Mapúa University

School of Electrical, Electronics and Computer Engineering

Real-Time Embedded System

COE187P/ E01

Potentiometer

Lab: #02

Submitted By:

Sardina, Kent Johnric M.

Submitted To:

Engr. Jocelyn Villaverde

I. Introduction

A Potentiometer is useful when you want to have a variable resistor which can be used in different applications like dimming an LED and also useful to use in Arduino to input a PWM signal then outputting the value by using map() function of the Arduino, also AnalogRead() is used since we want to read the value of the pin in PWM. The potentiometer has an advantage that it doesn't draw any current from the circuit, map() function was used in this experiment and its job is to map the raw value of the potentiometer and set it to your desired high and low values, constrain() function was also used and its job is to set minimum value and maximum value this function, constrain() function is useful to use after using the map() function since the map does not constrain the output.

II. Objectives

1. Familiarize the basic connection of Arduino
2. Familiarize on how to code the Arduino
3. Using AnalogRead() function from the Arduino library
4. Familiarize reading and writing analog signals
5. Use map(), constrain() function of the Arduino
6. Using Potentiometer in inputting PWM signals
7. Using LED to output PWM signals

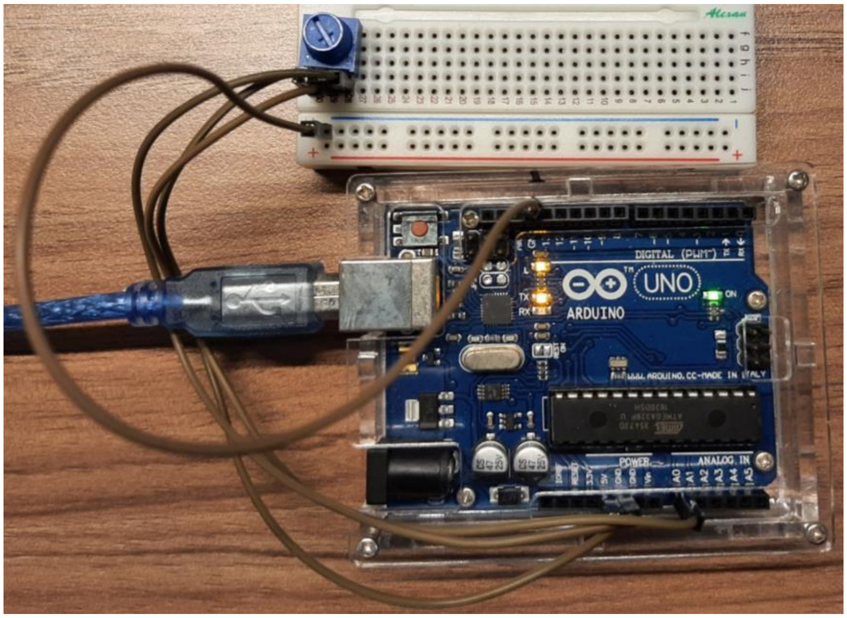
III. Materials and Components

- Arduino UNO
- Arduino UNO peripherals
- LED (Red, Green, Blue)
- 300-ohm resistor
- 10K Potentiometer
- Jumper Wires
- Breadboard
- Arduino IDE

IV. PROCEDURE

Part 01: Potentiometer

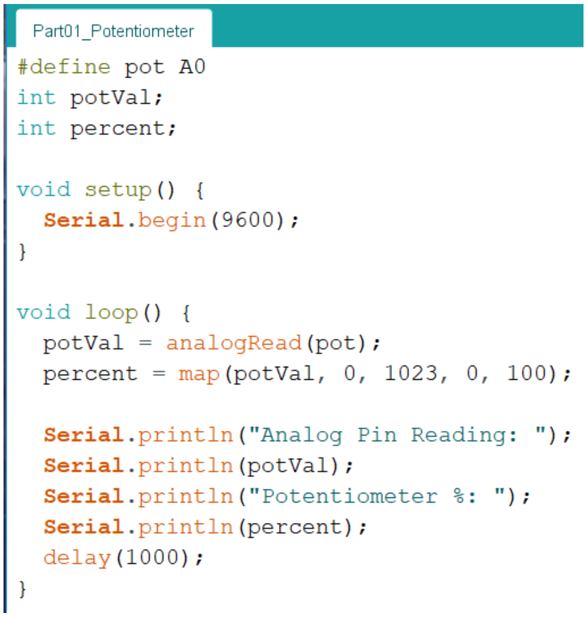
1. Follow the Connection of the Potentiometer to the Arduino



2. Left pin of the potentiometer to 5volts, Middle pin to A0 and Right pin to Ground

3. Open Arduino IDE

4. Enter the Code Below



```
#define pot A0
int potVal;
int percent;

void setup() {
  Serial.begin(9600);
}

void loop() {
  potVal = analogRead(pot);
  percent = map(potVal, 0, 1023, 0, 100);

  Serial.println("Analog Pin Reading: ");
  Serial.println(potVal);
  Serial.println("Potentiometer %: ");
  Serial.println(percent);
  delay(1000);
}
```

5. Select the correct COM port for the Arduino

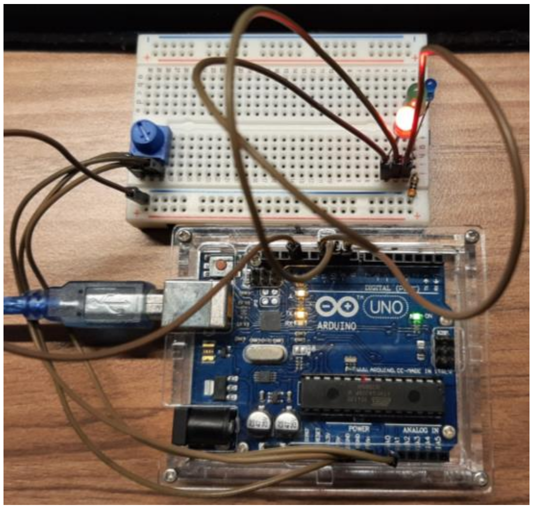
6. Verify, Save, and Upload the code to the Arduino

7. Open the Serial Monitor of the Arduino IDE to determine the percentage value of the potentiometer

8. Test and observe the results

Part 02: LED and Potentiometer

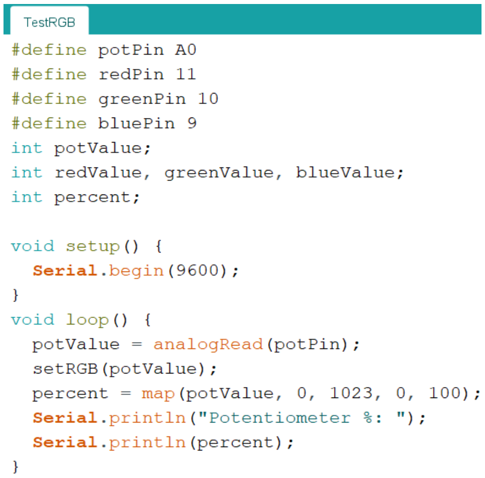
1. Follow the Connection of the LEDs to the Arduino (the connection for the potentiometer is the same as Part01).



2. Connect the Blue LED to pin 9, Green to pin 10, and Red to 11.

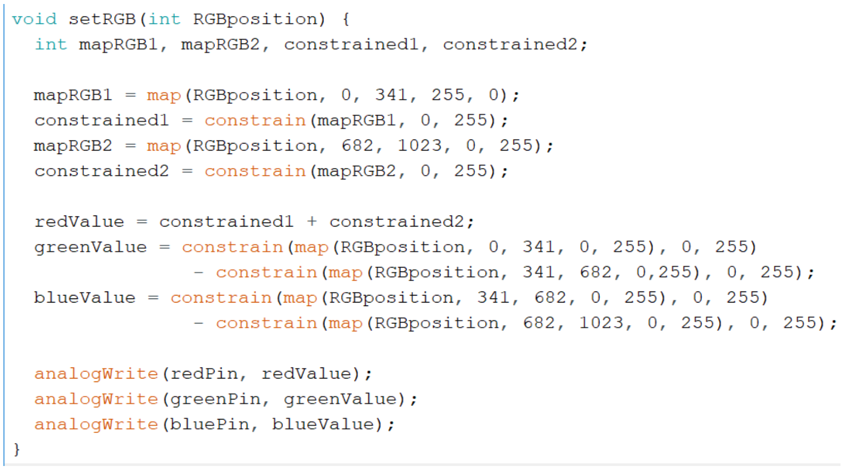
3. Open Arduino IDE

4. Enter the Code Below



```
#define potPin A0
#define redPin 11
#define greenPin 10
#define bluePin 9
int potValue;
int redValue, greenValue, blueValue;
int percent;

void setup() {
  Serial.begin(9600);
}
void loop() {
  potValue = analogRead(potPin);
  setRGB(potValue);
  percent = map(potValue, 0, 1023, 0, 100);
  Serial.println("Potentiometer %: ");
  Serial.println(percent);
}
```



```
void setRGB(int RGBposition) {
  int mapRGB1, mapRGB2, constrained1, constrained2;

  mapRGB1 = map(RGBposition, 0, 341, 255, 0);
  constrained1 = constrain(mapRGB1, 0, 255);
  mapRGB2 = map(RGBposition, 682, 1023, 0, 255);
  constrained2 = constrain(mapRGB2, 0, 255);

  redValue = constrained1 + constrained2;
  greenValue = constrain(map(RGBposition, 0, 341, 0, 255), 0, 255)
               - constrain(map(RGBposition, 341, 682, 0,255), 0, 255);
  blueValue = constrain(map(RGBposition, 341, 682, 0, 255), 0, 255)
               - constrain(map(RGBposition, 682, 1023, 0, 255), 0, 255);

  analogWrite(redPin, redValue);
  analogWrite(greenPin, greenValue);
  analogWrite(bluePin, blueValue);
}
```

5. Select the correct COM port for the Arduino

6. Verify, Save, and Upload the code to the Arduino

7. Open Serial Monitor to determine the percentage value of the potentiometer

8. Test and Observe the results

V. Results and Discussion

In this experiment by using map() function and using the 10 bit unsigned integer maximum value which is 0-1023 (used in potentiometer and other applications of AnalogRead() we could scale it down to 0-255 value because PWM has a maximum value of 255 only. In the first part of the experiment I used a 0-100 map() value because I want to output the percent value of the potentiometer, there are no problems in this part. The second part of the experiment is using an RGB LED but I don’t have an RGB so I used a red, green and blue LED to mimic the functions of the RGB LED; I had some errors when a some LED just turn off completely when using a map() function then I used a constrain() function but it still turns off immediately, I researched some sample codes on the internet and I used the formula for the RGB which is the redValue, greenValue, and the blueValue then the code worked with no problem.

VI. Conclusion

After the parts and activity, all the objectives in this experiment are achieved. By using the map() we can set the 10-bit unsigned integers being read by the analog pin to the potentiometer to a value or range that it is compatible with PWM since we are using RGB LED we need PWM to control the amount of voltage that will be outputted by the pin, and since the map() function does not constrain the values and I am having errors with the output I need to use the constrain() function and the RGB formula code I searched on the internet the led start behaving perfectly with no errors.