This lesson will teach you how to use the LCD display.

**Note:** Some LCD displays may require soldering. If your LCD requires soldering, before using it, you will need to solder the display to the connector pins.

We will be using a provided Arduino library “LiquidCrystal” that allows you to control LCD Displays that are compatible the Hitachi HD44780 driver.

Refer to the complete and original documentation here: [https://www.arduino.cc/en/Tutorial/LiquidCrystal](https://www.arduino.cc/en/Tutorial/LiquidCrystal)

**The Circuit**

The following figures and tables illustrate the usage of the LCD display on a breadboard.
### LCD Display Pin Connections

<table>
<thead>
<tr>
<th>LCD PIN #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>11-14</th>
<th>15-16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>GND</td>
<td>5 Volts</td>
<td>Potentiometer (10KΩ) thru. to GND</td>
<td>Pin 12</td>
<td>GND</td>
<td>Pin 11</td>
<td>Pins 5-2</td>
<td>Vcc-GND</td>
</tr>
</tbody>
</table>

The circuit:
- LCD RS pin to digital pin 12
- LCD Enable pin to digital pin 11
- LCD D4 pin to digital pin 5
- LCD D5 pin to digital pin 4
- LCD D6 pin to digital pin 3
- LCD D7 pin to digital pin 2
- LCD R/W pin to ground
- LCD VSS pin to ground
- LCD VCC pin to 5V
- 10K potentiometer:
  - ends to +5V and ground
  - wiper to LCD VO pin (pin 3)

### Programming

The LiquidCrystal library is available with Arduino but it needs to be tweaked in order to behave properly with the Intel Galileo.

Instructions provided by Mikal.Hart on communites.intel.com shows us how to properly update the LiquidCrystal Library with the following steps:

1. Download the latest archive at Releases · mikalhart/galileo-LiquidCrystal · GitHub
2. Unzip to the Galileo-specific library area at ../hardware/arduino/x86/libraries/LiquidCrystal. (Make sure to rename the root folder to LiquidCrystal, because Arduino doesn't like dashes in library names.)
3. Exit the Arduino IDE software and restart it.

Original Documentation can be found here: [http://intel.ly/1LsB0d4](http://intel.ly/1LsB0d4)

Now we are ready to upload an example sketch to our Galileo!
Instructions:

1. Navigate to File -> Examples -> LiquidCrystal -> HelloWorld

2. Select Verify, Then Upload

Two things will now appear on the LCD Screen. On the first row you will see “hello, world!” printed and on the seconds row, the number of seconds since reset.

Reading through the code, you’ll see that there are some lines of code that are important in order to initialize the LCD.

```c
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
    // set up the LCD's number of columns and rows:
    lcd.begin(16, 2);
}
```

- `#include <LiquidCrystal.h>` - Includes library
- `LiquidCrystal lcd(12, 11, 5, 4, 3, 2);` - declares lcd according to its pins.
- `Lcd.begin(16, 2);` - Sets the number of rows and columns you want to use.

We are now ready to begin our exercises.
Exercise

Video Demonstration: [https://youtu.be/epkYVeYqNn8](https://youtu.be/epkYVeYqNn8)

1. Buttons are connected to A0 and A1. The LCD display is setup and initially displaying 0. Pressing A0 increments the display (stopping at 9). Pressing A1 decrements the display (stopping at 0). If both buttons are depressed (even if not initially simultaneously), the display resets to 0. Be sure to set a logical threshold for both buttons being pressed, so that both buttons can be used independently. *If a button is held, then the display continues to increment (or decrement) at a rate of once per second.* Use a synchronous state machine captured in C. Writing with `lcd.print()` should only occur once pre increment/decrement/reset in order to avoid repeated writes, and maximize LCD readability.

Challenge Problem

2. Extend Part2 of Lesson 3’s light game to maintain a score on the LCD display. The initial score is 5. Each time the user presses the button at the right time, the score increments. Each time the user fails, the score decrements. When reaching 9, show victory somehow. Receiving a score of 0 causes the player to lose the game. Show failure somehow.