

## Display



This is an intelligent 8 lines by 24 characters LCD display with a Philips 80C32 microcontroller, 64 kB RAM and 64 kB EPROM. The slider pot at the top is for the contrast setting. The EPROM contains an application that receives and displays data sent to it over a serial line at 19k2 baud 8/N/1. RS-232 level serial communication requires an external MAX232. The display requires a negative contrast voltage, which can be derived from the same MAX232.

---

## Interface

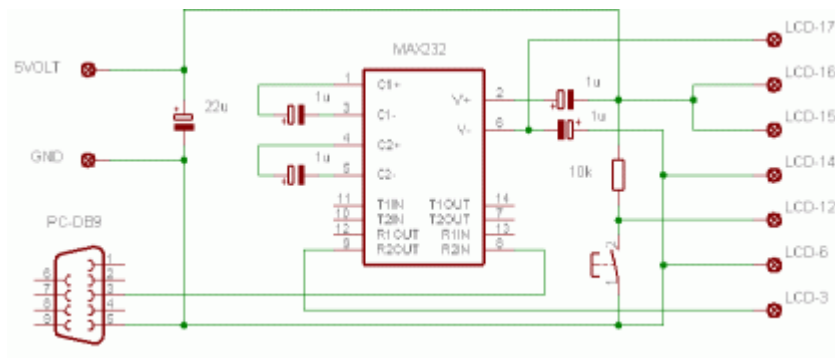
The display has a 17 pin connector, connections can be made with hookup wire. A wire can be inserted and will be kept in place by the connector. To remove wires, pull the moving part of the connector slightly away from the board. The use of the pins is shown in the table below. The Rx/D and Tx/D of the 80C31 UART and 7 other I/O pins are available. The Rx/D and Tx/D lines are at logic level, for connection to a serial port an inverter / level-shifter must be used. For use as serial display only the pins marked with a \* need be connected.

PIN	USE
17 *	Vee - 5 Volt (contrast voltage)
16 *	Vcc + 5 Volt
15 *	Vcc + 5 Volt
14 *	Gnd
13	80c31 port 1 pin 0
12	reset, active low, pull high for operational use
11	80c31 port 1 pin 7
10	80c31 port 3 pin 3
9	unknown, do not use

- 8 unknown, do not use
  - 7 80c31 port 1 pin 4
  - 6 \* Gnd
  - 5 80c31 port 1 pin 2
  - 4 80c31 port 1 pin 3
  - 3 \* 80c31 RxD (logic level)
  - 2 80c31 port 3 pin 2
- 
- 1 80c31 TxD (logic level)



The circuit below can be used to connect the display to a serial communication line. The main component is a MAX232, which adapts the logic levels to RS-232 and provides the negative contrast voltage.



## Protocol

Basically the display shows the printable ASCII characters you send to it. The backslash (\, code 0x5C) starts a

command sequence. Send two backslashes to print a backslash. Character 0xFF causes the next character to be ignored. Room for 32 fonts is available, but only fonts 0..3 are pre-defined. All fonts can be written by the using the Custom Character command.

## Buffer Control

format     0x5C 0x49 [command]

Use this command to prevent 'flickering' by writing to an off-screen buffer and updating the display from this buffer.

command = '9' : instant display updates on, writes go to buffer and display

command = ':' turns instant display updates off, writes go to the buffer only

command = '0' - '7' : write one line ( 0 .. 7 ) from buffer to display

command = '8' : write buffer to display

## Cursor Set

format     0x5C 0x42 [column + 0x20] [row + 0x20]

Use this command to put the cursor at an absolute position. The row and column positions are zero based.

## Custom Character

format     0x5C 0x44 [char high] [char low] [font + 0x20] [bitmap1] .. [bitmap8]

Use this command to download a custom character to RAM. The character number is sent as two hexadecimal characters, followed by a font number, and an 8-byte bitmap. You may write to any font. User defined fonts should be placed in 8 through 31. Low numbered fonts may also be written, but this should generally only be done if the appearance of particular characters needs to be changed. See the LCD Fonts page for images of the fonts included in the display board's firmware. All downloaded font bitmaps are stored in volatile memory and are lost when the board is rebooted.

## Cursor Move

format     move left                    0x5C 0x46 [offset + 32]

            move right                  0x5C 0x46 [offset + 64]

            move up                     0x5C 0x46 [offset + 112]

            move down                  0x5C 0x46 [offset + 96]

Use this command to move the cursor relative to its current position.

## Font Select

format     0x5C 0x43 [font + 0x20]

Use this command to select a font. It will affect characters displayed after this command, but it will not affect characters already on the display.

### Font Select and Clear

format      0x5C 0x40 [font+ 0x20] 0x30

Use this command to clear the display and return the cursor to the upper left corner. The current font is set to the specified font.

### Font Test

format      0x5C 0x40 [font + 0x20] [0x31 (first half) or 0x32 (second half)]

Use this command to show the characters defined in a font. A font contains more characters than fit on the display, so you must specify whether the first half (characters 0 .. 0x7F) or the second half (0x80 .. 0xFF) of the font is to be displayed. The 4 resident fonts are shown below.



### Line Clear

format      0x5C 0x4C [line + 0x20]

Use this command to clear the indicated line.

## Mode

format	wrap off, scroll off	0x5C 0x41 0x30
	wrap on, scroll off	0x5C 0x41 0x31
	wrap off, scroll on	0x5C 0x41 0x32
	wrap on, scroll on	0x5C 0x41 0x33

Use this command to set the wrap and scroll modes. Wrapping causes the cursor to advance automatically from beyond the last position of a line to the first position of the next line. When wrapping is off, additional characters are not drawn until the cursor is manually positioned back into the screen's area.

Scrolling causes the entire display to scroll vertically when the cursor is advanced downward beyond the last row of the display.

When scroll mode is off, the behavior is determined by the wrap mode. If wrap is on and scroll is off, downward cursor movement beyond the last row causes the cursor to reposition at the top of the screen. If both modes are off, no additional characters are drawn until the cursor is manually repositioned into the screen's area.

## Special Characters

format      see text

Use these commands to display characters with ascii codes outside the range of normal printable characters. To display a character with code 0 .. 0x19 send the two characters 0x5C [code + 0x20]. To display the character with code 0x7F send the two characters 0x5C 0x5F. To display a character with code 0x80 .. 0x9F send the two characters 0x5C [code - 0x20].

The display contains a program from [PRJC](#). More information about the program can be found at their website. The source is public domain, so you are free to modify it to suit your needs. It can be downloaded from the PRJC site (it is part of the software for their MPEG player). The PRJC site also gives a information about how to develop your own application for the display. The PRJC page also has a more complete but IMHO less readable description of the protocol.