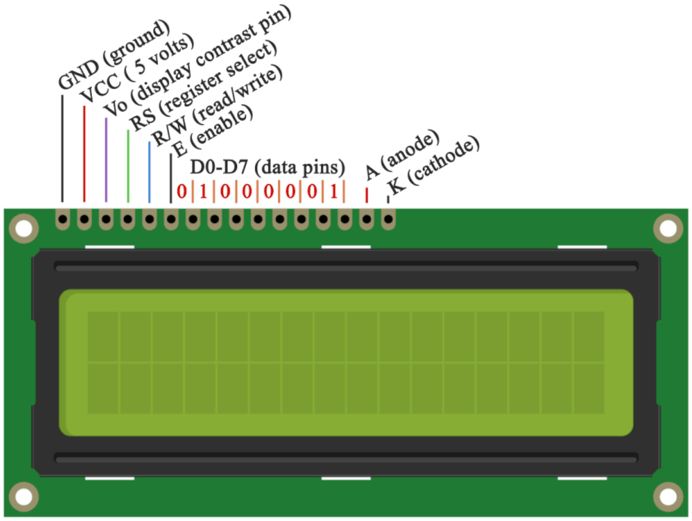
Introduction to Arduino

Liquid Crystal Display

Note: Marks will be taken off for incorrect answers or messy breadboarding

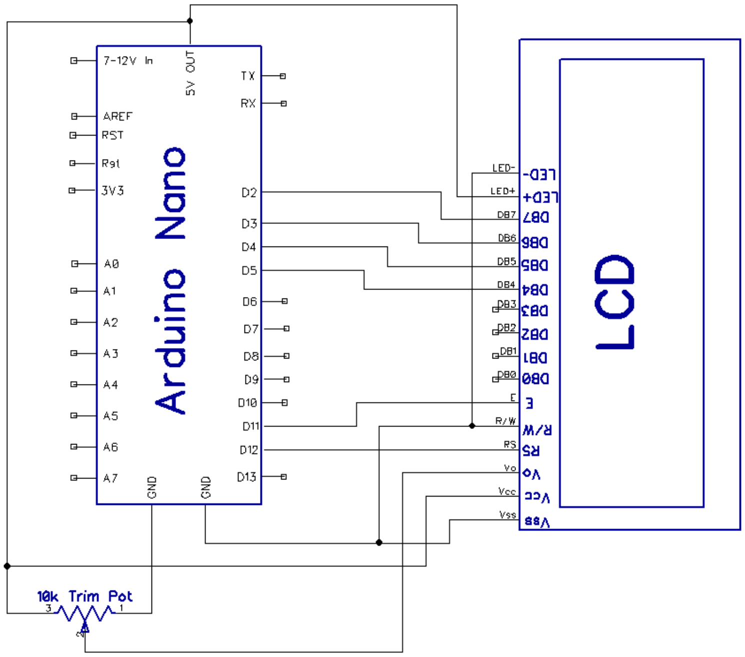
Introduction and Wiring

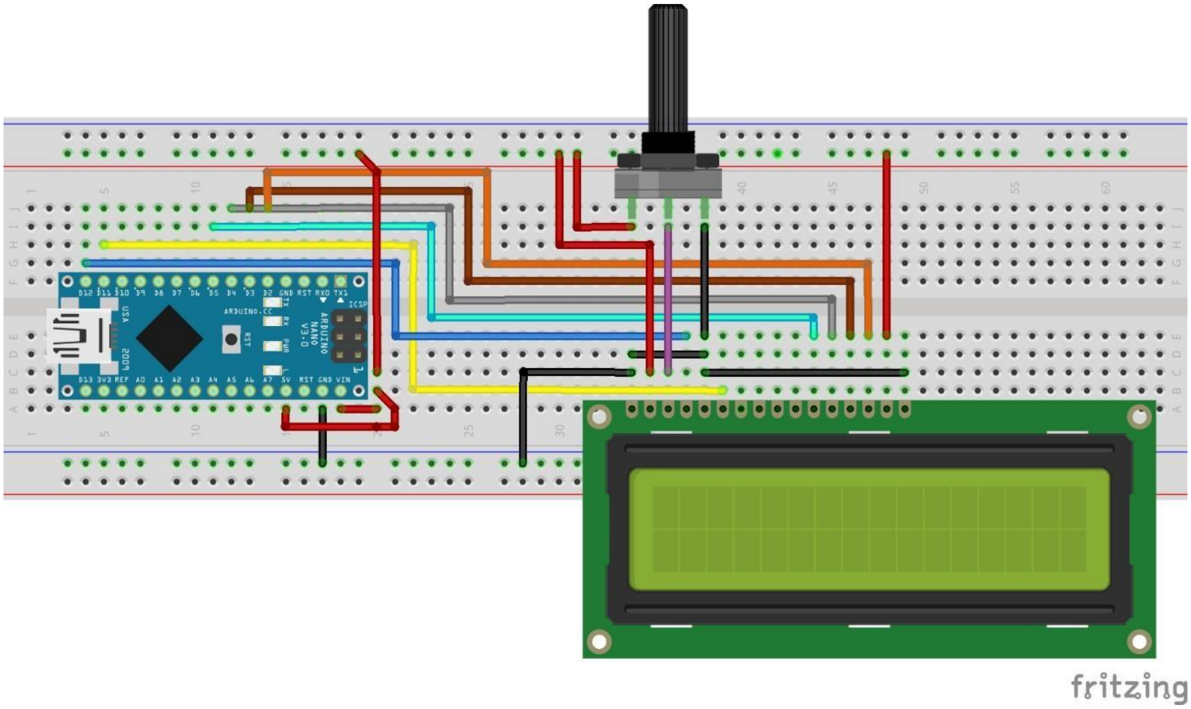
You are going to connect a 'Liquid Crystal Display', or 'LCD' to your Arduino Nano. The LCD uses many pins for its operation (see details of these below). For the best chance of success, it is VERY important, not only to conect these correctly to your Arduino, but also to that the wiring be neat and the wire ends be inserted deep enough into the breadboard to ensure a good connection. One bad connection can be diffiicult to find and will result in your LCD not working!



1. Use the schematic diagram, and/or the pictorial diagram that follows on the next page, to connect your Liquid Crystal Display (LCD) to your Nano, which you will then use to display data back to you.

2. Go to the official Arduino website, under the "DOCUMENTATION" tab, click on 'LIBRARIES EXAMPLES' and scroll down to "LIQUID CRYSTAL LIBRARY" and select the "Hello World" tutorial. Read the content and **understand** the code.





3. In the Arduino programming software, open the source code (Files > Examples > LiquidCrystal > HelloWorld) and upload it to your Nano.

4. You should now have "Hello World" and the seconds counting on the LCD. You may need to adjust the potentiometer to get the contrast set correctly so you can see the text. (if you don't see "hello world" you need to re-check your wiring and the quality of your connections).

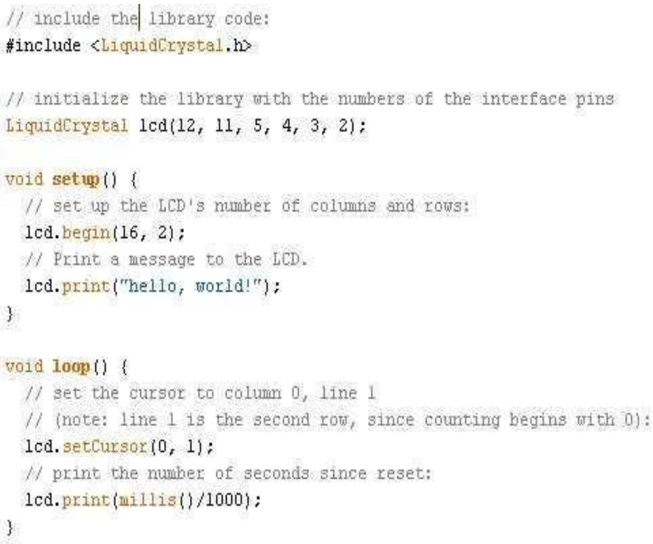
Show the teacher the working lab and code.

LCD Lab Cont'd – Understanding Libraries…

1. In the 'Hello World' code you will see something new (shown on the right). Notice the following line of code BEFORE the 'void setup ()' function…

   #include <LiquidCrystal.h>

   This is a '**library**' that needs to be **included** in the program for the LCD to work. A library is a set of separate code written specifically to work with certain devices, of which there are many (ie, LCD, Ultrasonic Module, Transmitter/Receiver, Bluetooth Module, etc).



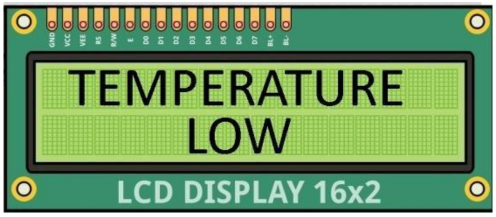
```
// include the library code:
#include <LiquidCrystal.h>

// 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);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

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);
}
```

Including ‘libraries’ removes a lot of the heavy lifting that would otherwise be required to communicate with different devices like your LCD screen. Once included, ‘pre-programmed commands’ can be used to control the device and these commands will be recognized by the programming software. Many libraries exist some are included with the Arduino programming environment when installed, like the ‘LiquidCrystal’ library, others can be downloaded and installed as/when needed. You will find these libraries and library management under the tab ‘Sketch’ in the Arduino programming environment.

Follow this link to view the commands/functions that the ‘LiquidCrystal’ library makes possible

LCD Programming Goal…

1. Remember Lab #3 where you had a single LED turn on based on the position/value input to pin A0 from a potentiometer? This task is similar only you won’t have an LED turn when a value range is detected by the Arduino, but words will be printed to the LCD.



   You will have achieved the goal when “Temperature” is displayed on the top line of the LCD and then “LOW, MED, HIGH” on the bottom line based on the following readings of the potentiometer” AND these words are centered on the screen.

   - *LOW* - for anything less than a reading of 250
   - *MED* – for anything between a reading of 250 and 750
   - *HIGH* – for anything over 750.

Code Development Procedure:

You will be combining the concepts learned in Lab 3 and borrowing from its code to create the LCD code:

- Open both lab #3’s code and the liquid crystal ‘Hello World Code. Arrange both side by side on the screen.
- Save the ‘Hello World’ LCD example code as Lab 4 LCD (under ‘File’ > ‘Save As…’)
- Examine both codes and understand what each is doing, then complete the following:
    - As previously stated, you will need ‘to write’ three functions that will print three different conditions to the LCD screen. Look at how you constructed the ‘functions’ in Lab 3 that turned on a single LED when a specified value was read from the potentiometer

- In the ‘Lab 4 LCD’ code delete all syntax between the curly brackets in the void loop, in other words, clear the loop
- Copy from lab #3 a single function that turned on a single LED for a specified value range and paste this into Lab 4’s loop.
- Alter this function in the Lab 4 code to print ‘Temperature Low’ (on two lines) to the LCD screen. Consider the following when writing this code (you will need to include all three of these commands:

  lcd.clear(); – *command to clear previous words*

  lcd.setCursor(col, row); – *places the start of the text, with the first position being zero*

  lcd.print(“words here”); – *prints to the screen the words between* quotes

- For the first line of code in the loop, write a statement that will read the input value from the potentiometer on A0 (*check lab 3 for how this was done*)
- You don’t have anything connected to the A0 pin on your Nano. You now need to add a second potentiometer and connect its output to A0 as was done in Lab 3 (*see Lab 3 for how to do this*)
- In Lab 4’s setup function, include a line of code that will initiate serial communications (*check Lab 3*)
- In Lab 4’s loop, just before the closing of the loop, include a line of code that will print the input value sensed at A0 to the serial monitor (*check lab 3’s code*)
- Include a 500 ms delay as the first line of code in your print function. Doing this will reduce flicker of the LCD screen that would result if too many writes per second were sent to it
- Upload your code. Once it is working, write two more functions (copy, paste and alter the working function), to display the three conditions, ‘Low’/’Med’/’High’

Show the teacher the working lab and code.