

Peptu Instruction Set Details

Author : John Hudson
Version : 1
Date : 9th August 2004

Command	Parameters	Action
0		Null Action
1		Cursor Home - places cursor at top left of the display (row 0, column 0)
2		Hide Display - blanks display but retains displayed data
3		Restore Display - returns from Hide Display and shows original data
4		Cursor Off - removes visible cursor from the display
5		Underline Cursor On - place an underline cursor at the current cursor position (replaces block cursor if currently displayed)
6		Block Cursor On - place a block cursor at the current cursor position (replaces underline cursor if currently displayed)
7		Null Action
8		Backspace and Erase - moves cursor back 1 character position and erases the data at that position
9	screen_display	<p>Boot Screen Options - this allows the user to specify the required boot screen display.</p> <p>screen_display has three possible values:</p> <p>0 - clears the display and awaits the first character</p> <p>1 - presents the Peptu copyright screen and awaits the first character</p> <p>2 - presents the currently active user defined screen and awaits the first character. Note that a user defined screen must first be created and saved to the EEPROM. If no user defined screen has been created and saved the screen display is undefined.</p> <p>The default value is 1, thus presenting the Peptu screen.</p>
10		<p>Line Feed - moves cursor down one row retaining the current column position.</p> <p>If on bottom row:</p> <p>If Scroll is OFF - it will move to the same character position in the top row</p> <p>If Scroll is ON - it will scroll the display up by one row, clear the bottom row and set the cursor to the left hand position of the bottom row</p>
11		Delete Character At Cursor Position - deletes the character at the cursor position and leaves the cursor at that position
12		Clear the Display - clears all rows of the display and sets the cursor position to row 0 column 0
13		Carriage Return - moves cursor to the column 0 on the current row (moves to the left hand end)
14		Null Action
15		Null Action
16		Null Action
17	column row	<p>Set Cursor Column and Row</p> <p>column is the column number required. On a 16 character display it can be anything between 0 & 15, on a 20 character display between 0 & 19.</p> <p>row is desired row number; valid values are 0 and 1 for a two row display and 0 - 3 for a 4 row display. The 'home' position (extreme left on top row) is at column 0, row 0.</p>

Command	Parameters	Action
18	graph_UDC_nos bar_style start_col end_col length row	<p>Horizontal Bar Graph</p> <p>The horizontal bar graph feature allows a bar graph to be created that runs from left to right, right to left or both at the same time.</p> <p>The area occupied by the graph is totally cleared for each graph write so it is not necessary to clear the area for successive updates to the same graph</p> <p>graph_UDC_nos specifies the UDC to be used, valid number are 0 - 7</p> <p>bar_style defines the 'look' of the UDC character to be used:</p> <p>255 sets all bits in all of the character generation rows making up a character - this produces a maximum height bar</p> <p>0 turns all pixels off so no bar will be seen</p> <p>85 sets all pixels in character generation rows 0,2,4 and 6 and turns all others off - this produces a striped bar</p> <p>60 sets all pixels in character generation rows 2,3,4 & 5 and turns the others off - this produces a medium height bar centrally displayed in the character block</p> <p>15 sets all pixels in character generation rows 0 - 3 producing a medium height bar displayed at the bottom of the character block</p> <p>240 sets all pixels in character generation rows 4 -7 producing a medium height bar displayed at the top of the character block.</p> <p>start_col / end_col - are the start and end column for the graph. Valid values are 0 - 19 for a 20 character display and 0 - 15 for a 16 character display - start_col must be equal to , or less than end_col .</p> <p>length is the length of the bar in pixels. Note that in order to stay compliant with Crystalfontz displays I have left this at pixels despite that fact that, unlike the 632/634 units which are graphics displays and thus can handle individual vertical pixel rows, Peptu uses a character based display and thus can only handle characters (or 6 columns of pixels). Peptu modifies the value sent to use characters. This does mean that the length of the graph is less precise than the 632/634 devices but.....</p> <p>It would be possible to have the same degree of accuracy as the Crystalfontz units but only by using 5 UDCs for each graph, UDC 0 has just the left hand col set, UDC 2 has the left hand plus the next on the right and so on, but only on graphs going from left to right (or only going from right to left) but I chose not to do this as it would be very restrictive. A compromise would be to use UDC 0 (for example) as a full width character and UDC 1 as a half with character and UDC 2 as a half width character going the other way (right to left) thus improving the accuracy and allowing graphs going in both directions to operate at the same time - I may do this at a later date.</p> <p>Note that for graphs going from left to right the length value must be positive whilst for graphs going from right to left that value must be negative. Positive values start the graph at the start_col and move to the right whilst negative values start at the end_col and move left.</p> <p>row specifies the row into which the graph will be placed and valid value are 0 - 3 for a 4 row display and 0 or 1 for a 2 row display.</p>
19		<p>Scroll On - with scroll on a line feed at the bottom line will scroll the display up by one row, clear the bottom row and set the cursor to column 0. If Wrap is also on a wrap occurring on the bottom line will also cause the display to scroll up one row, clear the bottom row and position the cursor at column 0.</p>

Command	Parameters	Action
20		Scroll Off - If scroll is set to off then a line feed on the bottom row will move the cursor to the same column in the top row.
21		Null Action
22		Null Action
23		Wrap On - if the cursor is currently at the end column of a row and a new character is received the display will wrap to column 0 in the next row. If that action occurs when the cursor is on the bottom row of the display the cursor will wrap to column 0 of the top row. If Scroll is also set to on then the display will scroll up one row, clear the bottom row and set the cursor to column 0.
24		Wrap Off - if the cursor is set to the last column of a row and another character is received the cursor will move off the display to the right and all additional input will be 'lost' until a subsequent command moves the cursor back to the visible area of the display.
25	character number data 0 data 1 data 2 data 3 data 4 data 5 data 6 data 7	<p>Set User Defined Character</p> <p>Up to 8 user defined characters (UDC) can be created, each character occupying a five pixel wide and 8 pixel deep character block. To create a UDC the user must first send the character number followed by 8 data bytes, one for each row that makes up the character. The UDC character data is stored at character positions 128 to 135 (decimal) so to display UDC 0 the value 128 should be sent to the screen.</p> <p>The character number specifies the user defined character being created and can be any value between 0 & 7.</p> <p>The next 8 bytes are the bit data for the 8 rows that make up a character. The value for each of these bytes can vary between 0 and 63 (the Peptu controller will remove any bits that occur in bit positions 7 and 8).</p> <p>Data 0 is the top row of the character and data 7 the bottom. The bit significance of row data is bit 0 to the right hand (least significant) end and bit 7 (most significant) at the left hand end.</p>
26		<p>Reboot - the reboot action is to cause the Peptu controller to be reset and thus emulate the action that occurs when the interface is initially powered up. It will set the modules parameters to those previously stored into the controllers EEPROM</p> <p>As it is possible that the interface could be stuck in a loop awaiting data to complete a previous command, issuing a single reset might not work. In the worst case the interface could be awaiting up to 9 characters so the best way to ensure that a 'clean' reboot occurs is to send 9 blanks (32 - decimal) followed by the reboot command. Sending 6 reboot commands should also work.</p>

Command	Parameters	Action
27	91 65 - cursor up 91 66 - cursor down 91 67 - cursor right 91 68 - cursor left	<p>Escape Sequence - the escape sequence essentially emulates the action of the up, down, left and right 'arrow' cursor keys.</p> <p>The escape sequence only impacts upon the cursor position and do not impact on the displayed data. No wrap action occurs. When the cursor reaches the extremity of the display the cursor stops at that position, if it is moving left and hits column 0 then it ignores any further move left commands; if moving down and it reaches the bottom line then it ignores any more attempts to go lower.</p>
28	style number column	<p>Large Block Characters</p> <p>This feature allows a large numeral to be displayed on a 4 row LCD. Only digits 0 to 9 are available and all of the user defined characters are required. This means that any already defined UDCs will be 'corrupted' by using large characters so large characters can only be used if no other UDC requirement exists. The characters are pre-defined in Peptu and no initialisation is required, i.e. if you send command 28 followed by style, column, number then a large character spanning all 4 rows and each occupying 3 horizontal character positions will be presented. This means that it is only possible to display 6 large number digits on a 20 x 4 display (6 numbers = 18 character positions but this is not too practical as a further 3 characters are required to present a 'blank column' between each character position so for all practical purposes you are restricted to 5. However, that is sufficient to present a clock showing hours and minutes in large characters and seconds in normal size characters.</p> <p>The Crystalfontz USB system 634 has two styles a 3 x 4 (style 0) and a 4 x 4 (style 1) character format. Peptu has only the 3 x 4 format but the style value can be either 0 or 1. (4 x 4 may come later)</p> <p>The column parameter specifies the column that the left hand side of the large character will occupy.</p> <p>number is the number to display (0 to 9).</p>
29		Null Action
30		Null Action

Command	Parameters	Action
31	no_of_bytes start_address	<p>Pass EEPROM Content to PC</p> <p>This command does not show any information on the LCD. It returns to the PC the data held in the 128 byte Peptu EEPROM.</p> <p>no_of_bytes is the number of bytes required - valid values are between 0 and 127</p> <p>start_address is the EEPROM address that the first byte is to be read from. For multiple address (i.e. Fetch first 10 bytes) only the start address needs to be sent to Peptu - valid addresses are between 0 and 127</p> <p>The more important EEPROM data positions are shown below - the displacement is in bytes.</p> <p>0 - number of rows 1 - row length 2 - flags bit 0 = Scroll - if set then scroll is on bit 1 = wrap - if set is on bit 5 may or may not be set but has no significance 3 - Cursor 0 = cursor off 1 = underline cursor 2 = block cursor 4 - Boot Option 0 = clear screen and await input 1 = Set Peptu copyright screen and await input 2 = Set current user defined screen and await input 5 - 15 - not defined 16 - 80 - User Defined Character data 81 - 111 - not defined 112 - Major Software Version (PIC code) 113 - Minor Software Version (PIC code) 114 - Patch Level (PIC Code) 115 - Hardware Version 116 - Build Date (Day) 117 - Build Date (Month) 118 - Build Date (Year) 119 - 127 - not defined</p>