* **Project 7 : The Working Of Potentiometer**
* ***What we do:***

A potentiometer is a simple knob that provides a variable resistance, which we can read into the Arduino board as an analog value. In this project, that value controls the rate at which an LED blinks.

* ***What we need:***

1. Breadboard
2. Arduino and USB Cable
3. Connecting Wires
4. One LED
5. Two 220ohm resistor
6. One Potentiometer

* ***The Code:***

// Project 7 - Demonstrating the working of potentiometer

int potPin = 0; // select the input pin for the potentiometer

int ledPin = 3; // select the pin for the LED

int val = 0; // variable to store the value coming from the sensor

void setup()

{

pinMode(ledPin, OUTPUT); // declare the ledPin as an OUTPUT

}

void loop()

{

val = analogRead(potPin); // read the value from the sensor

digitalWrite(ledPin, HIGH); // turn the ledPin on

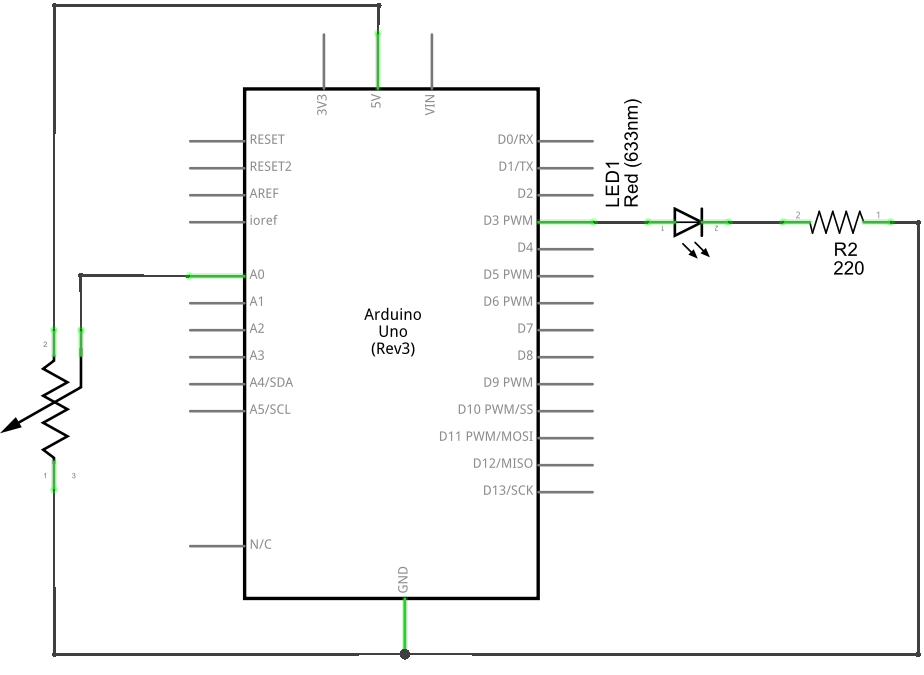
delay(val); // stop the program for some time

digitalWrite(ledPin, LOW); // turn the ledPin off

delay(val); // stop the program for some time

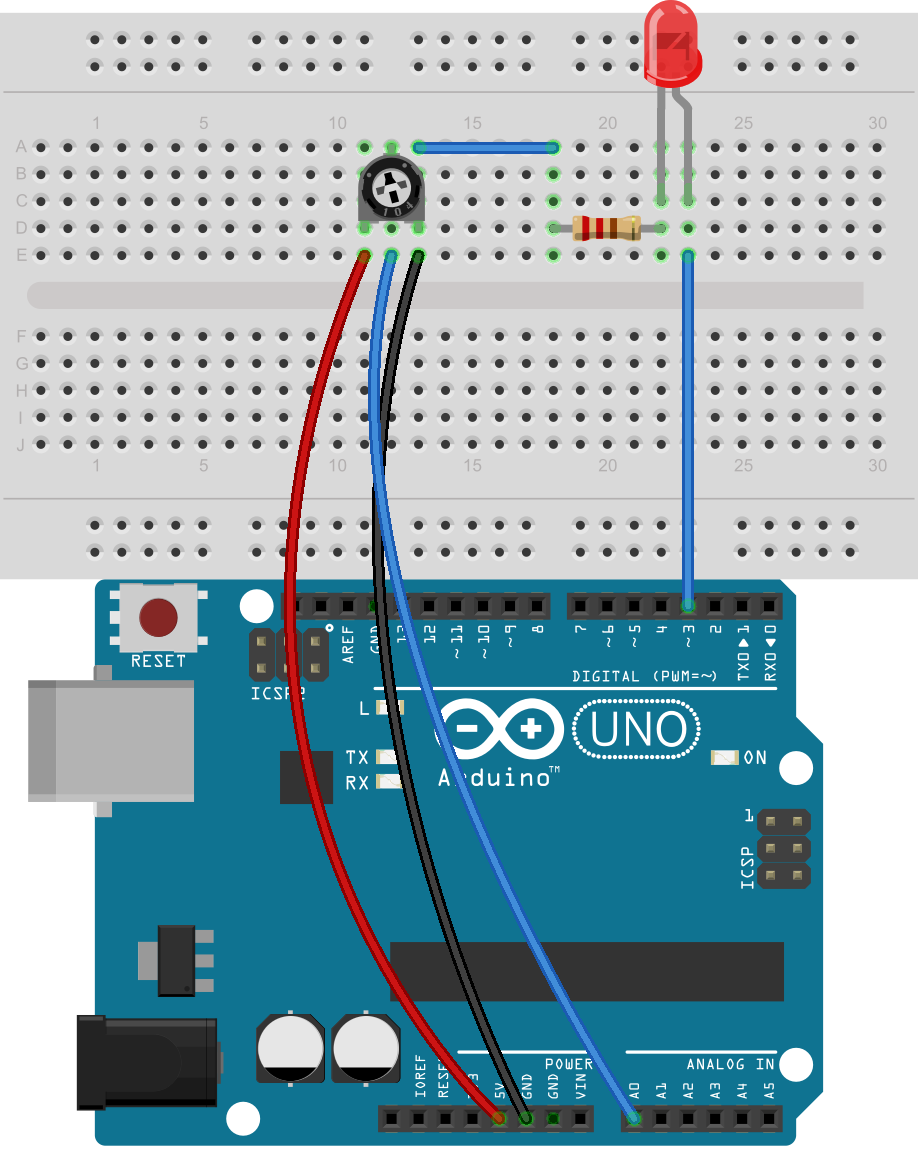
}

* ***The Schematic & Breadboard Circuit:***

******

We connect three wires to the Arduino board. The first goes to ground from one of the outer pins of the potentiometer. The second goes from 5 volts to the other outer pin of the potentiometer. The third goes from analog input 2 to the middle pin of the potentiometer.

By turning the shaft of the potentiometer, we change the amount of resistence on either side of the wiper which is connected to the center pin of the potentiometer. This changes the relative "closeness" of that pin to 5 volts and ground, giving us a different analog input. When the shaft is turned all the way in one direction, there are 0 volts going to the pin, and we read 0. When the shaft is turned all the way in the other direction, there are 5 volts going to the pin and we read 1023. In between, analogRead() returns a number between 0 and 1023 that is proportional to the amount of voltage being applied to the pin.

******

* ***Remember:*** The middle leg of the potentiometer will connect to analog pin.
* **Project 8 : The Working Of Photo-resistor**
* ***What we do:***

In this experiment we are going to use a element called photoresistor. This sensor allows us to interact with the external environment, through intensity of light. The photoresistor is based on light resistance, it will sense the light and will allow the microcontroller in this case Arduino to react and change the intensity of Led Diode.

The photoresistor creates a different resistance based on the intensity or the light. Changing the resistance through intensity changes the voltage too. The microntroller reads different values and will light up the Led with more or less intensity. A low resistance value will occur when the sensor is well lighted and a high resistance value will occur when it is in darkness.

* ***What we need:***

1. Breadboard
2. Arduino and USB Cable
3. Connecting Wires
4. One LED
5. Two 220ohm resistor
6. One Photo-resistor

* ***The Code:***

// Project 8 - Demonstrating the working of photo-resistor

int ledpin = 3; // Define led pin

int photoresistor = 0; // Variable to store data from analog pin for photoresistor

void setup()

{

pinMode(ledpin, OUTPUT); // output for the led pin

}

void loop()

{

photoresistor = (analogRead(0)/4); // Divides input 0-1023 to resemble 0-255

analogWrite(ledpin,photoresistor); // Getting the desired dimming effect

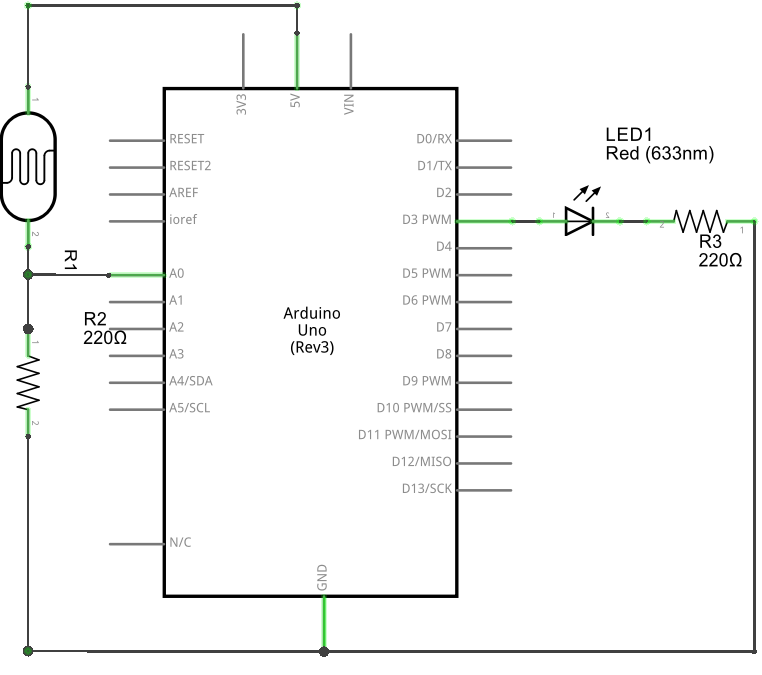
delay(20);

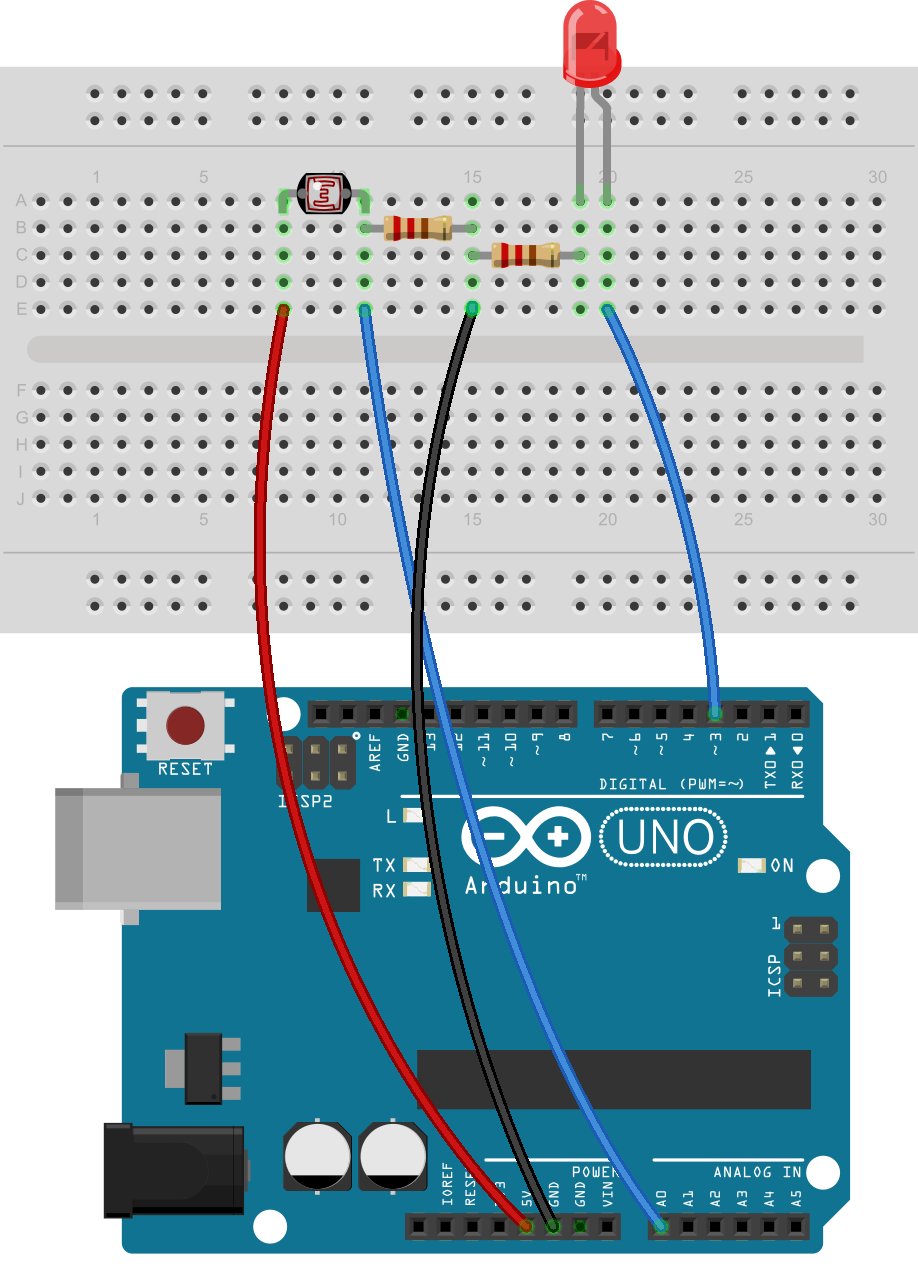
}

We declare the pins that we are going to use for the photoresistor and led diode. We then declare the function of the components that we have connected if is an input or an output . Line 8 is the starting of the cycle that is going to perform while Arduino is powered on. Line 10 is used for reading analog values from photoresistor and storing the values to a variable called “int readAnalogValue”.

Line 11 is a function that stores the value analog read divided by 4 to a variable called “int photoresistor”. Line 13 is the function that makes the Led lighting to change and line 14 is the delay that Arduino performs between different values.

* ***The Schematic & Breadboard Circuit:***

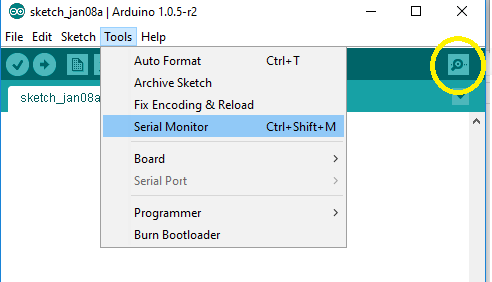




* ***Remember:*** Do not connect led or photo-resistor without the use of resistors as it can harm the components
* **Project 9 : Arduino Serial Monitor Display**
* ***What we do:***

So far, we have sent sketches to the Arduino and used the LEDs to show us output. Blinking LEDs make it easy to get feedback from the Arduino, but blinking lights can tell us only so much. In this project you’ll learn how to use the Arduino’s cable connection and the IDE’s Serial Monitor window to display data from the Arduino.

**The Serial Monitor** is a separate pop-up window that acts as a separate terminal that communicates by receiving and sending Serial Data. See the icon on the far right of the image below.

******

We will us the same schematic as the previous project

* ***What we need:***

1. Breadboard
2. Arduino and USB Cable
3. Connecting Wires
4. One LED
5. Two 220ohm resistor
6. One Photo-resistor

* ***The Code:***

// Project 9 – Arduino Serial Monitor Display

int ledpin = 3; // Define led pin

int photoresistor = 0; // Variable to store data from analog pin for photoresistor

void setup()

{

pinMode(ledpin, OUTPUT); // output for the led pin

Serial.begin (9600); // Serial communication begins

}

void loop()

{

Serial.println(analogRead(0); //Write the value of the photoresistor to the serial

// monitor

photoresistor = (analogRead(0)/4); // Divides input 0-1023 to resemble 0-255

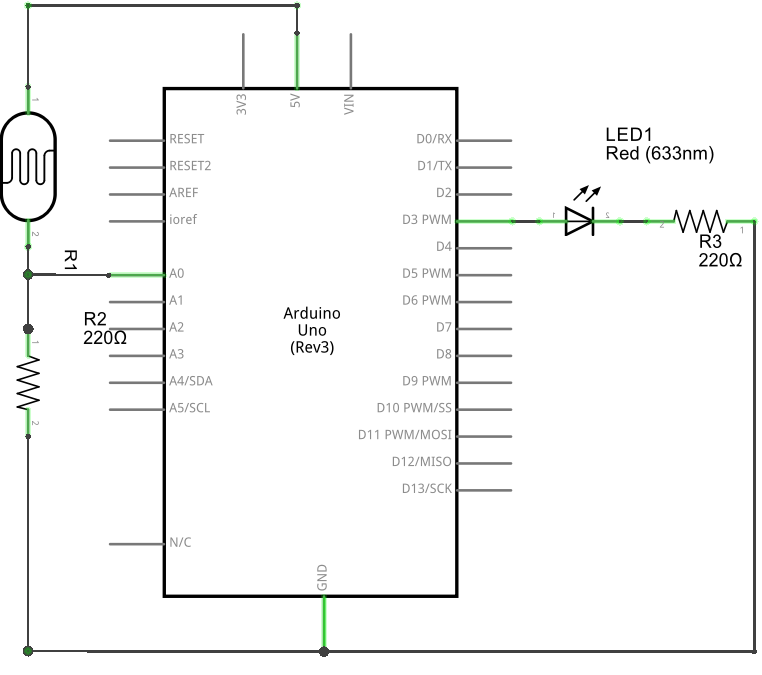
analogWrite(ledpin,photoresistor); // Getting the desired dimming effect

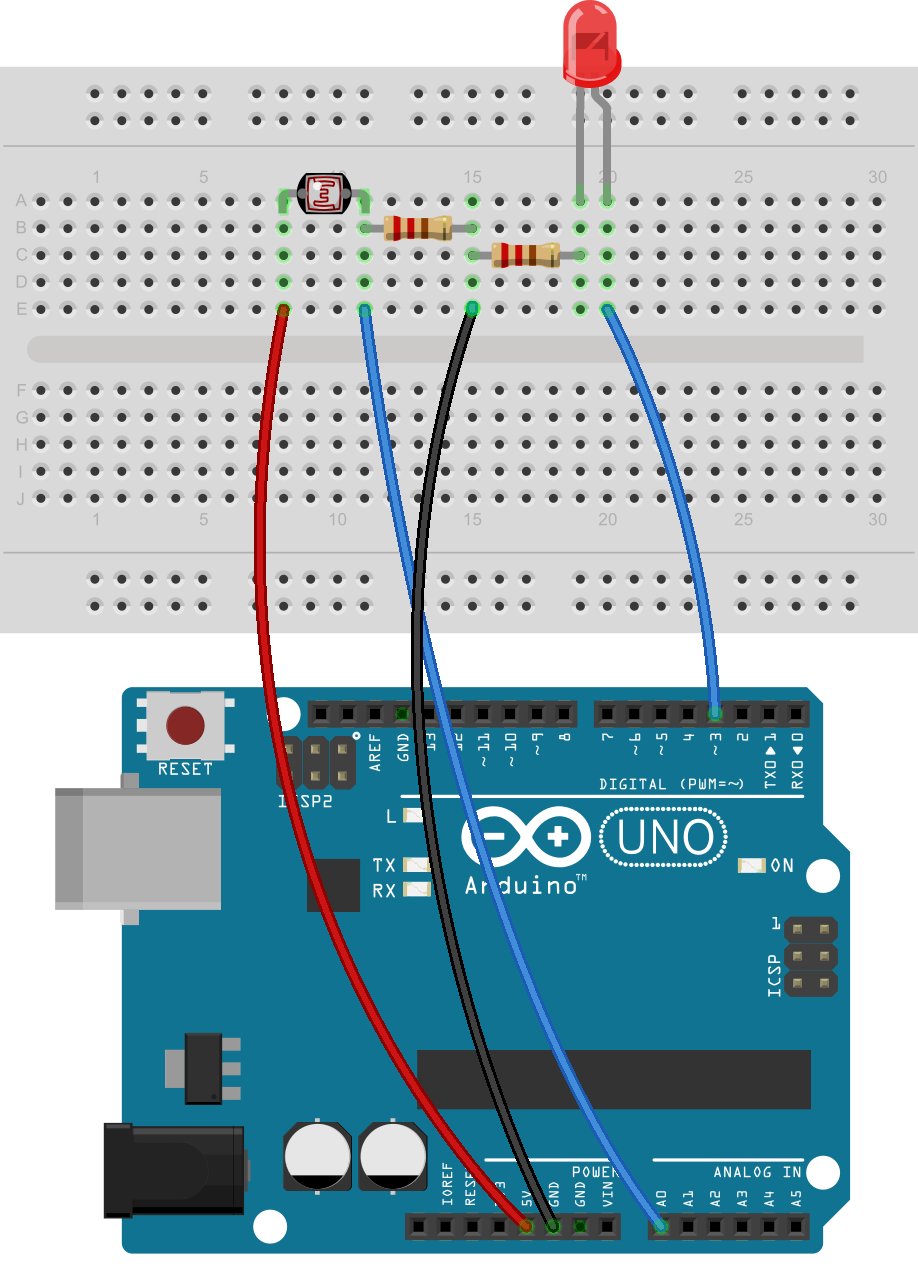
delay(20);

}

Once the code is uploaded, open the serial Monitor and you will be able to monitor the values from the photoresistor on the serial monitor.

* ***The Schematic & Breadboard Circuit:***





* ***Remember:*** Do not connect led or photo-resistor without the use of resistors as it can harm the components
* **Project 10 : Make Some Noise With Piezo Buzzer**
* ***What we do:***

A Piezo is nothing but an electronic device that can both be used to play tones and to detect tones. In our project we are plugging the Piezo on the pin number 9, that supports the functionality of writing a PWM signal to it, and not just a plain HIGH or LOW value. We are taking advantage of the processors capability to produde PWM signals in order to play music.

* ***What we need:***

1. Breadboard
2. Arduino and USB Cable
3. Connecting Wires
4. One Piezo
5. One 1kohm resistor

* ***The Code:***

// Project 10 – Make Some Noise With Piezo Buzzer

#define piezo\_pin 3 // pin 3 is capable of PWM output to drive tones

int del = 500;

void setup()

{

pinMode (piezo\_pin, OUTPUT);

}

void loop()

{

analogWrite (piezo\_pin, 128); // 50 percent duty cycle tone to the piezo

delay(del);

digitalWrite (piezo\_pin, LOW); // turn the piezo off

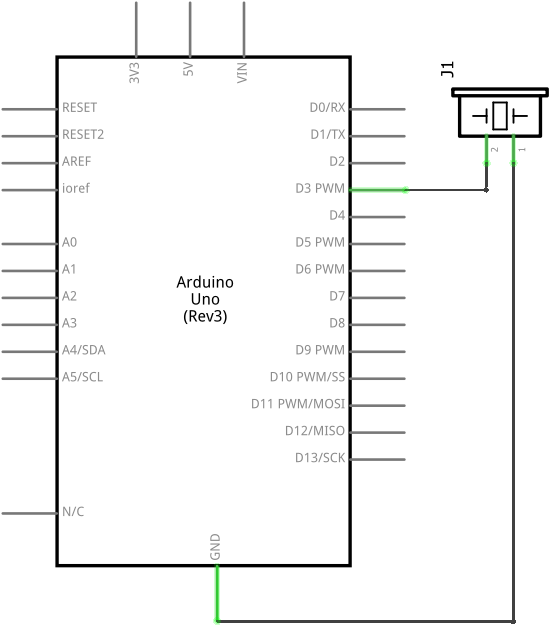
delay(del);

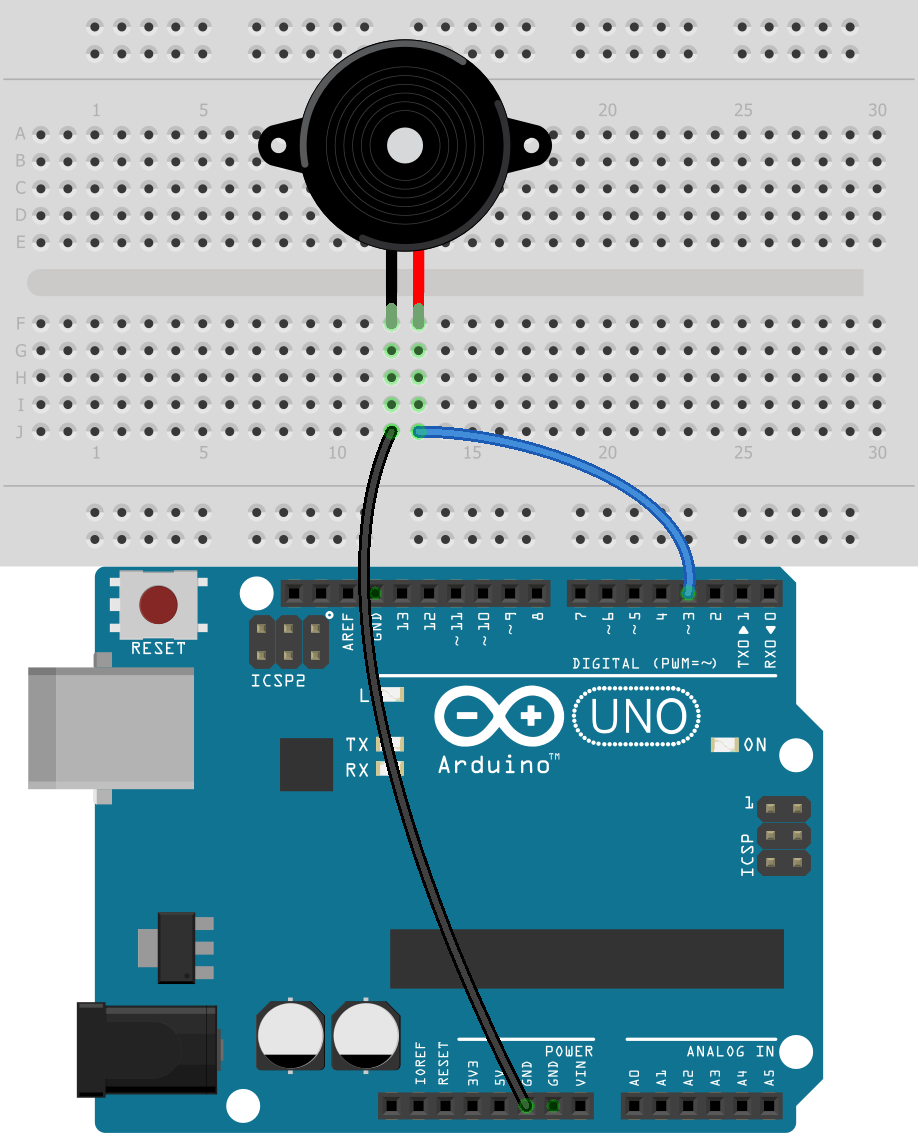
}

This sketch uses pulse-width modulation on digital pin three. If we change the duty cycle in the analogWrite() function (currently it’s 128, which is 50 percent on), then you can alter the volume of the buzzer.

The other thing to remember is that Piezos have polarity, commercial devices have one long and one short pin. We connect the short one to ground and the long one to the output.

* ***The Schematic & Breadboard Circuit:***

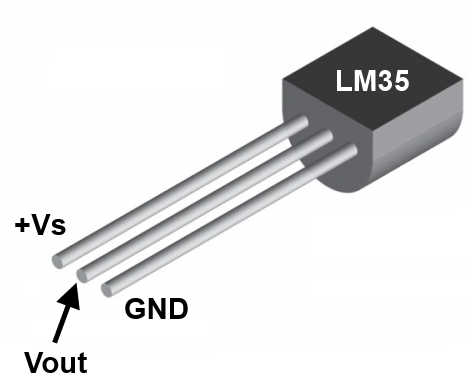
******

******

* ***Note:*** Resistor is not needed if you have 5V Piezo***.*** Also remember to connect the short pin to Ground.
* **Project 11 : Create A Quick Read Thermometer**
* ***What we do:***

In this project, we will make a simple temperature sensor using one LM35 Precision Temperature Sensor and Arduino. The circuit will send serial information about the temperature that you can see on your computer with the help of Serial monitor.

**LM35** is an analog, linear temperature sensor whose output voltage varies linearly with change in temperature. LM35 is three terminal linear temperature sensor from National semiconductors. The pin out of  LM35 is shown in the figure below.



* ***What we need:***

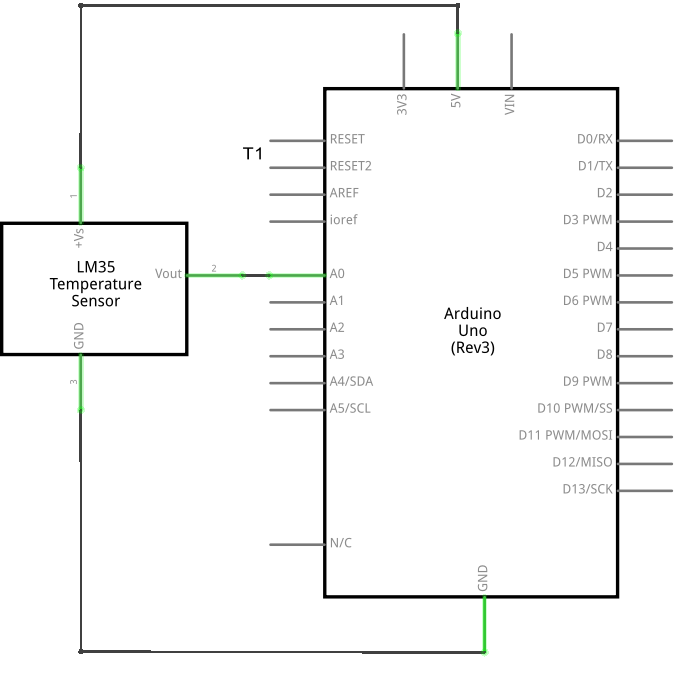
1. Breadboard
2. Arduino and USB Cable
3. Connecting Wires
4. LM35 Temperature Sensor

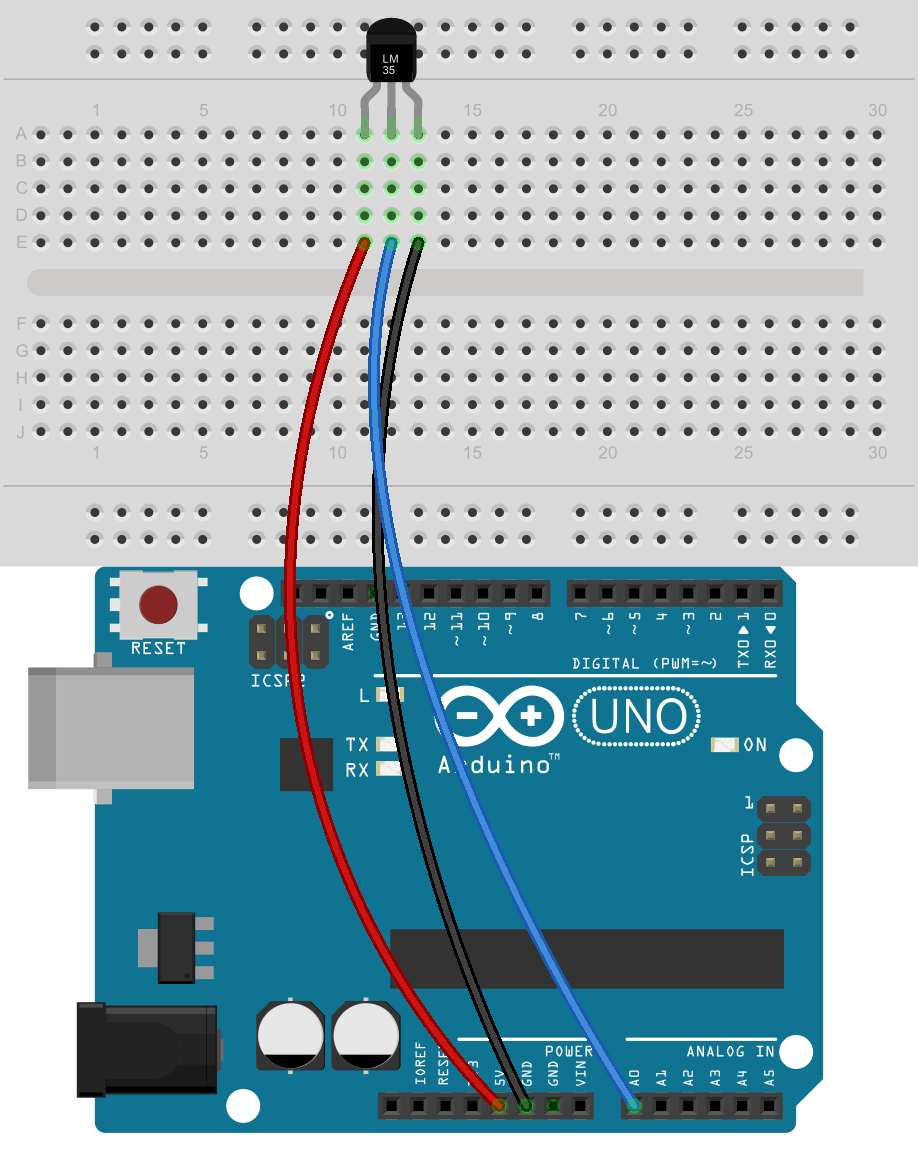
* ***The Code:***

// Project 11 – Create A Quick Read Thermometer

int val; // to store value from LM35  
int tempPin = 0; // analog pin to connect LM35  
  
void setup()  
{  
 Serial.begin(9600);   
}  
void loop()  
{  
 val = analogRead(tempPin);  
 float mv = ( val/1024.0)\*5000;   
 float cel = mv/10;  
  
 Serial.print("TEMPRATURE = ");  
 Serial.print(cel);   
 Serial.print("\*C");  
 Serial.println();  
 delay(1000);  
}

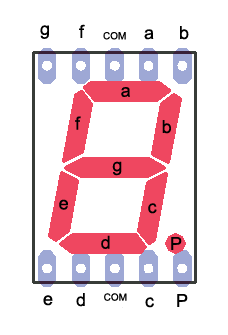
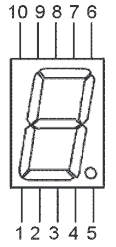
* ***The Schematic & Breadboard Circuit:***

******

******

* ***Note:*** Check twice the connection of LM53 before switching on the circuit or else the temperature sensor will burn up.
* **Project 12 : Hands On Seven Segment Display**
* ***What we do:***

LEDs are fun, but there are limits to the kinds of data that can be displayed with individual lights. In this section we’ll begin working with numeric digits in the form of seven-segment LED displays.The severn-segment display has seven LEDs arranged in the shape of number eight

* ***What we need:***

1. Breadboard
2. Arduino and USB Cable
3. Connecting Wires
4. 7 Segment display
5. 220 ohm resistor

* ***The Code:***

// Project 11 – Hands on Seven Segment Display

// we will display number ‘1’ and ‘2’ on the 7-Segment and will leave rest for you to

// explore

int pin\_1 = 6;

int pin\_2 = 5;

int pin\_4 = 4;

int pin\_6 = 9;

int pin\_7 = 8;

int pin\_9 = 2;

int pin\_10 = 3;

void setup()

{

pinMode (pin\_1, OUTPUT);

pinMode (pin\_2, OUTPUT);

pinMode (pin\_4, OUTPUT);

pinMode (pin\_6, OUTPUT);

pinMode (pin\_7, OUTPUT);

pinMode (pin\_9, OUTPUT);

pinMode (pin\_10, OUTPUT);

}

void loop()

{

// first we will display number ‘1’ :

pinMode (pin\_6, LOW);

pinMode (pin\_4, LOW);

delay(5000);

pinMode (pin\_6, HIGH);

pinMode (pin\_4, HIGH);

// now we will display number ‘2’ :

pinMode (pin\_7, HIGH);

pinMode (pin\_6, HIGH);

pinMode (pin\_10, HIGH);

pinMode (pin\_1, HIGH);

pinMode (pin\_2, HIGH);

delay(5000);

}

The above code will display number

Connect the pins described below:

1. Arduino Pin 2 to Pin 9

2. Arduino Pin 3 to Pin 10

3. Arduino Pin 4 to Pin 4.

4. Arduino Pin 5 to Pin 2

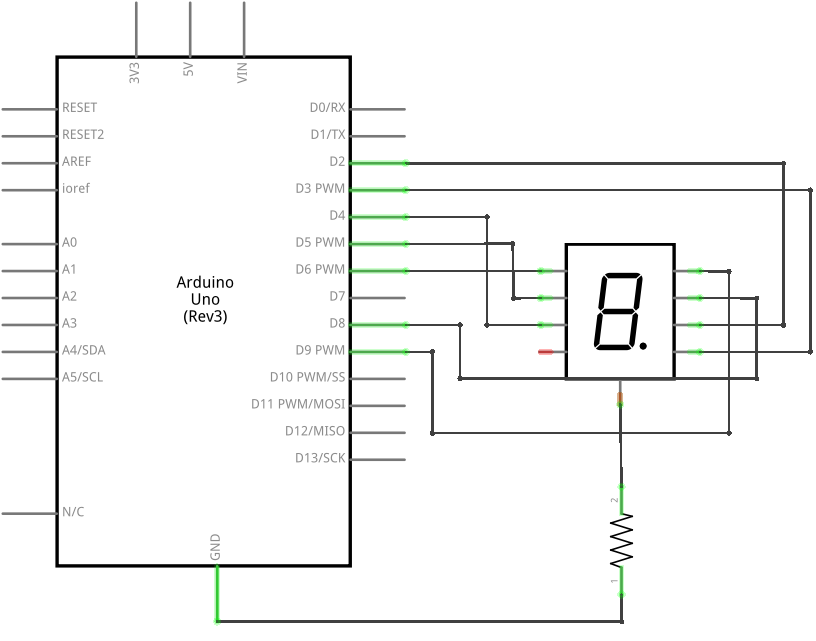
5. Arduino Pin 6 to Pin 1

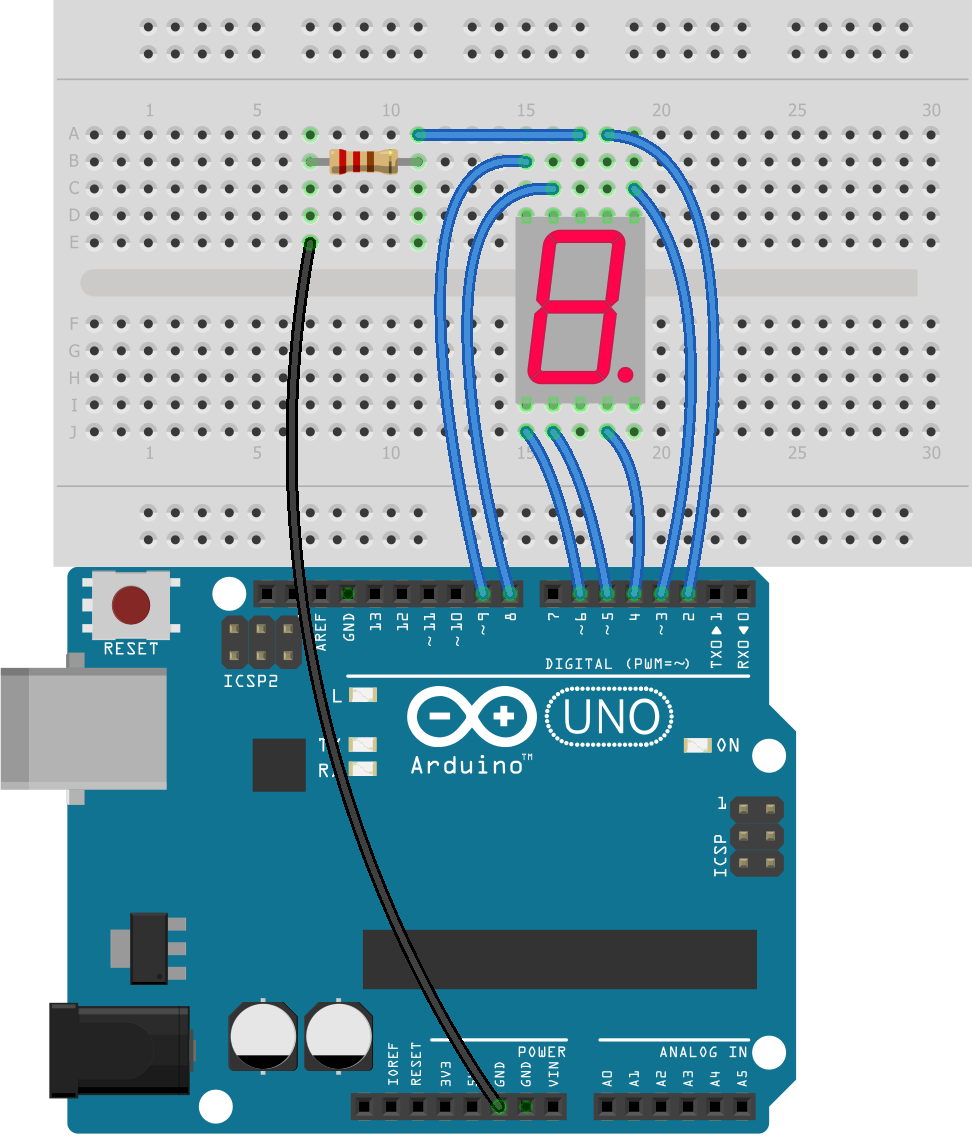
6. Arduino Pin 8 to Pin 7

7. Arduino Pin 9 to Pin 6

8. GND to Pin 3 or Pin 8 connected with 220 ohm resistor

* ***The Schematic & Breadboard Circuit:***

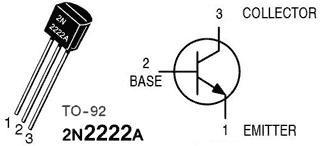
******

******

* ***Note:*** Do connect the resistor before switching on the circuit or the leds in the 7 Segment will break down
* **Project 13 : Working of DC Motor**
* ***What we do:***

In this lesson, you will learn how to control a small DC motor using an Arduino and a transistor.

We will use of the circuit and code, we make in project 6, the only difference is that we will use a DC Motor instead of led and will power it up using the 9V Battery



* ***What we need:***

1. Breadboard
2. Arduino and USB Cable
3. Connecting Wires
4. DC Motor
5. 220 ohm resistor
6. 1N4007 Diode
7. 22pf Capacitor
8. 2N2222A Transistor

* ***The Code:***

// Project 13: Working of DC Motor

#define base\_pin 8

void setup()

{

pinMode(base\_pin, OUTPUT); // output for the transistor base pin

}

void loop()

{

digitalWrite(base\_pin, HIGH); // provide current to base pin of transistor

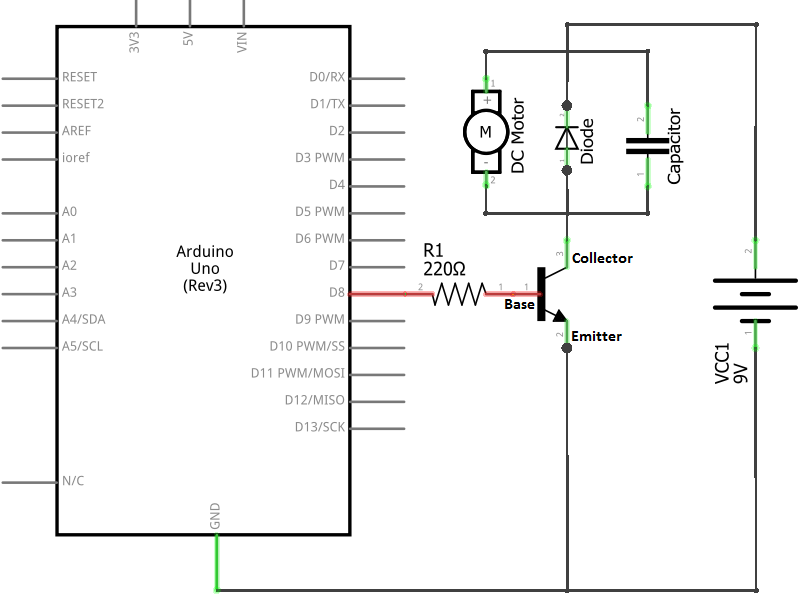
delay(5000); // wait for 5 seconds

digitalWrite(base\_pin, LOW); // stop the current flow to the transistor base pin

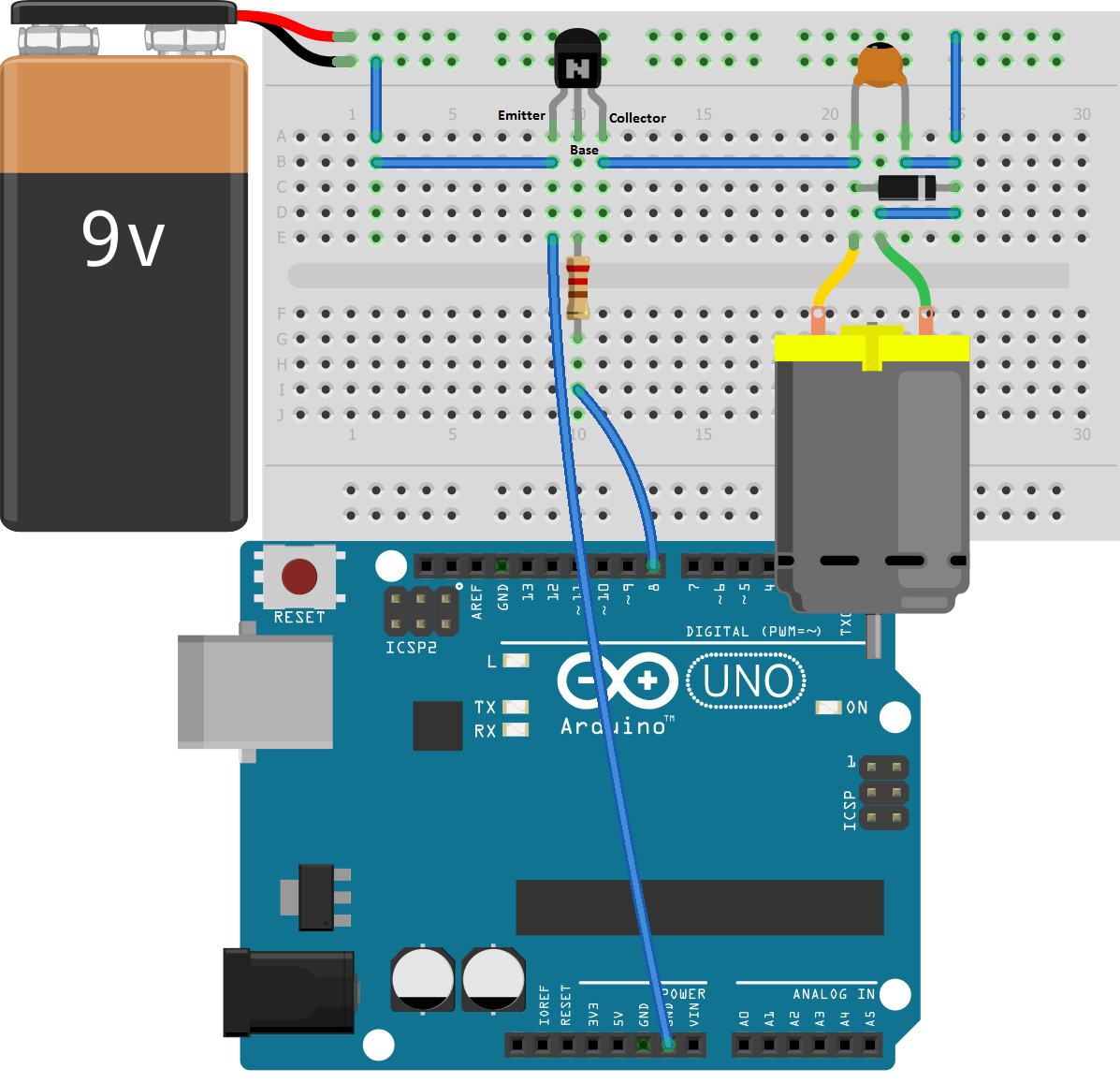
delay(5000); // wait for 5 seconds

}

* ***The Schematic & Breadboard Circuit:***

******

When the current is switched off from the motor, stray current exists for a brief amount of time inside the motor’s coil and has to go somewhere. The diode allows the stray current to loop around through the coil until it dissipates as a tiny amount of heat.

******

* ***Note :*** Please complete the project 6 before attempting this one
* ***Warning:*** Do not try to run the motor using the Arduino digital Pin or 5V as Arduino digital pin is not capable of providing the amount of current required by DC Motor.