

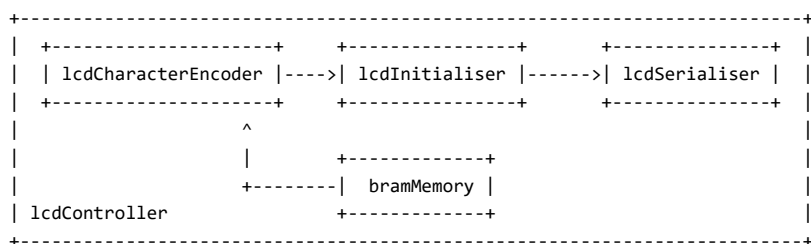
LCD Usage Explanations

Overview

The Lcd and Lcd_test library is intended to use with the HEB_LCD_V2_0, see http://wiki.hevs.ch/uit/index.php5/Hardware/Parallelport/heb_lcd.

The HEB_LCD_V2_0 features a EAD06M132-5 LCD. Its a 4 Line and 132 column black- light display. The font created for it uses 6 columns per character which allows to have 4 lines with 22 characters each.

The connection is I2C based and some initial configuration needs to be done. The bloc lcdController has everything needed to use the LCD. See the vague toplevel overview below.



Bloc description

- lcdController - Main bloc containing all needed subblocs
- lcdCharacterEncoder - This bloc receives the ASCII data to display and converts it to display data or display commands
- bramMemory - Contains the command for command chars (<0x20) and the display (pixel) data for a the symbol chars (>=0x20). All the data of this ram are stored in the font.txt file
- lcdInitialiser - This bloc lets send predefined commands for 2 features. One is for initialising the display in the beginning, and the other is for clear all data from the display
- lcdSerialiser - This bloc takes the parallised data and sends them over the I2C bus.

Supported Characters

In VHDL the characters can be written with the following command:

```
character'pos("Value")
```

where Value is indicated in the table below.

Hex	Value	Description	Function
0x00	"nul"	Null	Not supported
0x01	"soh"	Start of Header	Not supported
0x02	"stx"	Start of Text	(0,0) Goto pos line 0 character 0
0x03	"etx"	End of Text	(3,21) Goto pos line 3 character 21
0x04	"eot"	End of Transmission	Not supported
0x05	"enq"	Enquiry	Not supported
0x06	"ack"	Acknowledgment	Not supported
0x07	"bel"	Bell	Not supported
0x08	"bs"	Backspace	(y,x-1) Jump one char back

Hex	Value	Description	Function
0x09	"ht"	Horizontal Tab	Not supported
0x0A	"lf"	Line Feed	(y+1,x)Goto next line
0x0B	"vt"	Vertical Tab	(y-1,x) Goto previous line
0x0C	"ff"	Form Feed	Not supported
0x0D	"cr"	Carriage Return	(y,0) Goto beginning of line
0x0E	"so"	Shift Out	Not supported
0x0F	"si"	Shift In	Not supported
0x10	"dle"	Data Link Escape	Not supported
0x11	"dc1"	XON Device Control 1	Not supported
0x12	"dc2"	Device Control 2	Not supported
0x13	"dc3"	XOFFDevice Control 3	Not supported
0x14	"dc4"	Device Control 4	Not supported
0x15	"nak"	Negativ Acknowledgemnt	Not supported
0x16	"syn"	Synchronous Idle	Not supported
0x17	"etb"	End of Trans. Block	Not supported
0x18	"can"	Cancel	Clear entire display
0x19	"em"	End of Medium	Not supported
0x1A	"sub"	Substitute	Not supported
0x1B	"esc"	Escape	Not supported
0x1C	"fsp"	File Separator	Not supported
0x1D	"gsp"	Group Separator	Not supported
0x1E	"rsp"	Reqst to SendRec. Sep.	Not supported
0x1F	"usp"	Unit Separator	Not supported
0x20	" "	Space	Print char
0x21	"!"	exclamation mark	Print char
0x22	"\""	double quote	Print char
0x23	"#"	number sign	Print char
0x24	"\$"	dollar sign	Print char
0x25	"%"	percent	Print char
0x26	"&"	ampersand	Print char
0x27	"\""	single quote	Print char
0x28	"("	left/open parenthesis	Print char
0x29	")"	right/closing parenth.	Print char
0x2A	"*"	asterisk	Print char
0x2B	"+"	plus	Print char
0x2C	","	comma	Print char

Hex	Value	Description	Function
0x2D	"-"	minus or dash	Print char
0x2E	"."	dot	Print char
0x2F	"/"	forward slash	Print char
0x30	"0"		Print char
0x31	"1"		Print char
0x32	"2"		Print char
0x33	"3"		Print char
0x34	"4"		Print char
0x35	"5"		Print char
0x36	"6"		Print char
0x37	"7"		Print char
0x38	"8"		Print char
0x39	"9"		Print char
0x3A	":"	colon	Print char
0x3B	";"	semi-colon	Print char
0x3C	"<"	less than	Print char
0x3D	"="	equal sign	Print char
0x3E	">"	greater than	Print char
0x3F	"?"	question mark	Print char
0x40	"@"	AT symbol	Print char
0x41	"A"		Print char
0x42	"B"		Print char
0x43	"C"		Print char
0x44	"D"		Print char
0x45	"E"		Print char
0x46	"F"		Print char
0x47	"G"		Print char
0x48	"H"		Print char
0x49	"I"		Print char
0x4A	"J"		Print char
0x4B	"K"		Print char
0x4C	"L"		Print char
0x4D	"M"		Print char
0x4E	"N"		Print char
0x4F	"O"		Print char
0x50	"P"		Print char

Hex	Value	Description	Function
0x51	"Q"		Print char
0x52	"R"		Print char
0x53	"S"		Print char
0x54	"T"		Print char
0x55	"U"		Print char
0x56	"V"		Print char
0x57	"W"		Print char
0x58	"X"		Print char
0x59	"Y"		Print char
0x5A	"Z"		Print char
0x5B	"["	left/opening bracket	Print char
0x5C	"\"	back slash	Print char
0x5D	"]"	right/closing bracket	Print char
0x5E	"^"	caret/circumflex	Print char
0x5F	"_"	underscore	Print char
0x60	""		Print char
0x61	"a"		Print char
0x62	"b"		Print char
0x63	"c"		Print char
0x64	"d"		Print char
0x65	"e"		Print char
0x66	"f"		Print char
0x67	"g"		Print char
0x68	"h"		Print char
0x69	"i"		Print char
0x6A	"j"		Print char
0x6B	"k"		Print char
0x6C	"l"		Print char
0x6D	"m"		Print char
0x6E	"n"		Print char
0x6F	"o"		Print char
0x70	"p"		Print char
0x71	"q"		Print char
0x72	"r"		Print char
0x73	"s"		Print char
0x74	"t"		Print char

Hex	Value	Description	Function
0x75	"u"		Print char
0x76	"v"		Print char
0x77	"w"		Print char
0x78	"x"		Print char
0x79	"y"		Print char
0x7A	"z"		Print char
0x7B	"{"	left/opening brace	Print char
0x7C	" "	"	vertical bar
0x7D	"}"	right/closing brace	Print char
0x7E	"~"	tilde	Print char
0x7F	"del"	arrow to the left	Print char