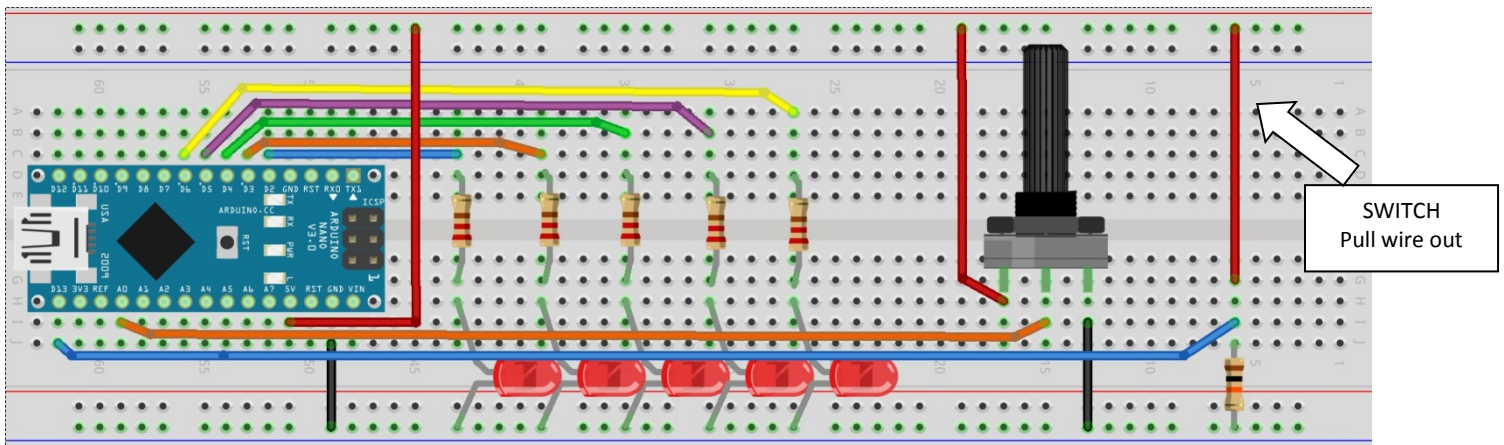
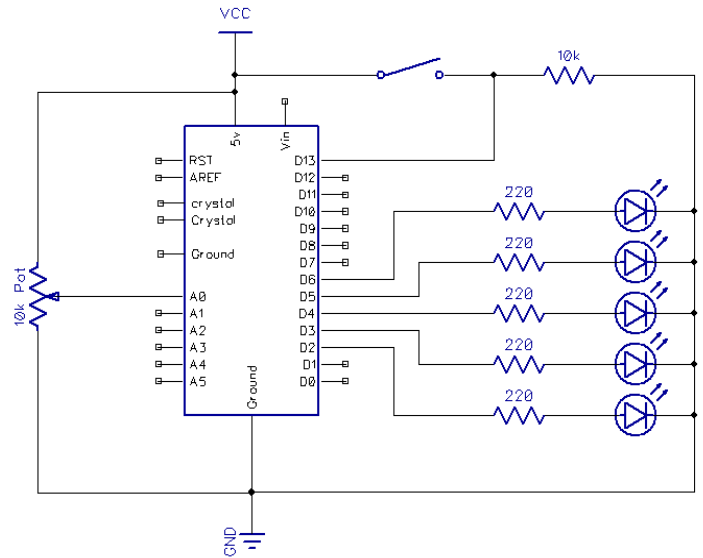


Introduction to Arduino

Digital/Analog Inputs - If Statements

1. You are going to add a switch and a potentiometer to your circuit (see schematic on right) and use them as inputs. Using the Arduino website, click on the “Learning” tab, and then scroll down under “Tutorials”, then select “Built In Examples” and find under 1. Basics both the “DigitalReadSerial” and “AnalogReadSerial” tutorials. Read the content and understand the code.
2. In the Arduino Software, open the DigitalReadSerial source code under “File”, “Examples”, “01.Basics”. Change the **“pushbutton pin to 13”** in the Arduino software and then upload the code.



fritzing

3. Open up the “Serial Monitor” under the Tools drop-down menu. It will take a few seconds to get started. Once it is running you will see the number “1” scrolling down the page. Unplug the wire that connects the 10k resistor to 5V. The number changes to “0” and continues scrolling down the page. The Arduino is reading a HIGH (5v) or LOW (0v) on Pin 13.
4. Now, open the AnalogReadSerial source code under “File”, “Examples”, “01.Basics” and then upload the code. Once again, open up the “Serial Monitor” under the Tools drop-down menu. It will take a few seconds to get started and then you should see a stream of numbers changing as you vary the potentiometer. The numbers will change from 0 to 1023 depending on what resistance the potentiometer is at. Also, depending on the way you connected the potentiometer, turning the potentiometer knob clockwise may increase or decrease the resistance. The potentiometer is setup as a voltage divider that will send an input voltage to the A0 analog input from 0v – 5v.

5. In the AnalogRead Serial code you uploaded to the right, you will notice the following line of code in the first line of the “LOOP”;

```
int sensorValue = analogRead(A0);
```

The analogRead(A0) statement reads the “state” of the A0 Pin that the potentiometer is connected to and returns the value to the “variable” called sensorValue. A “variable” is simply a way for the program to store the “state” of a pin (voltage between 0-5v that has been converted to a number between 0 and 1023) for use in the program. The “int” in front of the “variable” sensorValue stands for integer. This just means that the “variable” sensorValue can only store an integer between -32,768 and 32,767.

6. On the Arduino website, read up on “IF” statements. Under the “Learning” Tab, select “Reference”, then find “IF” on the left side of the page. Copy the example code to the right, have each LED come ON when the resistance is at 2k, 4k, 6k, 8k and 10k and then go OFF when the value is below each resistance.

7. Now, using what you know, change the existing code to the right and as you turn the potentiometer, have **ONLY** one LED come on and the others remain out.

Hint: use the following line of code. A double “&&” means AND. Therefore, if the value is greater than (>) 200 AND the value is also less than 400, turn the light on.

```
if (analogValue > 200 && analogValue < 400)
{
  digitalWrite(2, HIGH);
}
```

As a reference;

	== (equal to)
	!= (not equal to)
&& = Logical AND	< (less than)
= Logical OR	> (greater than)
!! = NOT	<= (less than or equal to)
	>= (greater than or equal to)

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1); // delay in between reads for stability
}
```

```
void setup() {
  // initialize the LED pin as an output:
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(2, OUTPUT);

  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() {
  // read the value of the potentiometer and store as a variable:
  int sensorValue = analogRead(A0);

  // set all the LEDs to be off
  digitalWrite(6, LOW);
  digitalWrite(5, LOW);
  digitalWrite(4, LOW);
  digitalWrite(3, LOW);
  digitalWrite(2, LOW);

  // if the sensorValue is above the value read, turn on the LED:
  if (sensorValue > 200)
  {
    digitalWrite(6, HIGH);
  }

  if (sensorValue > 400)
  {
    digitalWrite(5, HIGH);
  }

  if (sensorValue > 600)
  {
    digitalWrite(4, HIGH);
  }

  if (sensorValue > 800)
  {
    digitalWrite(3, HIGH);
  }

  if (sensorValue > 1000)
  {
    digitalWrite(2, HIGH);
  }
}
```

Show the teacher the working lab and code.

Marks will be taken off if the wires on the breadboard are not flat like shown above!!!!