Mapúa University

School of Electrical, Electronics and Computer Engineering

Real-Time Embedded System

COE187P/ E01

Photosensor

Lab: #03

Submitted By:

Sardina, Kent Johnric M.

Submitted To:

Engr. Jocelyn Villaverde

I. Introduction

A photocell resistor is used in this experiment and it reads the amount of light incident on it, photocell works by decreasing and increasing the resistance value depending on the amount of light that it is receiving; when a large amount of light is given to the photocell its resistance value is decreasing while a small amount of light is given the resistance is increasing. A photocell is used mostly in automation like automatic lights to activate when it is dark. A photocell has only 2 connections these are VCC and GND, and the way we want to measure the value and resistance of the photocell we need to put a connection between the GND pin where the Arduino can read the resistance value of the photocell.

II. Objectives

1. Familiarize the basic connection and functions of Arduino
2. Familiarize on how to code the Arduino
3. Familiarize reading and writing analog and digital signals
4. Using AnalogRead() function to read the photocell
5. Using map() function to set the range of the values PWM
6. Use Photocell as the Photosensor
7. Use a PWM signal to control the LED
8. Increment counter using 7-segment display

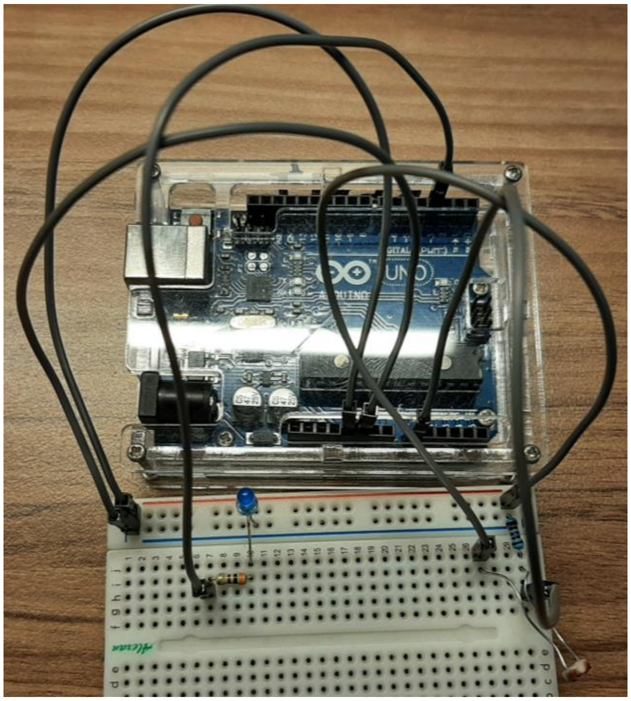
III. Materials and Components

- Arduino UNO and peripherals
- LED Blue
- 300-ohm resistor
- Jumper Wires
- Bread Board
- 7-segment display
- Arduino IDE

IV. PROCEDURE

Part 01: Photosensor and LED

1. Follow the Connection of the photocell and LED to the Arduino

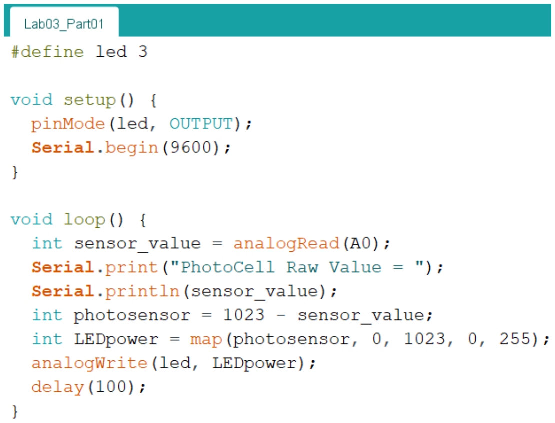


2. Connect the signal button(GND) of the photocell to pin A0

3. Connect the positive pin of the LED to pin 3

4. Open Arduino IDE

5. Enter the Code Below



```
#define led 3

void setup() {
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int sensor_value = analogRead(A0);
  Serial.print("PhotoCell Raw Value = ");
  Serial.println(sensor_value);
  int photosensor = 1023 - sensor_value;
  int LEDpower = map(photosensor, 0, 1023, 0, 255);
  analogWrite(led, LEDpower);
  delay(100);
}
```

6. Select the correct COM port for the Arduino

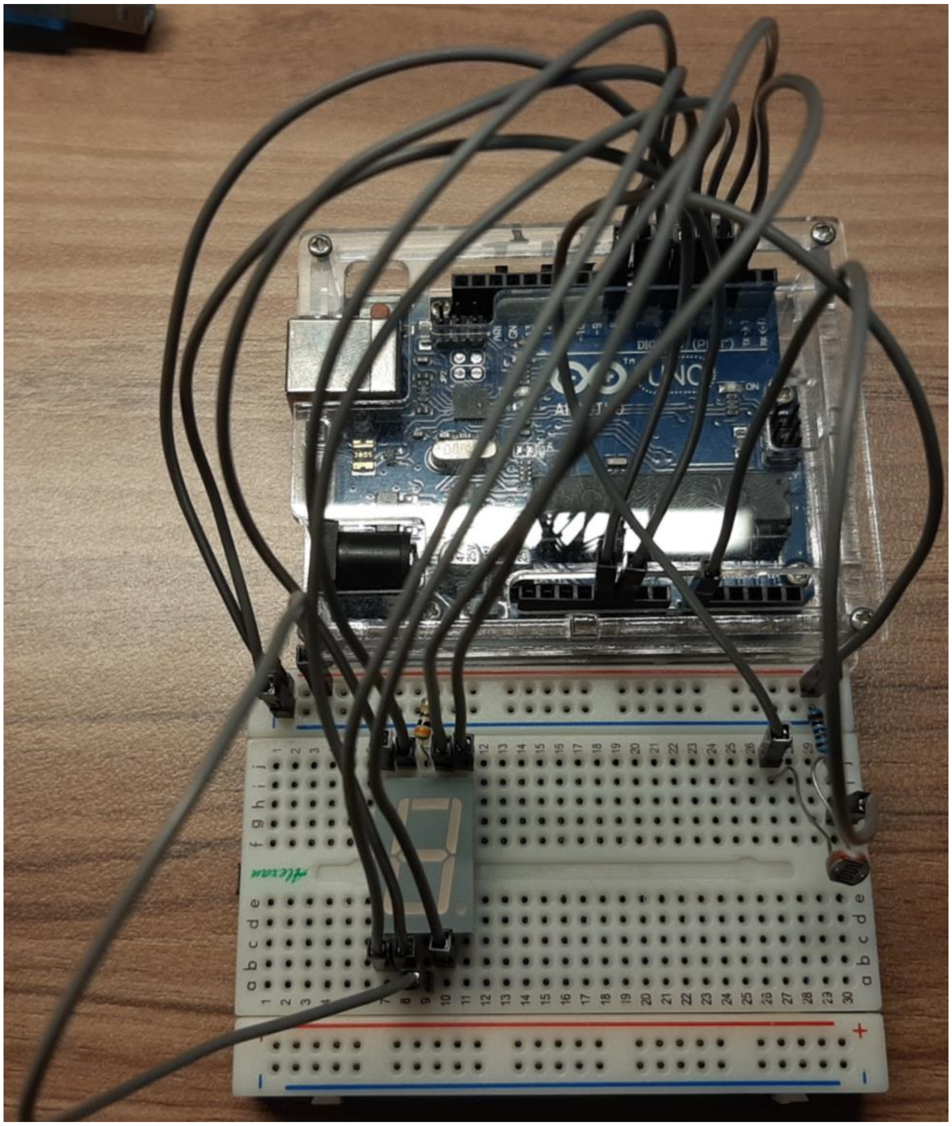
7. Verify, Save, and Upload the code to the Arduino

8. Open the Serial monitor of the Arduino IDE to determine the raw value of the Photocell

9. Test and observe the results

Part 02: 7-Segment Display and Photosensor

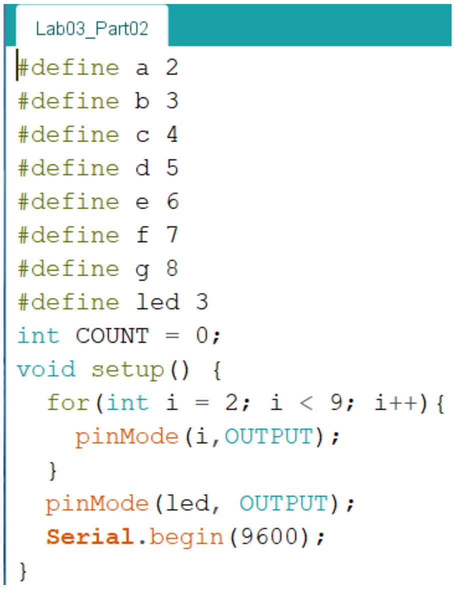
1. Follow the Connection of the 7-Segment display and the Photocell



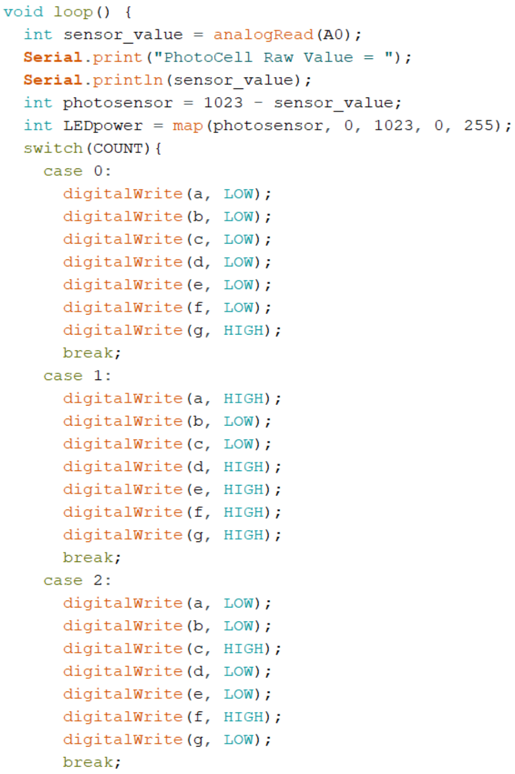
2. Connect the 7-Segment pin to pin3 to pin8 to the Arduino in order

3. Open Arduino IDE

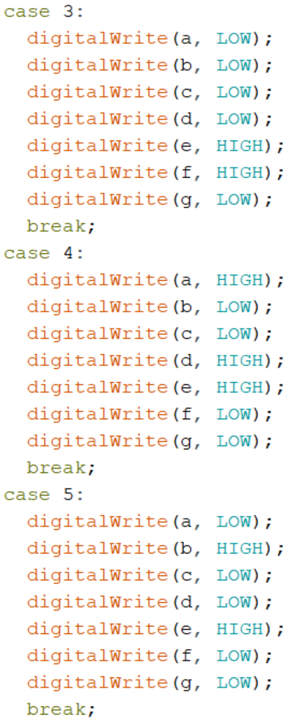
4. Enter the code below



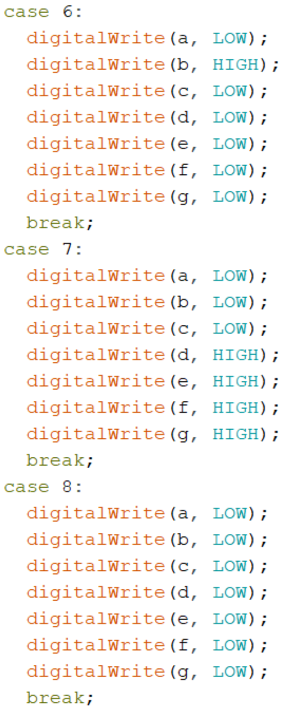
```
#define a 2
#define b 3
#define c 4
#define d 5
#define e 6
#define f 7
#define g 8
#define led 3
int COUNT = 0;
void setup() {
  for(int i = 2; i < 9; i++){
    pinMode(i,OUTPUT);
  }
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}
```



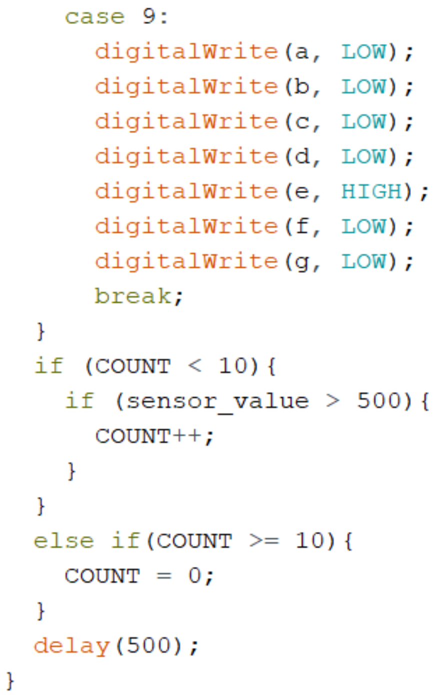
```
void loop() {
  int sensor_value = analogRead(A0);
  Serial.print("PhotoCell Raw Value = ");
  Serial.println(sensor_value);
  int photosensor = 1023 - sensor_value;
  int LEDpower = map(photosensor, 0, 1023, 0, 255);
  switch(COUNT){
    case 0:
      digitalWrite(a, LOW);
      digitalWrite(b, LOW);
      digitalWrite(c, LOW);
      digitalWrite(d, LOW);
      digitalWrite(e, LOW);
      digitalWrite(f, LOW);
      digitalWrite(g, HIGH);
      break;
    case 1:
      digitalWrite(a, HIGH);
      digitalWrite(b, LOW);
      digitalWrite(c, LOW);
      digitalWrite(d, HIGH);
      digitalWrite(e, HIGH);
      digitalWrite(f, HIGH);
      digitalWrite(g, HIGH);
      break;
    case 2:
      digitalWrite(a, LOW);
      digitalWrite(b, LOW);
      digitalWrite(c, HIGH);
      digitalWrite(d, LOW);
      digitalWrite(e, LOW);
      digitalWrite(f, HIGH);
      digitalWrite(g, LOW);
      break;
```



```
case 3:
  digitalWrite(a, LOW);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, LOW);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, LOW);
  break;
case 4:
  digitalWrite(a, HIGH);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
  break;
case 5:
  digitalWrite(a, LOW);
  digitalWrite(b, HIGH);
  digitalWrite(c, LOW);
  digitalWrite(d, LOW);
  digitalWrite(e, HIGH);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
  break;
```



```
case 6:
  digitalWrite(a, LOW);
  digitalWrite(b, HIGH);
  digitalWrite(c, LOW);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
  break;
case 7:
  digitalWrite(a, LOW);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
  break;
case 8:
  digitalWrite(a, LOW);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
  break;
```



```
    case 9:
      digitalWrite(a, LOW);
      digitalWrite(b, LOW);
      digitalWrite(c, LOW);
      digitalWrite(d, LOW);
      digitalWrite(e, HIGH);
      digitalWrite(f, LOW);
      digitalWrite(g, LOW);
      break;
  }
  if (COUNT < 10){
    if (sensor_value > 500){
      COUNT++;
    }
  }
  else if(COUNT >= 10){
    COUNT = 0;
  }
  delay(500);
}
```

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 raw value of the Photocell

8. Test and Observe the results of the Arduino

V. Results and Discussion

In this experiment, by using map() function I could set a PWM range 0-255 to control the brightness of the LED. In the first part of the experiment I used an LED and Photocell, in the code I used the serial monitor to output the raw value of the photocell and since we need a system that when photocell receives light will make the LED dim therefore I subtract the sensor value to 1023 and uses the map() function to set range 0-255, and lastly used a AnalogWrite() function to output a PWM to the LED. In the second part of the experiment, I used a 7-segment display and photocell, when the photocell receives light the 7-segment display will increment; the photocell has the same connection with the first part and I wrote a loop code that will increment the value and using an if-else statement with a photocell value argument; when the raw value of the photocell gets to 500 above the number will increment.

VI. Conclusion

After the parts and activity, all the objectives in this experiment are achieved. By using several functions of the Arduino, I was able to create my code for this experiment. If you want to use the value of a specific sensor and you want to the value to output to a PWM controlled part you first need to convert that 10-bit unsigned value to 0-255 PWM range by using the map() function of the Arduino for it to work.