I And M Electronics

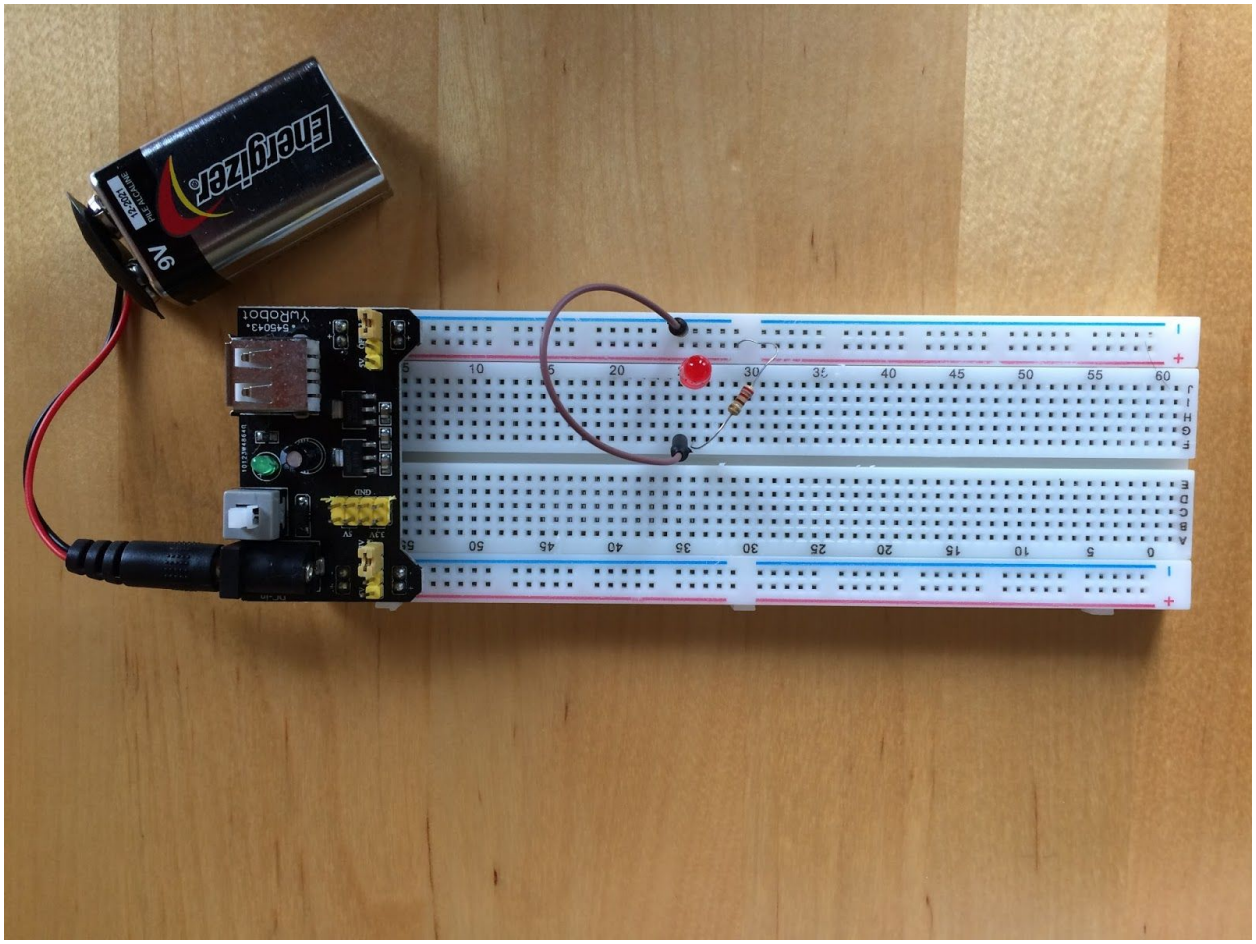
Experiment 1 - Let there be light.

Parts:

- Breadboard power supply
- 9 Volt Battery
- 9 Volt Battery Connector
- LED
- 220OHM Resistor (red,red,brown,gold or silver)
- Jumper Wire

Build:

- Connect power supply to breadboard.
- Set one side to 3volt.
- Put resistor between positive strip and column 26.
- Put LED long leg in column 26 so it is connected to resistor and positive.
- Put jumper between column 25 and negative strip.
- Connect battery to power supply with battery connector.



Test:

- Turn power supply on.
- LED should light up.
- Turn power supply off.
- Switch power supply to 5 volt.
- Turn power supply on.
- Is the LED brighter now?

Troubleshooting:

- LED backwards? Remember LED are diodes current can only flow one way. The long leg needs to be on the positive power side.
- Is each connection in the correct row (components) or column (power)?

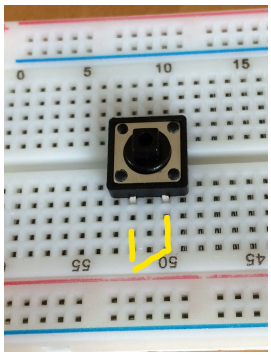
Experiment 2 - Let me control the light.

Parts:

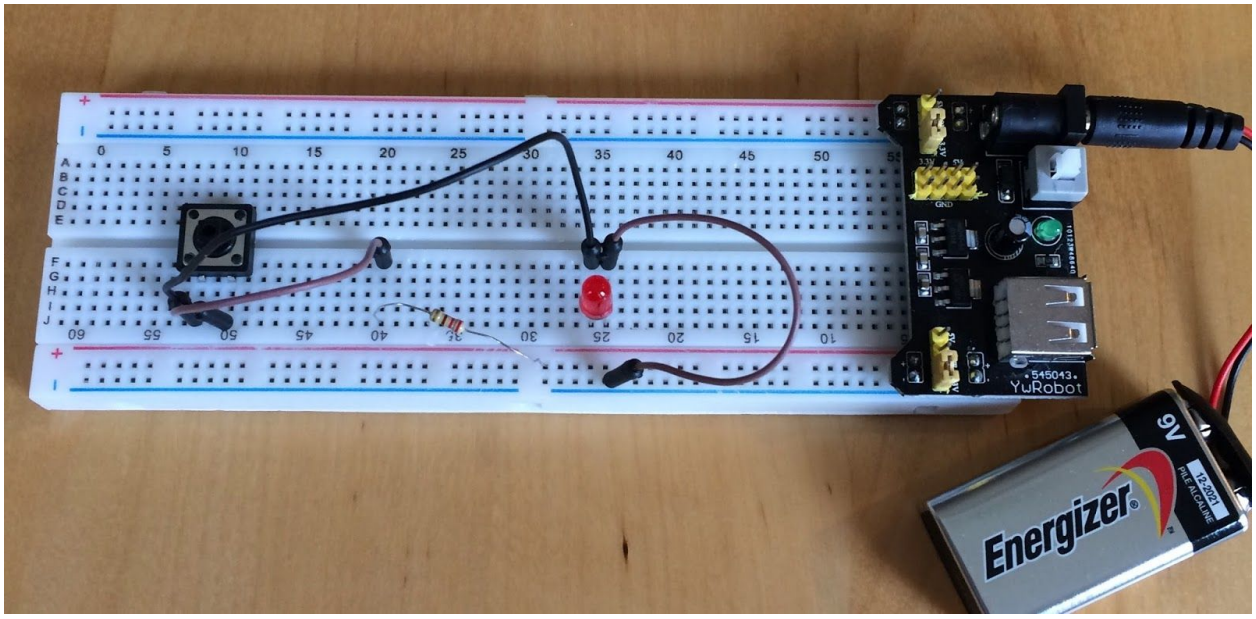
- Same as Experiment 1
- Momentary Switch
- 2 jumper wires

Build:

- Add a momentary button to the breadboard. The momentary switch connects the two connects on one sided together when the button is pressed. This means you should place it in way where the connections on one side of the switch are in separate columns. In the image you can see one connection is in column 50 and on is in column 52. When the button is pressed column 50 and 52 will be connected together.



- Move one side of the resistor from column 26 and place it in a open column like 40.
- Place a jumper from that will connect the resistor to one leg of the switch and the resistor (connect column 40 and column 50).
- Place a jumper between the other switch connection and the positive leg of the LED (connect column 52 and column 26).



Test:

- Turn power supply on.
- Is the LED on? If it is you made a mistake.
- Press the momentary button.
- Is the LED on when you hold the button down and off when you release the button?
- The LED should turn on when you press the button and turn off when you release the button.
- Can you make it do Morse code?

Troubleshooting:

- Check all columns are correct.
- Does the momentary switch have the connections that are on the same side in separate columns.

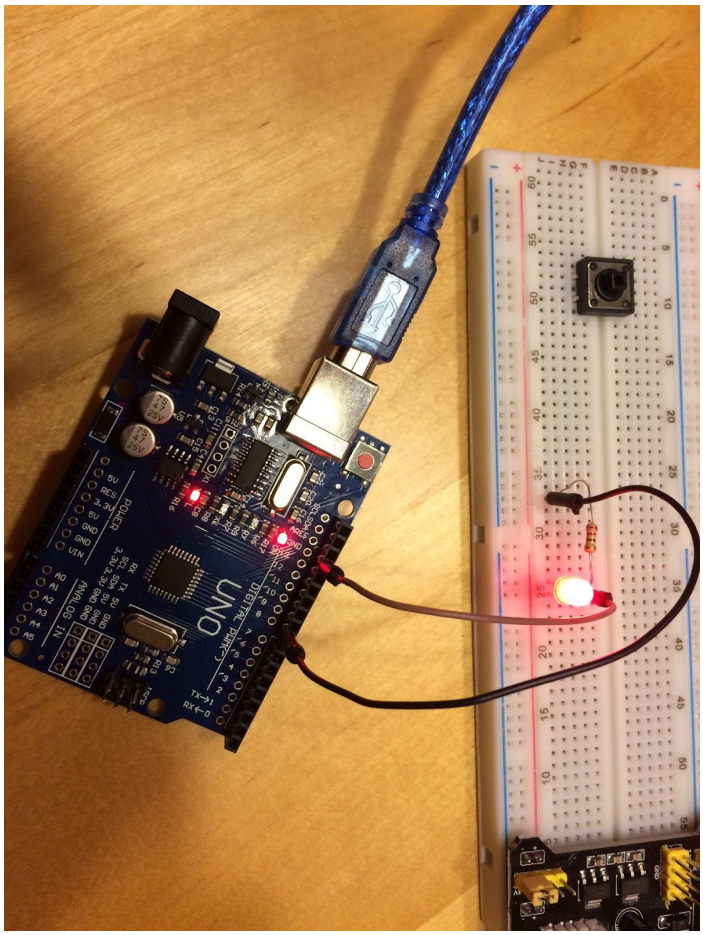
Experiment 3 - Blink

Parts:

- Arduino UNO
- LED
- 2 Jumpers
- Computer with Arduino Integrated Development Environment (IDE)
- USB A Cable

Build:

- Place an LED on the breadboard (Column 25 - and 26 +).
- Place an 220 OHM resistor between the positive LED leg and another column (33).
- Place a jumper between Arduino digital output 8 and the resistor (column 33).
- Place a jumper between Arduino ground(GND) connection and the LED negative leg (column 25).



- Kick mom or dad off of FaceWhatever and type the following program into the Arduino IDE:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(8, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(8, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(8, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

Test:

- Upload run the program on the Arduino.
- The light should blink about once per second.
- You see in the code that we are turning the LED connected to digital output 8 high then delaying 1000 and then turning it to low. The 1000 is how many cycles we are waiting. Since the clock on this CPU ticks 1000 times per second a 1000 delay is 1 second.
- Arduino's are designed to run all the code in the setup first and then run all the code in loop every cycle of the CPU. So the first thing this program does is pinMode(8,OUTPUT). This tells the Arduino to set digital connection 8 to be an output connection. The next instruction to get executed is digitalWrite(8,HIGH). This tells the Arduino to turn on power to connection number 8. The next instruction delay(1000) tells the Arduino to do nothing for the next 1000 cycles. The next instruction digitalWrite(8,LOW) tells the Arduino to turn off the power to connection number.
- Now let's play with the program. What happens if you change the first delay to 300 and the second to 300?
- How about delay 1 to 1500 and delay 2 to 300?
- Could you program it to repeat SOS in Morse Code?

Troubleshooting:

- Are there lights on the Arduino? This experiment uses the computer to power the Arduino. When you plug the Arduino to the computer it should light up the on LED and L LED.
- Check all jumpers are in the correct columns.
- Check the jumpers are in the correct digital output connection.

Experiment 4 - What's my name.

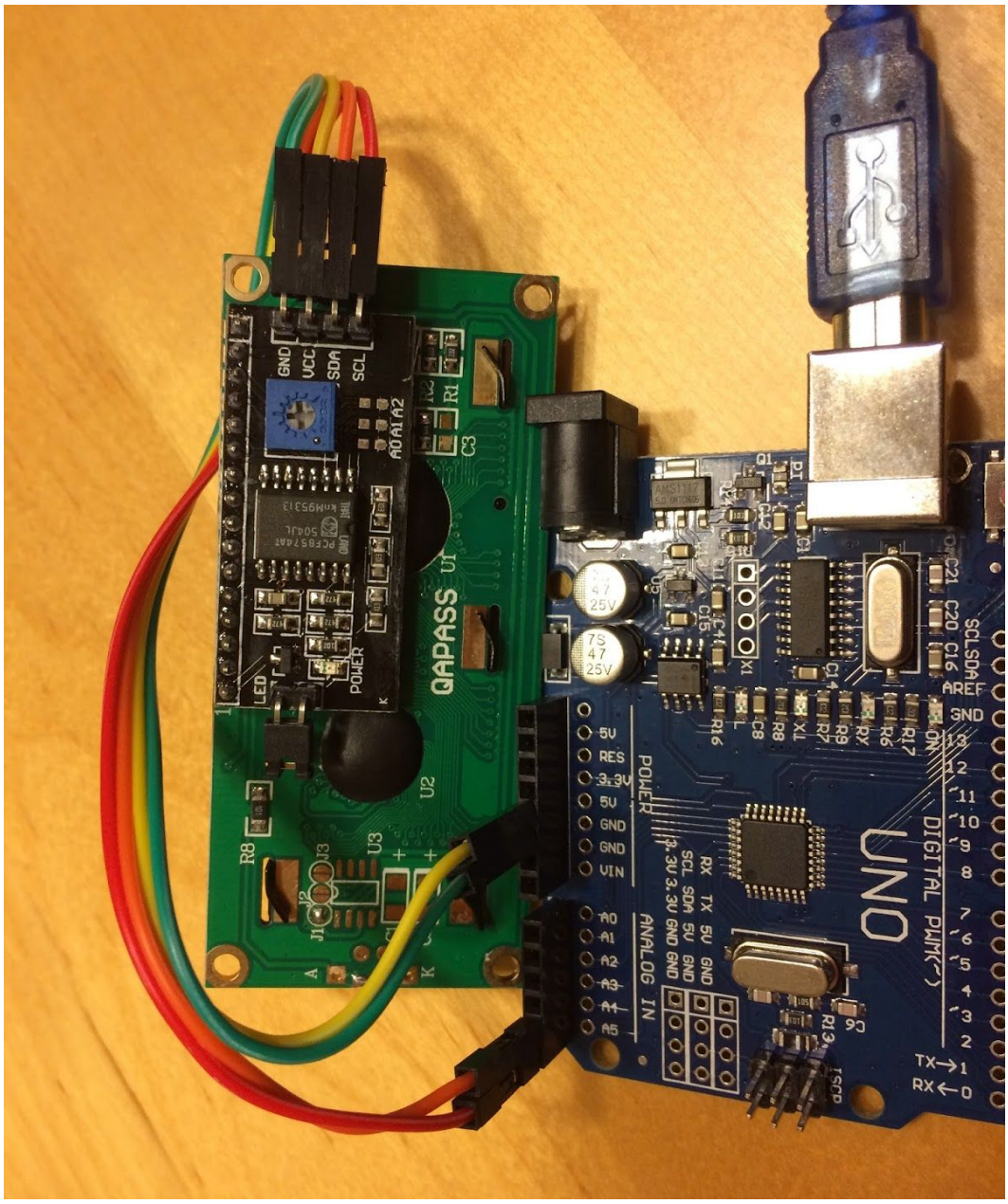
Parts:

- Arduino UNO
- 4 Jumpers

- LCD With I2C Connected
- Computer with Arduino Integrated Development Environment (IDE)
- USB A Cable

Build:

- Connect a different color jumper to each of the four connections on the LCD I2C board. They are marked GND, VCC, SDA and SCL.
- Connect the other end of these jumpers to the Arduino. The GND jumper should connect to the ground (GND) on the Arduino. The VCC jumper should connect to the 5v power connection on the Arduino. Connect the SDA jumper to analog input number 4. Connect the SCL jumper to analog input number 5.



- In the Arduino IDE enter the following program. Feel free to replace your name for What is My Name in the program.

```
#include <Wire.h> // Include code for talking analog
#include <LiquidCrystal_I2C.h> //Include code for controlling LCD
LiquidCrystal_I2C lcd(0x3F,16,2); //Set the LCD
void setup()
{

  lcd.init(); // initialize the lcd
  // Print a message to the LCD.
  lcd.backlight(); //Turn on Backlight
  lcd.setCursor(0,0); // go to column 0 row 0
  lcd.print("What is"); //Type What is on the display
  lcd.setCursor(3,1); // go to column 3 row 1
  lcd.print("My Name"); //Type My Name on the display
}

void loop() // We do not do anything after the setup.  Just show the text
{
}
```

Test:

- Upload run the program on the Arduino.
- What is your name should be displayed (or whatever you changed it to).
- Try changing the text in the program to different values. What happens if the text is longer than fits on a line.
- On the back of the LCD there is a blue box with a little screw dial in the middle. Use a screwdriver to turn the screw while the display is on. What happens on the display?
- The blue box is a potentiometer. It is like a resistor. When you turn it the screw one way it increases the resistance and the turning the it other other way lowers it.

Troubleshooting:

- Are there lights on the Arduino? This experiment uses the computer to power the Arduino. When you plug the Arduino to the computer it should light up the on LED and L LED.
- Check all jumpers are in the correct columns.
- Check the jumpers are in the correct analog output connections.
- Try turning the contrast up or down on the blue potentiometer on the LCD.

Experiment 5 - I know your name.

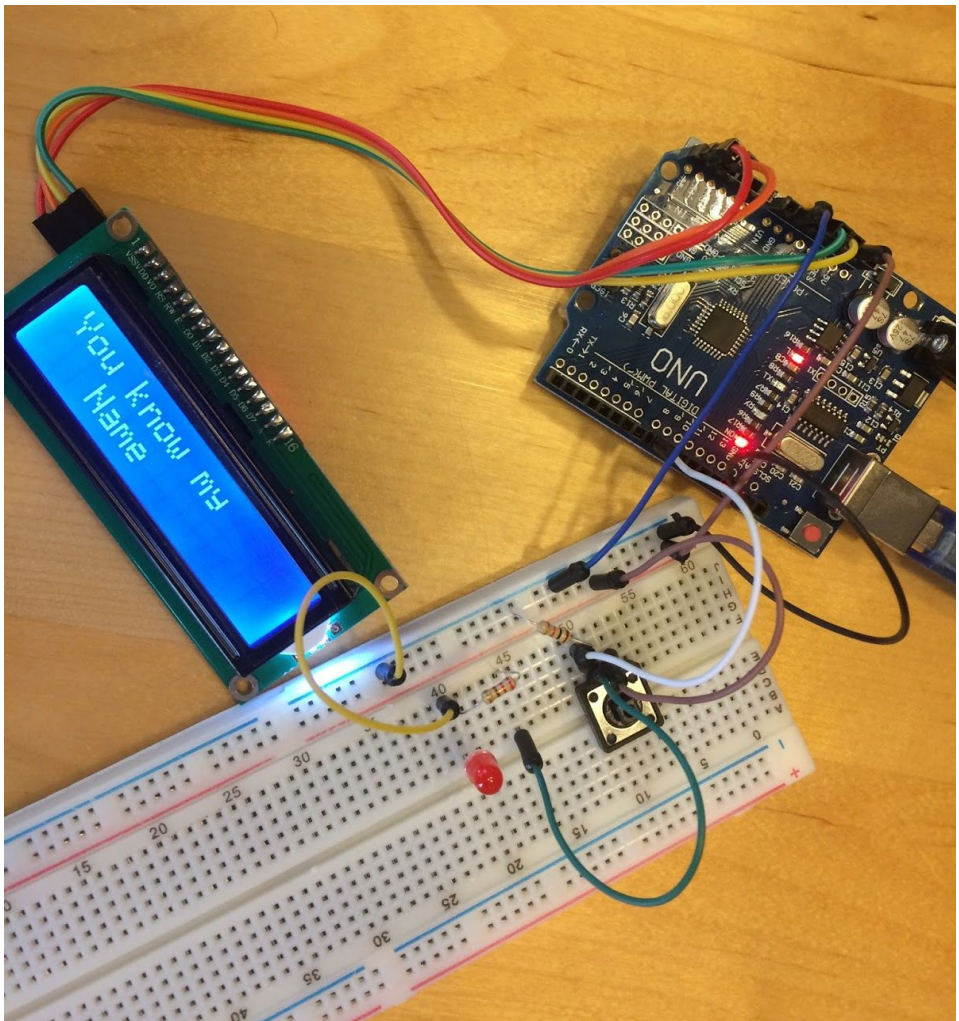
Parts:

- Arduino UNO
- 11 Jumpers
- LED
- Momentary switch

- 220 OHM resistor
- 1K OHM resistor
- Breadboard
- LCD With I2C Connected
- Computer with Arduino Integrated Development Environment (IDE)
- USB A Cable

Build:

- Connect a jumper from a 5V pin on the Arduino to the positive strip on the breadboard.
- Connect a jumper from a ground GND in on the Arduino to the negative strip on the breadboard
- Connect the LCD the same as you did in Example 3.
- Connect a leg of the momentary switch to the positive strip. We will call this leg 1. (column 52 to + strip)
- Connect the other leg of the momentary switch to digital pin number 8. We will call this leg 2. (column 50 to Arduino digital 8)
- Connect the 1K OHM resistor from momentary switch leg 2 to the negative strip. (column 50 to - strip)
- Connect a 220 OHM resistor from the positive leg of the LED to an open column. (column 41 to column 44)
- Connect 220 OHM resistor to leg 2 of the momentary switch. (column 44 to column 50)
- Connect the negative leg of the LED to the negative strip. (column 40 to - strip)
- Connect a jumper from ground on the digital side of the Arduino to negative strip.



- In the Arduino IDE enter the following program. Feel free to replace your name for You know my name in the program.

```
#include <Wire.h> // Include code for talking analog
#include <LiquidCrystal_I2C.h> //Include code for controlling LCD

int state = HIGH;   // the current state of the output pin
int reading;        // the current reading from the input pin
int previous = LOW; // the previous reading from the input pin

// the follow variables are long's because the time, measured in milliseconds,
// will quickly become a bigger number than can be stored in an int.
long time = 0;      // the last time the output pin was toggled
long debounce = 200;   // the debounce time, increase if the output flickers

LiquidCrystal_I2C lcd(0x3F,16,2); //Set the LCD
void setup()
{
  pinMode(8, INPUT); //Set pin 8 to look for a button press

  lcd.init(); // initialize the lcd
  lcd.backlight(); //Turn on Backlight
  lcd.setCursor(0,0); // go to column 0 row 0
  lcd.print("What is"); //Type What is on the display
  lcd.setCursor(3,1); // go to column 3 row 1
  lcd.print("My Name"); //Type My Name on the display
  state = LOW;
}

void loop() // We just keep looping waiting for someone to push the button.
{
  reading = digitalRead(8);
  // if the input just went from LOW and HIGH and we've waited long enough
  // to ignore any noise on the circuit, toggle the output pin and remember
  // the time
  if (reading == HIGH && previous == LOW && millis() - time > debounce) {
       if (state == HIGH){
       state = LOW;
       lcd.clear();
       lcd.setCursor(0,0); // go to column 0 row 0
       lcd.print("What is"); //Type What is on the display
       lcd.setCursor(3,1); // go to column 3 row 1
       lcd.print("My Name"); //Type My Name on the display
       digitalWrite(8,LOW);
       }
       else {
       state = HIGH;
```

```
      lcd.clear();
      lcd.print("You know my"); //Type What is on the display
      lcd.setCursor(3,1); // go to column 3 row 1
      lcd.print("Name"); //Type My Name on the display
      digitalWrite(8,HIGH);
      }
      time = millis();
  }
 previous = reading;
}
```

Test:

- Upload run the program on the Arduino.
- What is my name should be displayed.
- Press the momentary switch. Did the display change? Did the LED light up?
- Press the momentary switch again. Did it go back to what it was?
- Can you identify the code that is executing each time you press the button? Can you see why the code executing switches each time you press the button?
- Could you add more steps and states to the code?
- What happens if you disconnect the Arduino from the computer? Why does the Arduino need the computer to run? What happens if you connect your nine volt battery to the Arduino? Does the Arduino remember what to do?

Troubleshooting:

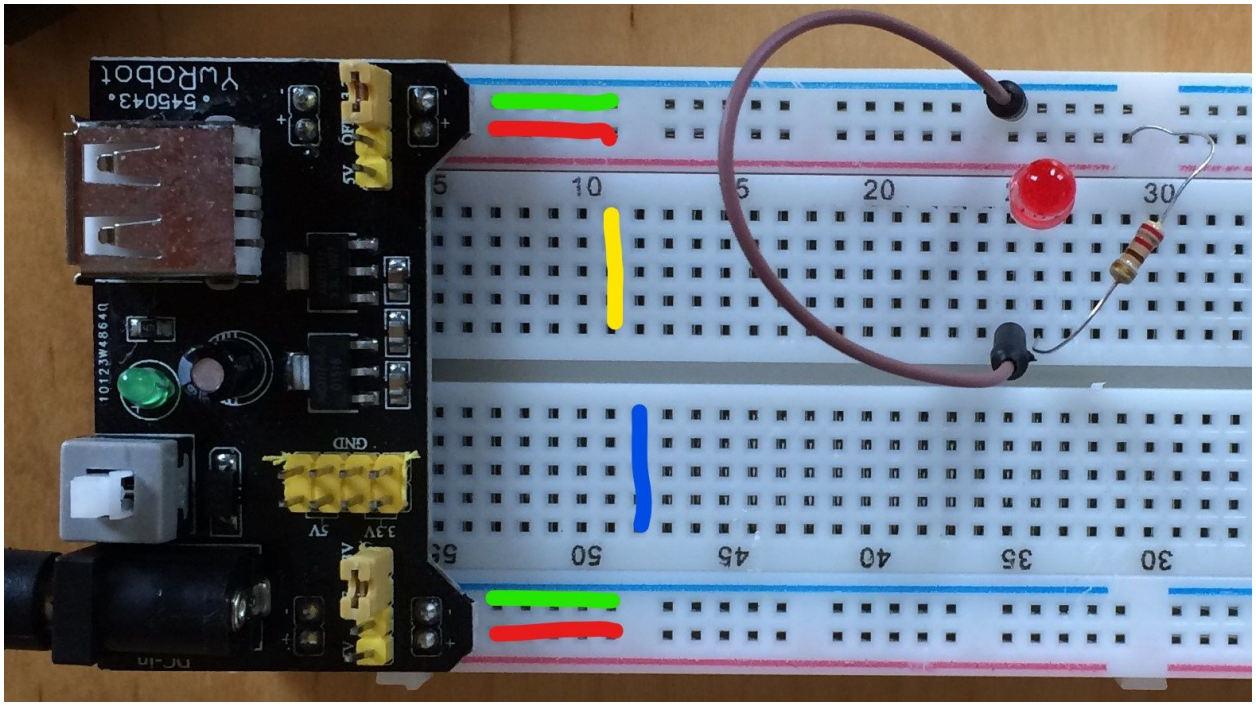
- Are there lights on the Arduino? This experiment uses the computer to power the Arduino. When you plug the Arduino to the computer it should light up the on LED and L LED.
- Check all jumpers are in the correct columns.
- Check the jumpers are in the correct analog and digital output/input connections.
- Try turning the contrast up or down on the blue potentiometer on the LCD.

Experiment 6 - You are on your own. Here are some ideas though.

- Create a game where you connect 2 momentary switches to the Arduino. Create a countdown time that shows the seconds remaining on the LCD. Once the seconds reach zero look for the digital input for both momentary switches. Display which momentary switch was pressed first. This could now be a game to see who is the fastest button presser.
- Same game as above but instead of a countdown timer randomly light up a LED the first person to press the button after the LED lights is the winner.
- Create a book with the Arduino and a LCD. Each time button is pressed display the next words of the book. Add another momentary switch so that you can go forward and backwards.
- Create a secret password keeper. With two momentary switches you could make a code like S1 S1 S1 S2 S1 S2. If this correct code is entered you display the secret.

Appendix

Breadboard Information:
The breadboard makes it easy to connect components together. It is comprised of columns of holes and rows of holes. The top and bottom rows indicated with + and - signs are connected across the rows. So all the holes on the + row are connected together and all the holes in the - row are connected. The middle of the breadboard is comprised of two rows of columns. Each column has 5 holes. Each of the five holes in a column are connected.

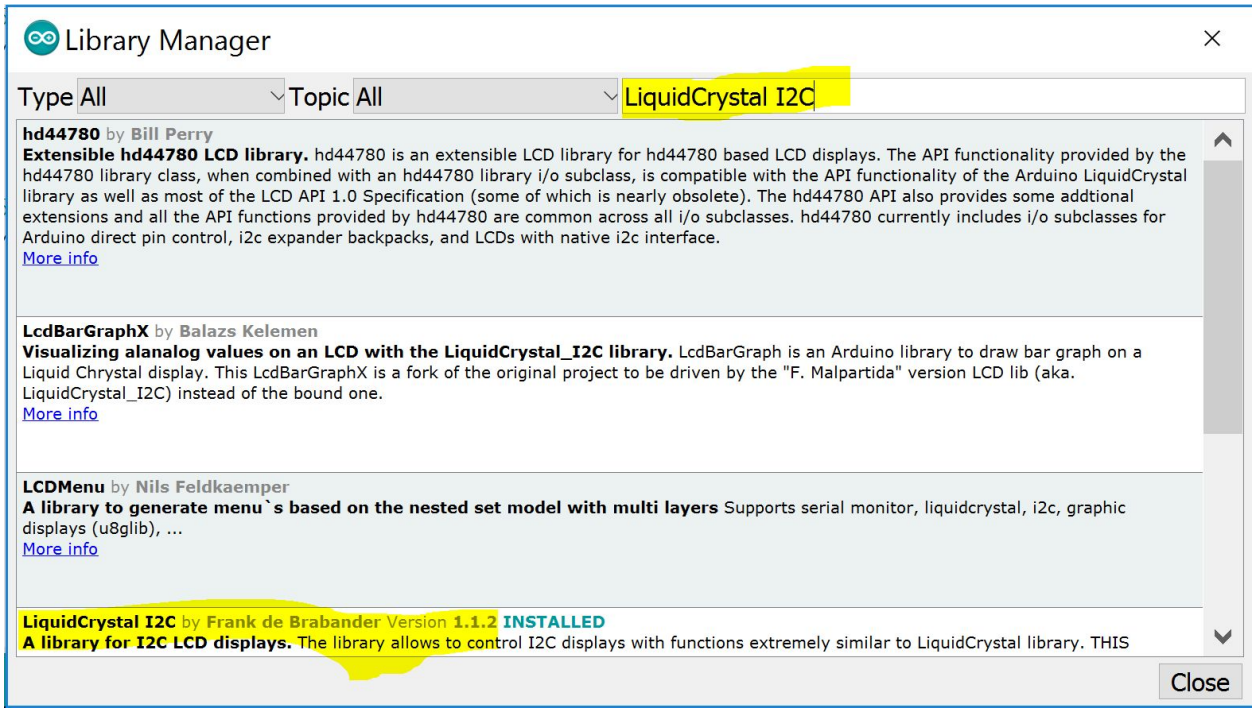


Examine the colored lines in the picture above. The green and red lines depict the power rows that are connected all the way across the breadboard on the top and bottom. The yellow line shows that all five holes in column 11 are connected. The blue line shows that all holes of column 48 are connected. The top columns are separated from the bottom columns(column 10 does not connect to column 50).

Arduino IDE Information:
You can download the Arduino IDE for free at: https://www.arduino.cc/en/Main/Software
You can also find other really cool things and help for projects there.
In order to get the LCD projects to work you must include the “LiquidCrystal I2C” Library. To install the library run the Arduino IDE:

Select the Sketch Menu:

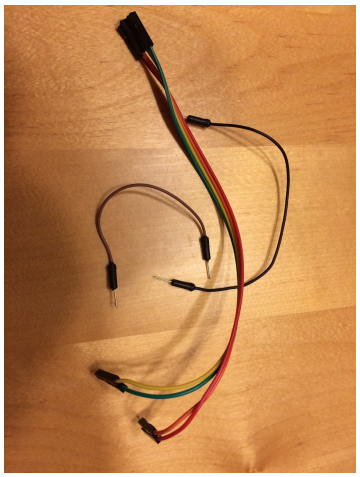
Next Click Include Library and Manage Libraries. This will open a new window:



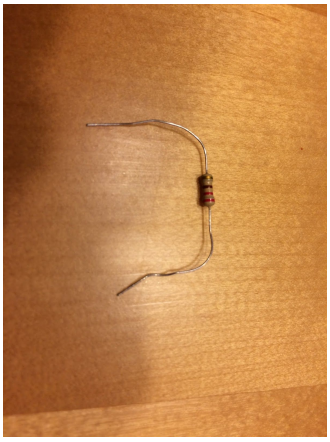
I used the LiquidCrystal I2C by Frank de Brabander.
When you click on that entry an install button will appear. Click that and it will install the library.

Glossary

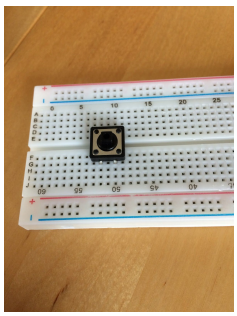
Jumper - Piece of wire for connect between holes on a circuit board or other components.



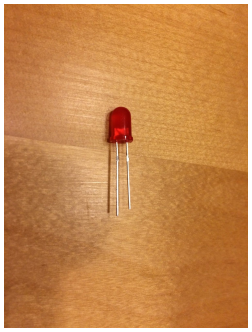
Resistor - An electrical component used to restrict the passage of electrical current. The color bands of the resistor indicate how much they reduce the current.



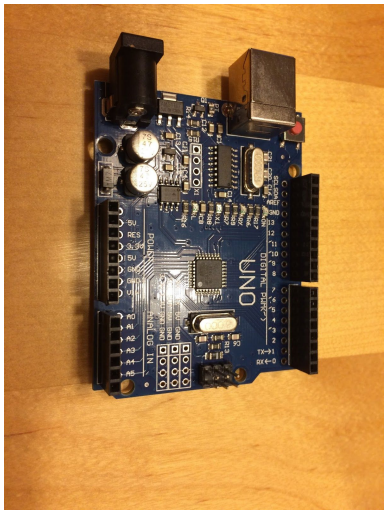
Momentary Switch - A switch that will open or close an electrical connection only while the user is pushing the button or actuator.



LED - Light Emitting Diode - An electrical component that allows direct current (DC) electricity to flow from one connection to another. It also produced light when a current is applied. The longer connection (leg) is the positive side.

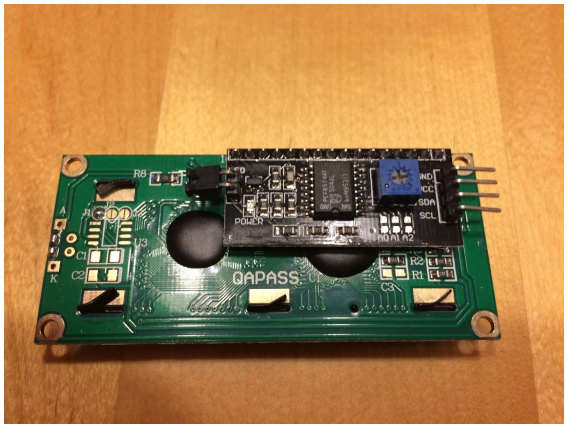


Arduino - A small electronic processing board that has a CPU and several input/output connections. It can be used to build really cool electronic prototypes.

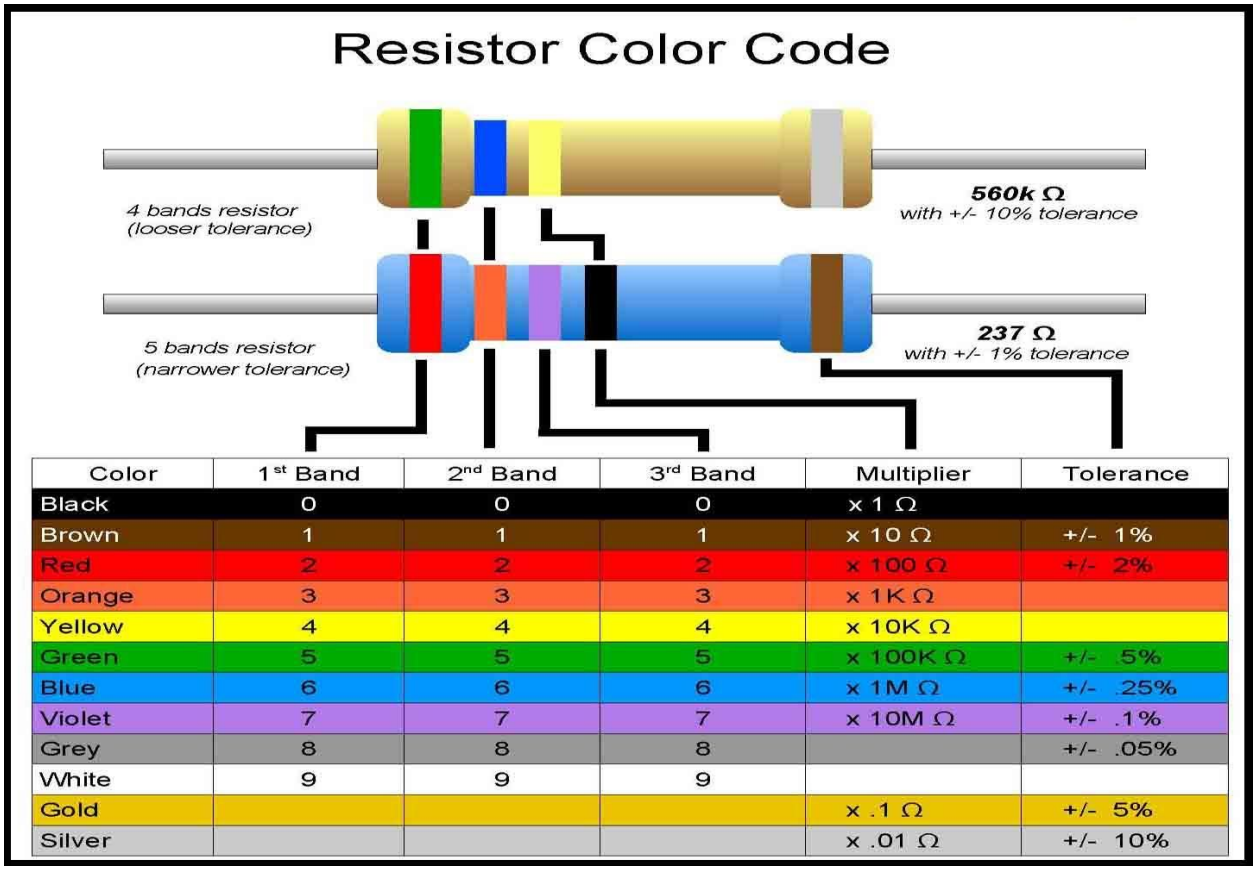


LCD - Liquid Crystal Display - An electrical component that can be used to display information.

LCD I2C Component - A small IC2 integrated circuit (IC) attached to an LCD that can interpret analog input/output and control the LCD. Using the IC2 circuit allows you to connect the LCD to an Arduino and only use two analog connections. Because using up all your connections for just the LCD makes an inventor sad. A sad inventor will build world destroying robots. World destroying robots are bad.



Resistor Color Code:



Color	1st Band	2nd Band	3rd Band	Multiplier	Tolerance
Black	0	0	0	x 1 Ω	
Brown	1	1	1	x 10 Ω	+/- 1%
Red	2	2	2	x 100 Ω	+/- 2%
Orange	3	3	3	x 1K Ω	
Yellow	4	4	4	x 10K Ω	
Green	5	5	5	x 100K Ω	+/- .5%
Blue	6	6	6	x 1M Ω	+/- .25%
Violet	7	7	7	x 10M Ω	+/- .1%
Grey	8	8	8		+/- .05%
White	9	9	9		
Gold				x .1 Ω	+/- 5%
Silver				x .01 Ω	+/- 10%