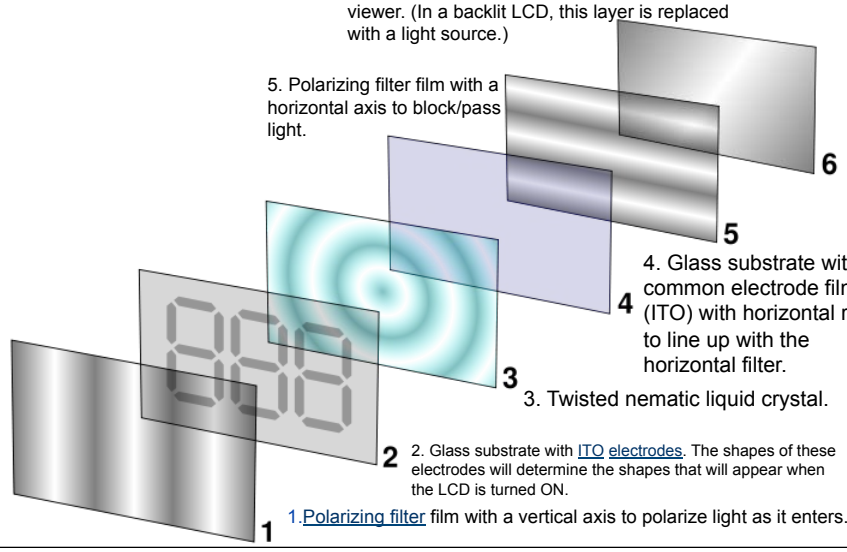
LCD displays

The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.

LCD Displays

6. Reflective surface to send light back to viewer. (In a backlit LCD, this layer is replaced with a light source.)

5. Polarizing filter film with a horizontal axis to block/pass light.



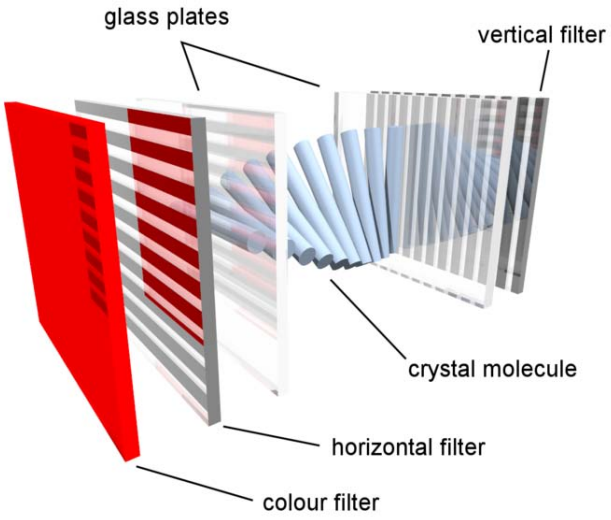
4. Glass substrate with common electrode film (ITO) with horizontal ridges to line up with the horizontal filter.

3. Twisted nematic liquid crystal.

2. Glass substrate with ITO electrodes. The shapes of these electrodes will determine the shapes that will appear when the LCD is turned ON.

1. Polarizing filter film with a vertical axis to polarize light as it enters.

LCD Displays



Workhorse Displays

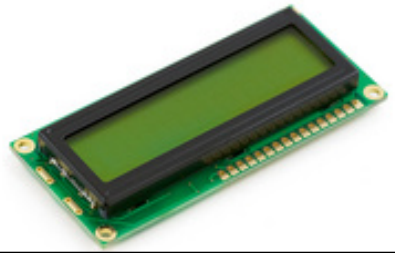
LCD Advantages

- Low Power – lots of battery life
- Lots of pixels are possible
- Lots of LCD shapes are possible
- Inexpensive

- But – LCDs don't emit light so they need a backlight

Form factors

- One common style is the 2-line 16-character display
  - Almost always uses the same controller
    - Hitachi HD44780 LCD controller chip
  - 4- or 8-bit data interface
    - Need 7-11 pins to drive the interface depending on how it's configured



HD 44780-compatable LCDs

Signal Name	FPGA Pin	Function	
SF_D<11>	M15	Data bit DB7	Shared with StrataFlash pins SF_D<11:8>
SF_D<10>	P17	Data bit DB6	
SF_D<9>	R16	Data bit DB5	
SF_D<8>	R15	Data bit DB4	
LCD_E	M18	Read/Write Enable Pulse 0: Disabled 1: Read/Write operation enabled	
LCD_RS	L18	Register Select 0: Instruction register during write operations. Busy Flash during read operations 1: Data for read or write operations	
LCD_RW	L17	Read/Write Control 0: WRITE, LCD accepts data 1: READ, LCD presents data	

HD 44780-compatable LCDs

The Display Data RAM (DD RAM) stores the character code to be displayed on the screen. Most applications interact primarily with DD RAM. The character code stored in a DD RAM location references a specific character bitmap stored either in the predefined CG ROM character set or in the user-defined CG RAM character set.

Figure 5-3shows the default address for the 32 character locations on the display. The upper line of characters is stored between addresses 0x00 and 0x0F. The second line of characters is stored between addresses 0x40 and 0x4F.

	Character Display Addresses																Undisplayed Addresses		
1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	…	27
2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	…	67
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	…	40

Figure 5-3: **DD RAM Hexadecimal Addresses (No Display Shifting)**

HD 44780-compatable LCDs

Figure 5-4: **LCD Character Set**

						Upper Nibble				Lower Nibble			
						Write Data to CG RAM or DD RAM							
A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
Character Address			Row Address			Don't Care			Character Bitmap				
0	1	1	0	0	0	-	-	-	0	■	0	■	0
0	1	1	0	0	1	-	-	-	■	0	■	0	■
0	1	1	0	1	0	-	-	-	0	■	0	■	0
0	1	1	0	1	1	-	-	-	■	0	■	0	■
0	1	1	1	0	0	-	-	-	0	■	0	■	0
0	1	1	1	0	1	-	-	-	■	0	■	0	■
0	1	1	1	1	0	-	-	-	0	■	0	■	0
0	1	1	1	1	1	-	-	-	0	0	0	0	0

Figure 5-5: **Example Custom Checkerboard Character with Character Code 0x03**

Table 5-3: **LCD Character Display Command Set**

Function	LCD_RS	LCD_RW	Upper Nibble				Lower Nibble			
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear Display	0	0	0	0	0	0	0	0	0	1
Return Cursor Home	0	0	0	0	0	0	0	0	1	-
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S
Display On/Off	0	0	0	0	0	0	1	D	C	B
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	-	-

Table 5-3: **LCD Character Display Command Set *(Continued)***

Function	LCD_RS	LCD_RW	Upper Nibble				Lower Nibble			
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Function Set	0	0	0	0	1	0	1	0	-	-
Set CG RAM Address	0	0	0	1	A5	A4	A3	A2	A1	A0
Set DD RAM Address	0	0	1	A6	A5	A4	A3	A2	A1	A0
Read Busy Flag and Address	0	1	BF	A6	A5	A4	A3	A2	A1	A0
Write Data to CG RAM or DD RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0
Read Data from CG RAM or DD RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0

HD 44780-compatable LCDs

Example Timing…

Clear Display

Clear the display and return the cursor to the home position, the top-left corner.

This command writes a blank space (ASCII/ANSI character code 0x20) into all DD RAM addresses. The address counter is reset to 0, location 0x00 in DD RAM. Clears all option settings. The I/D control bit is set to 1 (increment address counter mode) in the Entry Mode Set command.

Execution Time: 82 µs – 1.64 ms

Return Cursor Home

Return the cursor to the home position, the top-left corner. DD RAM contents are unaffected. Also returns the display being shifted to the original position, shown in Figure 5-3.

The address counter is reset to 0, location 0x00 in DD RAM. The display is returned to its original status if it was shifted. The cursor or blink move to the top-left character location.

Execution Time: 40 µs – 1.6 ms

HD 44780-compatable LCDs

Example Timing…

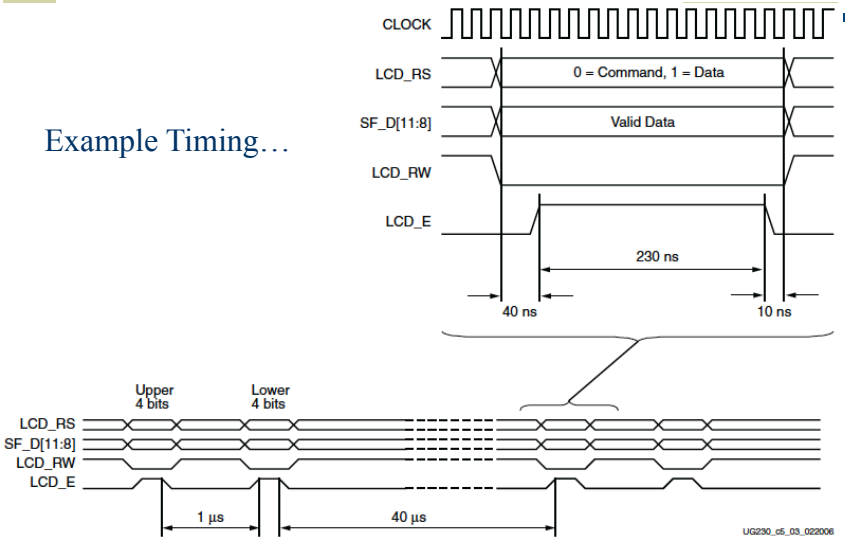


Figure 5-6: **Character LCD Interface Timing**

HD 44780-compatable LCDs

Example Startup Timing…

Power-On Initialization

The initialization sequence first establishes that the FPGA application wishes to use the four-bit data interface to the LCD as follows:

- Wait 15 ms or longer, although the display is generally ready when the FPGA finishes configuration. The 15 ms interval is 750,000 clock cycles at 50 MHz.
- Write SF_D<11:8> = 0x3, pulse LCD_E High for 12 clock cycles.
- Wait 4.1 ms or longer, which is 205,000 clock cycles at 50 MHz.
- Write SF_D<11:8> = 0x3, pulse LCD_E High for 12 clock cycles.
- Wait 100 µs or longer, which is 5,000 clock cycles at 50 MHz.
- Write SF_D<11:8> = 0x3, pulse LCD_E High for 12 clock cycles.
- Wait 40 µs or longer, which is 2,000 clock cycles at 50 MHz.
- Write SF_D<11:8> = 0x2, pulse LCD_E High for 12 clock cycles.
- Wait 40 µs or longer, which is 2,000 clock cycles at 50 MHz.

HD 44780-compatable LCDs

Writing Data to the Display

To write data to the display, specify the start address, followed by one or more data values.

Before writing any data, issue a Set DD RAM Address command to specify the initial 7-bit address in the DD RAM. See Figure 5-3 for DD RAM locations.

Write data to the display using a Write Data to CG RAM or DD RAM command. The 8-bit data value represents the look-up address into the CG ROM or CG RAM, shown in Figure 5-4. The stored bitmap in the CG ROM or CG RAM drives the 5 x 8 dot matrix to represent the associated character.

If the address counter is configured to auto-increment, as described earlier, the application can sequentially write multiple character codes and each character is automatically stored and displayed in the next available location.

Continuing to write characters, however, eventually falls off the end of the first display line. The additional characters do not automatically appear on the second line because the DD RAM map is not consecutive from the first line to the second.

HD 44780-compatable LCDs

- Luckily you don't have to do all this from scratch!
  - These LCDs are everywhere
  - So, helpful people have put all this stuff into libraries for you
  - One comes with Arduino…
    - LiquidCrystal library

LiquidCrystal library

- Comes built in to Arduino environment
  - LiquidCrystal(), begin(), clear(), home(), setCursor(), write(), print(), cursor(), noCursor(), blink(), noBlink(), display(), noDisplay(), scrollDisplayLeft(), scrollDisplayRight(), autoscroll(), noAutoscroll(), leftToRight(), rightToLeft(), createChar()

LiquidCrystal example

```
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //RS, En, D4, D5, D6, D7

void setup() {
  lcd.begin(16, 2); // set up the LCD's number of columns and rows
  lcd.print("hello, world!");  // Print a message to the LCD.
}
void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

LiquidCrystal issues

- Barebones LCD displays are pretty cheap
  - 16-char, 2-line LCD from Sparkfun: $14.95
  - Even cheaper from surplus places
- But, uses a lot of digital pins…
- Solution: Use a helper chip to convert to a serial interface
  - Uses only one wire (plus vdd and gnd)

Serial LCD

- Serial protocol
  - Only one TX wire
  - Takes longer to upload data to the LCD
  - But, still happens very fast to our eyes…
  - More expensive…
    - Sparkfun: $24.95

SparkSoftLCD Serial LCD Library

- Download from the "Playground"
  - Print(), begin(), clear(), backlight(), enable(), scroll(), cursor(), cursorTo(), moveCursor(), sendControl()

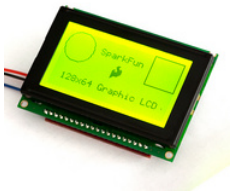
SparkSoftLCD Example

```
#include "SparkSoftLCD.h"
#define LCD_TX   2
#define LCD_WIDTH 16

SparkSoftLCD lcd = SparkSoftLCD(LCD_TX, LCD_WIDTH);
void setup()  {
   pinMode(LCD_TX, OUTPUT);
    lcd.begin(9600);  // leave at 9600 unless you change the baud rate of the
lcd   |
   lcd.clear();
}
void loop() {
  lcd.print("Testing 1 2 3");
  delay(1000);
  lcd.clear();
  float x = 5.14; // print a floating point number with two decimals
  lcd.print(x, 2);
  delay(1000);
  lcd.clear();
}
```

Other LCDs

- Graphic LCDs
  - Pixel programmable rather than characters
  - 128x64 for example
  - Harder to use
  - ~$35



- Color LCD displays
  - E.g. OLED 128x128 ~$60
  - E.g. Nokia 6100 ~$35

LCD Conclusions

- Standard 16-char 2-line LCDs are easy to use
  - "bare bones" with Hitachi HD44780 controller
    - 7 or 11 pins needed – built in LiquidCrystal library
  - Serial-enabled LCD
    - Only one pin needed
    - SparkSoftLCD library in the playground
- Graphic LCDs – a little harder to use…
- Lots of other surplus versions
  - Some easier to use than others…