all.
Protected tab set	**⟨ESC⟩ V** Causes a column of protected spaces to be written starting at the cursor location and all lines below it. If any line has a protected character at that column then that line is left unchanged and the process continues on the next line.
Read cursor coordinates	**⟨ESC⟩ ?** This command gives the remote computer the capability of reading the current cursor location. The IQ 135 will transmit first *line*, then *column* (as per *table 1*), followed by an *ASCII CR* code.

Request to send delay	(See *flexibility* table item 24)
Reset, hard	**shifted *reset* key** Initilizes the terminal completely. Clears the screen memory, resets all modes, and initializes the programmable keys and flexibility features. Typewriter tabs are cleared.
Reset, soft	***reset* key** Teminates any *block* or *print* transmission currently in progress. No modes are affected except *keyboard lock* which will be reset.
Return	**⟨CR⟩ or *return* key** Receipt of the *CR* code causes the cursor to move to the first column of the current cursor line. The *return* key generates a *CR* code when in *conversation* mode and a *newline* code when in *block* mode. Movement out of a protected field is forward.
Right hand margin	(See *flexibility* table item 19)
Send line all	**⟨ESC⟩ 6 or shift *send line* key** Causes all data from the beginning of the current line to the current cursor position to be transmitted to the computer followed by a *CR* code. Protected fields are bracketed by *ESC)* and *ESC (*.
Send line foreground	**⟨ESC⟩ 4 or *send line* key** Causes all unprotected data from the beginning of the current line to the current cursor position followed by a *CR* code. The end of each protected field skipped is indicated by an *FS* code.
Send message all	**⟨ESC⟩ s or shift *send msg* key** Identical to the *send message foreground* described below with regard to the search for the *start msg marker* and *end msg marker*. However, all characters (protected and unprotected) are sent to the computer. Protected fields are bracketed by *ESC)* and *ESC (*. *US* codes are sent at the end of each line. A *CR* code terminates the message.
Send message foreground	**⟨ESC⟩ S or *send msg* key** Initiates a backward search, starting at the current cursor position, for a *start msg marker* (see below). All unprotected characters from that point to the first *end msg marker* (see below) are transmitted to the computer. The end of each protected field skipped is indicated by an *FS* code and each end of line by a *US* code. A *CR* code is sent at the end of the message. If no *start msg marker* is found during the search, the block of data starts at the home position of the page. If no *end msg marker* is found, the block continues to the end of the page.
Send one character	**⟨ESC⟩ 1** Sends the character under the cursor to the computer. The curser is forespaced legally.
Send page all	**⟨ESC⟩ 7 or shift *send page* key** Causes all data between the home position and the current cursor position to be sent to the computer. Protected fields are bracketed *by ESC)* and *ESC (*. Line ends are indicated by a *US* code. The transmission is terminated with a *CR* code.
Send page foreground	**⟨ESC⟩ 5 or *send page* key** Causes all unprotected characters from the home position to the current cursor position to be sent to the computer. The end of each protected field skipped is marked by an *FS* code and the end of each line is indicated by a *US* code. The transmission is terminated with a *CR* code.

Send page unformatted

⟨ESC⟩ 3
Causes the IQ 150 to perform a *send page foreground* operation (see above) but will not send the *FS* or *US* delimiters. The trailing *CR* is sent.

Start message marker

⟨ESC⟩ D
Used with *end message marker.* Causes the writing of a 'start of message' code at the cursor location. This code is normally an *ASCII STX* code but may be reprogrammed (See *flexibility* item 11).

Tab

⟨FSC⟩ i or ⟨HT⟩ or *tab/bk tab* key
Protect mode — causes the cursor to skip to the start of the next unprotected field (*home* is a legal stopping point).
Not protect mode — writes an *ASCII HT* code if no *typewriter* tab stops exist. If tab stops are set, the cursor will move to the next stop.

Upline

⟨VT⟩ or ↑ key
Causes the cursor to move up one line while remaining in the same column. If the cursor was on the top line, then it is moved to the last line of the page. Movement out of a protected field is backward.

Write low off

⟨ESC⟩ (
Causes all successive data to be written as high intensity data. This data will be considered unprotected data when *protect* mode is on.

Write low on

⟨ESC⟩)
Causes all successive data to be written as low intensity data. This data will be considered protected data when *protect* mode is on.

Table 1
Cursor positioning coordinates

X/Y	code	X/Y	code	X/Y	code	X/Y	code	X/Y	code	X/Y	code
1	SP	17	0	33	@	49	P	65	\	81	p
2	!	18	1	34	A	50	Q	66	a	82	q
3	"	19	2	35	B	51	R	67	b	83	r
4	#	20	3	36	C	52	S	68	c	84	s
5	$	21	4	37	D	53	T	69	d	85	t
6	%	22	5	38	E	54	U	70	e	86	u
7	&	23	6	39	F	55	V	71	f	87	v
8	'	24	7	40	G	56	W	72	g	88	w
9	(	25	8	41	H	57	X	73	h	89	x
10	)	26	9	42	I	58	Y	74	i	90	y
11	*	27	:	43	J	59	Z	75	j	91	z
12	+	28	;	44	K	60	[	76	k	92	{
13	,	29	⟨	45	L	61	\	77	l	93	|
14	-	30	=	46	M	62	]	78	m	94	}
15	.	31	⟩	47	N	63	^	79	n	95	~
16	/	32	?	48	O	64	UL	80	o	96	RUB

IQ135 Control and action codes

Control codes			Action sequences					
765 4321	000	001	010	011	100	101	110	111
0000	NUL	DLE REV/BL/UL ATTRIBUTE	SP AUX OFF	0 TEST ONLY	@ AUX ON	P PRINT FORMAT	TEST ONLY	P PRINT UNFORMAT
0001	SOH ┌	DC1 BLINK/UL ATTRIBUTE	! PROG FCN KEYS	1 SEND ONE CHAR	A AUX OFF	Q INSERT CHAR SP	a	q INSERT CHAR NULL
0010	STX	DC2 REV/UL ATTRIBUTE	" KEYBOARD UNLOCK	2	B BLOCK MODE ON	R DELETE LINE SP	b	r DELETE LINE NULL
0011	ETX	DC3 REV/BLINK ATTRIBUTE	# KEYBOARD LOCK	3 SEND PG UNFMT	C CONV MODE ON	S SEND MSG FGND	c	s SEND MSG ALL
0100	EOT FIELD OFF ATTRIBUTE	DC4 ┴	$	4 SEND LN FGND	D ST MSG MARKER	T CLEAR EOL SP	d END MSG MARKER	t CLEAR EOL NULL
0101	ENQ ┐	NAK UNDERLINE ATTRIBUTE	%	5 SEND PG FGND	E INS LIN SPACES	U	e INS LIN NULLS	u
0110	ACK ┬	SYN BLINK ATTRIBUTE	& PROTECT MODE ON	6 SEND LN ALL	F	V PROT TAB COLUMN	f	v
0111	BEL BEEPER	ETB REVERSE ATTRIBUTE	/ PROTECT MODE OFF	7 SEND PG ALL	G	W DELETE CHAR SP	g	w DELETE CHAR NULL
1000	BS BACKSPACE CURSOR	CAN ┼	(WRT LOW MODE OFF	8 INSERT MODE OFF	H	X	h	x
1001	HT TAB CURSOR	EM ─	) WRT LOW MODE ON	9 INSERT MODE ON	I BACKTAB	Y CLEAR EOP SP	i TAB	y CLEAR EOF NULL
1010	LF LINE FEED CURSOR	SUB │	* CLR ALL NULLS	: CLR ALL SPACE	J	Z	j	z
1011	VT UPLINE CURSOR	ESC	+ CLR FGND NULLS	; CLR FGND SPACES	K	[SOFT RESET	k	} HARD RESET
1100	FF FORESPACE CURSOR	FS ├		<	L	/ CLR TABS TYPEWRTR	l	| SET TABS TYPEWRTR
1101	CR RETURN CURSOR	GS ┤	− (SEE BELOW)	= LOAD CURSOR	M	]	m	}
1110	SO └	RS HOME CURSOR	INTELL FLEX	>	N	^	n	~
1111	SI ┘	US NEWLINE CURSOR	/ FCN KEY MESSAGES	? READ CURSOR	O	UL	o	RUB

Special functions (ESC − sequences)

0-Answerback 1-Low markers 2-High markers 3-Multiple chars 4-Graphics on 5-Graphics off

Switch bank K8

Switch up (off)	*Switch down (on)*
1. PRINT ALL	PRINT UNPROT ONLY
2. SUPPRESS ECHOED ESC	DISABLED
3. BLOCK MODE PAD	DISABLED
4. IQ135 MODE	IQ120 MODE
5. CONV FDX	CONV HDX
6. SHIFTED LEADIN-KB	UNSHIFTED LEADIN
7. BEEP ENABLED	DISABLED
8. BREAK DISABLED	ENABLED

Switch bank K10

Switch up (off)	*Switch down (on)*
1. 50 HZ REFRESH	60 HZ REFRESH
2. LINE 25 ON	LINE 25 OFF
3. RTS 'ON'	RTS CONTROLLED
4. MAIN BIT 8 'ONE'	MAIN BIT8 'ZERO'
5. MAIN 8 BIT DATA	MAIN 7 BIT DATA
6. MAIN EVEN PAR	MAIN ODD PAR
7. MAIN PAR OFF	MAIN PAR ON
8. MAIN 2 STOPS	MAIN 1 STOP

Switch bank 12

Switch up (off)	*Switch down (on)*
1. NON-BLINKING	BLINKING CURSOR
2. BLOCK CURSOR	UNDERLINE
3. PRTR RDY	DISABLED
4. PRTR BUSY	DISABLED
5. POR AUX PRT	POR NORMAL PRT
6. PRTR EVEN PAR	PRTR ODD PAR
7. PRTR PAR OFF	PRTR PAR ON
8. PRTR 2 STOPS	PRTR 1 STOP

Soroc Technology Inc.
165 Freedom Avenue
Anaheim, California 92801
(800) 854-0147
(714) 992-2860

Soroc Technology (Canada) Inc.
1 Scarsdale Road,
Don Mills, Ontario
Canada, M3B 2R2
(416) 441-1133