

Model 100 ROM Functions  
(700-2245)

## Model 100 ROM Functions

These pages provide essential information for using ROM functions.

## LCD Functions

Function Name	Description	Entry Address (Hex.)
LCD	Displays a character on the LCD at current cursor position. (Also RST 4) Entry condition: A = character to be displayed Exit condition: none	4B44
SETCUR	Move cursor to specified location. Entry conditions: D = column number (1-40) E = row number (1-8) Exit conditions: none	7440
PLOT	Turn on pixel at specified location. Entry conditions: D = x coordinate (0-239) E = y coordinate (0-63) Exit condition: none	744C
UNPLOT	Turn off pixel at specified location. Entry conditions: D = x coordinate (0-239) E = y coordinate (0-63) Exit condition: none	744D
POSIT	Set cursor position. Entry conditions: H = column number (1-40) L = row number (1-8) Exit condition: none	427C
ESCA	Send specified Escape Code Sequence. Entry conditions: A = escape code Exit conditions: none	4270

## Routines for Generating Common LCD Functions and Escape Codes

Function Name	Description	Entry Address (Hex.)	Equiv. ESC
CRLF	Generate a Carriage Return and Line Feed	4222	--
HOME	Move cursor to Home position (1,1)	422D	--
CLS	Clear Display	4231	--
SETSYS	Set system line (lock line 8, LABEL)	4235	T
RSTSYS	Reset system line (unlock line 8, LABEL)	423A	U
LOCK	Lock display (no scrolling)	423F	Y
UNLOCK	Unlock display (scrolling)	4244	W
CURSON	Turn on cursor	4249	P
CUROFF	Turn off cursor	424E	Q
DELLIN	Delete line at current cursor position	4253	M
INSLIN	Insert a blank line at cursor position	4258	L
ERAEOL	Erase from cursor to end of line	425D	K
ENTREV	Set Reverse character mode	4269	p
EXTREV	Turn off Reverse character mode	426E	q

## Variable and Status Locations

Name	Contents	Address
CSRY	Cursor Position (ROW)	F639
CSRX	Cursor Position (Column)	F63A
BEGLCD	Start of LCD memory	FE00
ENDLCD	End of LCD memory	FF40



## Keyboard Functions

Function Name	Description	Entry Address (Hex.)
KYREAD	Scan keyboard for a key. Return with or without one. Entry conditions: none Exit conditions: A = Character, if any Z Flag -- set if no key found -- reset if key found Carry -- set (character in code table below) -- reset (normal character set code)	7242

When Carry is set (1), Register A will contain one of the following:

Register A	Key Pressed
0	F1
7	F2
2	F3
3	F4
4	F5
5	F6
6	F7
7	F8
8	LABEL
9	PRINT
0A	SHIFT-PRINT
0B	PASTE

CHGET	Wait and get character from keyboard. Entry conditions: none Exit conditions: A = character code Carry -- set if special character -- reset if normal character (<F1> - <F8> return preprogrammed strings)	12CB
CHSNS	Check keyboard queue for characters Entry conditions: none Exit conditions: Z flag set if queue empty, reset if keys pending	13DB

KEYX	Check keyboard queue for characters or BREAK Entry conditions: none Exit conditions: Z flag set if queue empty, reset if keys pending Carry -- Set when BREAK entered Reset with any other key	7270
BRKCHK	Check for BREAK characters only (CTRL-C or -S) Entry conditions: none Exit conditions: Carry -- set if BREAK or PAUSE entered -- reset if no BREAK characters	7283
INLIN	Get line from keyboard (terminated by <ENTER> Entry conditions: none Exit conditions: data stored at location F685	4644

#### Using Function Keys Routines

The function table consists of character strings to be used by the keyboard driver when processing <F1> - <F8> keys. The strings have maximum length of 16 characters and are terminated by a "80" (Hex.) code. If the last character of the string is OR'ed with 80, the character will also serve as a terminator. The entire string will be placed in the keyboard buffer when the appropriate strings for all 8 key is pressed. You must specify character strings for all 8 function keys (use the terminator byte for any you wish to ignore).

Example of Function table:

```

FCTAB  DEFM  'Files'   ; F1
        DEFW  0D80
        DEFM  'Load'   ; F2
        DEFB  80
        DEFM  'Save'   ; F3
        DEFB  80
        DEFM  'Run'    ; F4
        DEFW  0D80
        DEFM  'List'   ; F5
        DEFW  0D80
        DEFB  80      ; Ignore F6
        DEFB  80      ; Ignore F7
        DEFM  'Menu'   ; F8
        DEFW  0D80
  
```

Function Name	Description	Entry Address (Hex.)
STFNK	Set Function key definitions Entry conditions: HL = Address of function table (above) Exit conditions: none	5A7C
CLRFLK	Clear function key definition table (fills with 80's) Entry conditions: none Exit conditions: none	5A79
DSPNFK	Display function keys Entry conditions: none Exit conditions: none	42A8
STDSPF	Set and display function keys Entry conditions: HL = Start address of function table Exit conditions: none	42A5
ERAPNK	Erase function key display Entry conditions: none Exit conditions: none	428A
FNKSB	Display function table (if enabled) Entry conditions: none Exit conditions: none	5A9E



## Printing Routines

Function Name	Description	Entry Address (Hex.)
PRINTR	Send a character to the line printer Entry conditions: A = character to be printed Exit conditions: Carry -- set if cancelled by BREAK -- reset if normal return	6D3F
PNOTAB	Print character without expanding tab characters Entry conditions: A = character to be printed Exit conditions:	1470
PRTTAB	Print a character expanding tabs to spaces Entry conditions: A = character to be printed Exit conditions:	4B55
PRTLCD	Print contents of LCD Entry conditions: none Exit conditions: none	1E5E

## RS232-C and Modem Routines

Function Name	Description	Entry Address (Hex.)
DISC	Disconnect Phone Line Entry conditions: none Exit conditions: none	52BB
CONN	Connect Phone Line Entry conditions: none Exit conditions: none	52D0
DIAL	Dial a specified phone number Entry conditions: HL = ph. number address Exit conditions: none	532D

RCVX	Check RS232 queue for characters Entry conditions: none Exit conditions: A = number of characters in queue Z flag -- set if no data -- reset if characters pending	6D6D
RV232C	Get a character from RS232 receive queue Entry conditions: none Exit conditions: A = character received Z flag -- set if O.K. -- reset if error (PE, FF, or OF) Carry -- set if BREAK pressed, else reset	6D7E
SENSCQ	Send an XON resume character (CTL-Q) Entry conditions: none Exit conditions: none	6E0B
SENDCS	Send an XOFF pause character (CTL-S) Entry conditions: none Exit conditions: none	6E1E
SD232C	Send a character to the RS-232 or Modem (with XON/XOFF) Entry conditions: A = character to be sent Exit conditions:	6E32
CARDET	Detect carrier Entry conditions: none Exit conditions: A = 0 if carrier Z Flag -- Set if carrier, else reset	6EEF
SNDCOM	Send a character to RS232-C or modem (without XON/XOFF flow control) Entry conditions: C = character to be sent Exit conditions:	6E3A
BAUDST	Set Baud rate for RS232-C Entry conditions: H = Baud rate (1-9,M) Exit conditions: none	6E75



INZCOM Initialize RS232-C and Modem 6EA6  
 Entry conditions: H = Baud rate (1-9,M)  
                   L = UART configuration code  
                   (see UART byte description below)  
           Carry -- set if RS232-C  
                   -- reset if modem  
 Exit conditions: none

BIT(S)	Description
0	Specifies number of Stop Bits: 0=1, 1=2
1-2	Parity Setting: 00=None, 01=Even, 10=Odd
3-4	Word Length: 00=6, 01=7, 10=8

The byte is ANDed with 1FH to ignore Bits 5-7. The text string containing the current STAT setting is located at F65BH (5 Bytes): Baud, Length, Parity, Stop Bits, and XON/XOFF switch.

-----  
 SETSER Set serial interface parameters and activate 17E6  
 RS232-C/Modem  
 Entry conditions: HL = start address of ASCII string containing parameters terminated by a binary zero ('78E1E', 0). Syntax same as in Telcom's STAT  
           Carry -- set for RS232-C  
                   -- reset for Modem  
 Exit conditions: none

-----  
 CLSCOM Deactivate RS232-C/Modem 6ECB  
 Entry conditions: none  
 Exit conditions: none  
 -----

#### Cassette Recorder Routines

Function Name	Description	Entry Address (Hex.)
DATAR	Read character from cassette (no checksum) Entry conditions: none Exit conditions: D = character from cassette	702A

CTON	Turn motor on Entry conditions: none Exit conditions: none	14A8
CTOFF	Turn motor off Entry conditions: none Exit conditions: none	14AA
CASIN	Read a character from cassette and update checksum Entry conditions: C = current checksum Exit conditions: A = character C = contains the updated checksum	14B0
CSOUT	Send character to cassette and update checksum Entry conditions: A = character to be sent C = current checksum Exit conditions: C = updated checksum	14C1
SYNCW	Write cassette header and sync byte only Entry conditions: none Exit conditions: none	6F46
SYNCR	Read cassette header and sync byte only Entry conditions: none Exit conditions: none	6F85
DATAW	Write a character to cassette (no checksum) Entry conditions: A = character to be sent Exit conditions: none	6F5B

### RAM Files Routines

The Directory Table (located at F962) contains all file location, type, and status information.

Each file is managed by an 11-byte directory entry in the format:

Byte 1: Directory Flag (for file type and status)  
 Bytes 2-3: Address of file  
 Bytes 4-11: 8 Byte filename



The Directory Flag contains the following information:

Bit 7 (MSB)	1 if a valid entry
Bit 6	1 for ASCII text file (DO)
Bit 5	1 for Machine language (CO)
Bit 4	1 for ROM file
Bit 3	1 for invisible file
Bit 2	reserved for future use
Bit 1	reserved for future use
Bit 0	internal use only

Function Name	Description	Entry Address (Hex.)
MAKTXF	Create a text file Entry conditions: filename (max. 8 bytes) must be stored in PILNAM (0FC93). 'DO' extension not required Exit conditions: HL = TOP address of new file DE = address of Directory entry (Flag) Carry -- set if file already exists -- reset if new file	220F
CHKDC	Search for file in directory Entry conditions: DE = address of filename to find (ASCII filename + 0 byte terminator) Exit conditions: HL = start address (TOP) of file Z Flag -- 0 (file found) -- 1 (file not found)	5AA9
GTXTTB	Get top address of file Entry conditions: HL = address of directory entry for file Exit conditions: HL = TOP start address of file	5AE3
KILASC	Kill a text (DO) file Entry conditions: DE = file TOP start address HL = address of directory entry (flag) Exit conditions: none	1FBE
INSCHR	Insert a character in a file Entry conditions: A = character to insert HL = address to insert character Exit conditions: HL = +1 Carry -- set if out of memory	6B61





DAY            Read system DAY of the week            1962  
              Entry conditions: HL = address of 3 byte area  
  for DAY  
              Exit conditions: HL => DAY (ddd)

---