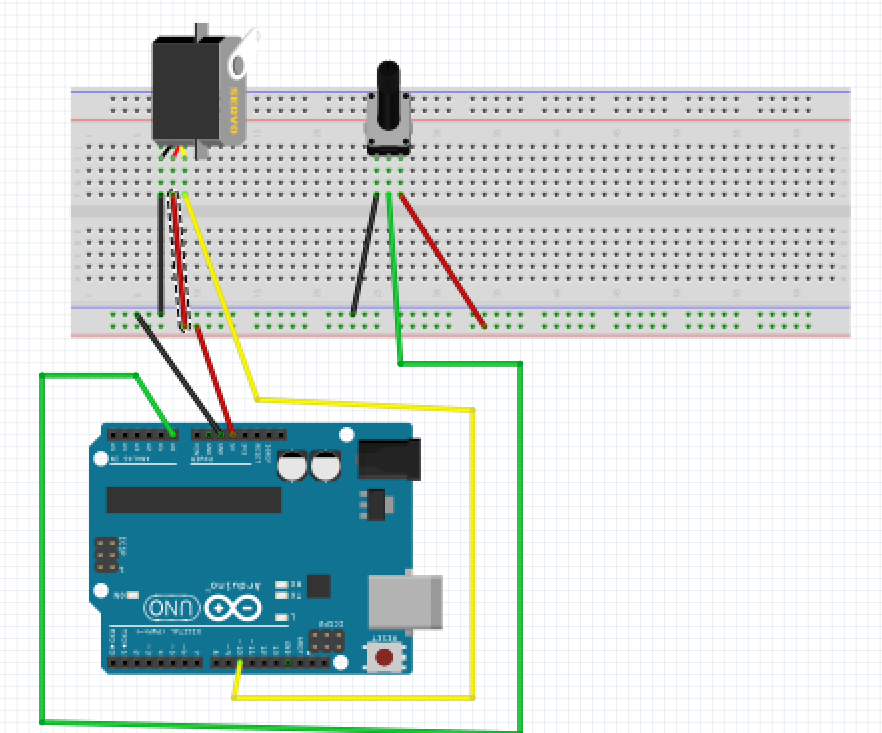
**ARDUINO LED PROJECT**

**Potentiometer to control servo position**



For this project we will need:

* Arduino board.
* Micro Servo
* potentiometer
* Breadboard.
* 3-pin connector.

Circuit Design:

1. First make sure that the Arduino is powered off (no USB cable plugged to power).
2. Check the servo motor, identify the three pins and respective colors.
3. Using a 3 pin connector,
4. Connect the yellow line to pin 10 of the Arduino. This will be our signal wire
5. Connect the red line to pin 5v of the Arduino.
6. Connect the black line to pin GND of the Arduino. This will be our signal wire.
7. Identify the 3 pins on the rotary potentiometer.
8. Plug the center pin of the rotary potentiometer to the breadboard. Take a green male jumper wire to pin A0 of the Arduino.
9. Plug the left pin of the rotary potentiometer to the breadboard. Take a black male jumper wire from the line connected to the left pin, to the ground common ground of the breadboard (marked by a blue (-) line
10. Plug the right pin of the rotary potentiometer to the breadboard. Take a red male jumper wire from the line connected from the line connected to the Right pin, to the 5V pin of the Arduino.
11. Plug the common ground of the breadboard to the GND pin of the arduino using a black male jumper wire.

|  |
| --- |
| #include <Servo.h>  Servo myServo;  int potentiometerPin = A0;  void setup() {  myServo.attach(10); }  /void loop(){  value=analogRead(potentiometerPin); //get value from Potentiometer  int value=map(value,0,1023,0,180); //map values from potentiometer range to servo range  myServo.write(value); // set servo pin to mapped value  delay(50);  } |

#include <Servo.h>

Servo myServo;

int servoPosition=0;

For the functionality of the server we need a library.

The library comes with inbuilt commands which control the functionality of the microservo. Instead of writing them from scratch we tell the program to use this library.

We then declare the name we want to use to refer to our servo

We then set the initial position of the servo.

int potentiometerPin = A0;

We declare the pin attached to the middle pin of the potentiometer

int value=analogRead(potentiometerPin);

This receives a value from the potentiometer. The value is in the range of 0-1023 since the position of the rotating dial can give 1023 levels of resistance.

int value=map(value,0,1023,0,180);

This converts the value of value from a range of 0-1023 to 0-180 .This range is used to give the position of the servo. Since the servo can rotate 180 degrees.

myServo.write(servoPosition);

This instructs the microservo to move to the expected position

voidloop(){}

After executing thevoid setup() function, we enter the void loop() and this function is executed continuously and repeatedly, until you Arduino is powered off.