

Module Terminal

The Module terminal program is a complete development system for the CMS Range of Modules. It provides simple serial terminal features for communication with the module as well as a number of extra facilities for file transfers, window management, and graphical display. The extra facilities are accessed using an escape character (ASCII 27) followed by a function number and a number of parameters. Some functions have a fixed number of parameters others have a variable number. Variable parameters are always strings and end with a return (ASCII 13). The notes below describe each function that the CMS terminal program is designed to accept and what the function does.

Function 1 - Initialise Graphics Board

This function takes a one byte parameter and is used to specifically initialise the PC video display board.

Parameter 1	10	Initialise CGA board
	20	Initialise EGA board
	30	Initialise VGA board

All other values should be ignored.

Function 2 - Switch to graphics mode

This function does not require any parameters and switches the display to graphics mode.

Function 3 - Switch to text mode

This function does not require any parameters and switches the display to text mode.

Function 4 - Draw a line from X1, Y1 to X2, Y2

This function takes nine parameters and draws a coloured line

Parameter 1	High byte of X1
Parameter 2	Low byte of X1
Parameter 3	High byte of Y1
Parameter 4	Low byte of Y1
Parameter 5	High byte of X2
Parameter 6	Low byte of X2
Parameter 7	High byte of Y2
Parameter 8	Low byte of Y2
Parameter 9	Colour

Function 5 - Draw a circle

This function takes 7 parameters and draws an unfilled circle with centre X, Y and radius R.

Parameter 1	High byte of X
Parameter 2	Low byte of X
Parameter 3	High byte of Y
Parameter 4	Low byte of Y
Parameter 5	High byte of R
Parameter 6	Low byte of R
Parameter 7	Colour

Function 6 - Draw a disk

This function uses the same parameters as function 5 but draws a filled circle.

Function 7 - Draw an ellipse

This function draws an ellipse with the centre at X,Y with major axis A and minor axis B. The ellipse is of colour C and can be filled or outline. This requires 10 parameters in all

Parameter 1	High byte of X
Parameter 2	Low byte of X
Parameter 3	High byte of Y
Parameter 4	Low byte of Y
Parameter 5	High byte of A
Parameter 6	Low byte of A
Parameter 7	High byte of B
Parameter 8	Low byte of B
Parameter 9	Colour
Parameter 10	0 = outline 1 = filled

Function 8 - Plot a coloured point.

This function plots a coloured point at position X, Y in colour C. It requires 5 parameters.

Parameter 1	High byte of X
Parameter 2	Low byte of X
Parameter 3	High byte of Y
Parameter 4	Low byte of Y
Parameter 5	Colour

Function 9 - Draw a coloured box

This function draws a coloured box with top left corner X1,Y1 and bottom right corner X2, Y2. It requires 10 parameters

Parameter 1	High byte of X1
Parameter 2	Low byte of X1
Parameter 3	High byte of Y1
Parameter 4	Low byte of Y1
Parameter 5	High byte of X2
Parameter 6	Low byte of X2
Parameter 7	High byte of Y2
Parameter 8	Low byte of Y2
Parameter 9	Colour
Parameter 10	0 = outline 1 = filled

Function 10 - Set text colour

This function set the text foreground colour and requires one parameter.

Parameter 1	New text colour
-------------	-----------------

Function 11 - Set background colour

This function sets the text background colour and requires one parameter.

Parameter 1	New background colour.
-------------	------------------------

Function 12 - Clear Screen

This function clears the current screen and returns the cursor to the top left corner. No parameters are required.

Function 13 - Move cursor

This function moves the cursor to position X, Y. The screen origin is assumed to be at 1, 1 and is in the top left corner. Two parameters are required

Parameter 1 X position
Parameter 2 Y position

Function 14 - Open a window

This function opens a new window with its top left corner at position X1, Y1 and its bottom right corner at position X2, Y2. The window is made the current output screen and is put on top of the other windows. 5 parameters are required

Parameter 1 Window number
Parameter 2 X1
Parameter 3 Y1
Parameter 4 X2
Parameter 5 Y2

Function 15 - Use a window

This function switches the output to a new window. The new window doesn't need to be the top one. One parameter is required

Parameter 1 Window number.

Function 16 - Hide a window

This function stops a window from being displayed. A window can still be written to while it is hidden. One parameter is required.

Parameter 1 Window number.

Function 17 - Move a window to the front

This function moves a window to the front of the display. This does not need to be the current window. One parameter is required

Parameter 1 Window number

Function 18 - Insert Line

This function inserts a blank line at the cursor position and scrolls the rest of the screen down one line. No parameters are required.

Function 19 - Delete Line

This function deletes the current line and scrolls the rest of the screen up one line to leave a new blank line at the bottom of the screen. No parameters are required.

Function 20 - Clear to end of line

This function clears from the character at the current cursor position to the end of the current line. No parameters are required.

Function 21 - Cursor On / Off

This function turns the cursor on and off. One parameter is required.

Parameter 1 0 = Turn cursor off
 1 = Turn cursor on

Function 22 - Open a file

This function is used by the module's filing system to open a file on the PC. One parameter is required but it must be in the form of a string and is used as the filename. The string is always terminated with a return character. E.g. the bytes sent by the Module to open a file called fred would be

27, 22, 102, 114, 101, 100, 13

The host system should respond to this in one of two ways. If the open call fails for some reason the terminal program should send the following bytes

0x9b, 0x80, error, 0x0d

If the open call is successful the terminal program should return a code number which can be used by subsequent calls to identify this file as follows

0x9b, 0x00, file id number, 0x0d

Function 23 - Close a file

This function is used by the Module's filing system to close a file that was previously opened using the 'open' or 'create' functions. One parameter is required.

Parameter 1 file id number

If the file is closed successfully the terminal program should respond with the following bytes

0x9b, 0x00, 0x0d

If the close function fails for some reason the terminal should respond as follows

0x9b, 0x80, error number, 0x0d

Function 24 - Read a block of data from a file

This function is used to read a block of binary data from a file. The file is identified by a file id number returned by 'open' or 'create'. This function requires 2 parameters.

Parameter 1 File id number
Parameter 2 Number of bytes to read.

If the read request fails completely for some reason (bad file id, file has reached EOF etc.) and no bytes are read the terminal should respond by sending

0x9b, 0x80, error number, 0x0d

If some data is read from the file the terminal should respond as follows

0x9b, 0x00, n bytes of data, 0x0d

NB the data field should be modified from the original as follows. If the original contains an 0x9b an extra 0x9b must be added directly after it and if the original contains an 0x0d it should be replaced with 0x9b followed by 0x2d. If the end of file is reached before the number of characters required has been read this function should return as many characters as it can. The end of file error should only be sent if the file was already at EOF when the read function was called.

Function 25 - Read a line of text from a file

This function works in exactly the same way as read except that the read will be terminated early if a return character is encountered in the original data.

Function 26 - Write a block of data to a file

This function is used to write to files on the PC which have previously been opened which 'open' or 'create'. Two parameters are required one is the file id number the other is a variable length data block which has been encrypted in the same way as the 'read' and 'readln' functions. If the write is successful the terminal should respond with

0x9b, 0x00, 0x0d

If there is an error it should respond with

0x9b, 0x80, error number, 0x0d

Function 27 - Send ESC

This function passes the ESC character (ASCII 27) to the screen. No parameters are required.

Function 28 - Set new file pointer.

This function allows the file pointer to be moved in a file allowing random access to data files. Five parameters are required

Parameter 1	File id number
Parameter 2	Lowest byte of new position
Parameter 3	Lower byte of new position
Parameter 4	Higher byte of new position
Parameter 5	Highest byte of new position

If the new file position is set correctly the terminal should respond with

0x9b, 0x00, 0x0d

If it fails it should respond with

0x9b, 0x80, error number, 0x0d

NB If the file pointer is moves passed the end of file subsequent read requests will return the EOF error but writes will extend the file with junk before starting to write at the new position.

Function 29 - Read file pointer

This function returns the current file pointer position in a file. One parameter is required

Parameter 1	File id number
-------------	----------------

If the file pointer can be read the terminal should respond as follows

0x9b, 0x00, current file position, 0x0d

If there is an error it should respond with

0x9b, 0x80, error number ,0x0d

NB the file position should be transmitted low byte first and should have the same encryption as the 'read' and 'write' functions.

Function 30 - Create a file

This function is used to create a file which doesn't already exist. The parameters are the same as for open but a new file is created. If the file already exists it will be truncated to zero length.

Function 31 - Delete a file

This function deletes an existing file. The filename is send in the same way as it is for 'open' and 'create'. If the file can be deleted the terminal should respond with

0x9b, 0x00, 0x0d

If an error occurs it should respond with

0x9b, 0x80, error number, 0x0d

NB it should not be possible to delete a file which is open.

Function 32 - Is never used

Function 33 - Read file size

This function returns the size of a file in bytes. One parameter is require

Parameter 1 file id number

If an error occurs the terminal should respond with

0x9b, 0x80, error number, 0x0d

If the file size can be read it should return

0x9b, 0x00, file size, 0x0d

NB the file size should be send low byte first and encrypted in the same way as read file position etc.

Future Developments

In order to make the terminal program more portable we would like a version which only implements the following functions

12, 13, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 & 33

If any of the other functions are received this version should collect the appropriate parameters but should not do any work