**LEC 291 Section 20C**

**LAB #: 1**

**LAB SECTION: 20D**

**TEAM #: D\_5C**

|  |  |  |
| --- | --- | --- |
| ***Student name*** | ***Student number*** | ***Contribution percentage*** |
| **LINDA MUNISI** | **22535158** | **33.3%** |
| **SIRINE TRIGUI** | **46443157** | **33.3%** |
| **YIXIN ZHAO** | **49427140** | **33.3%** |

*Contribution summary*:

We worked together during the lab session. Linda fixed the coding of 2d. She implemented the printing correctly for the LEDs. Yixin did the Fritzing diagram. We all contributed to the report. Linda and Sirine wrote most of the task 2 and Appendix III while Yixin also contributed to other parts of the report.

**A. Introduction and motivations**

The lab described in this report mainly focused on getting familiar with Arduino UNO board, coding in Arduino IDE and using Fritzing, which are very useful tools and basic knowledge needed in order to proceed other labs later in the course ELEC 291.

In the first part of the lab, we practised coding with Arduino IDE. Later in the second part, we applied the basic knowledge learned from part 1 and used basic Arduino UNO digital and analog IO interfacing with other electronic components like resistors, LEDs, photocell. We also used Fritzing to produce a breadboard schematic diagram of our complete circuits from both part of the lab. It is included in the Appendix I.

**B. Lab Description**

Task1:

We went through the example provided and had basic ideas on how Arduino works and how to make it blink every 2 seconds. Did not need to connect any LEDs since we were using the built-in LED. Through coding, we understood the configuration of pinMode and digitalWrite function, and successfully set the built-in LED’s status to high and low, resulted in the required blinking. Tested the circuit by observation.

Task2:

a) Digital Output Using LEDs: We used Arduino board and RGB LED. By programming through Arduino IDE, we made the different color LEDs lighted up in designed order. The order was controlled by our program. We had a deeper understanding of LOOP function. One problem we met was to tell which pin of the LED’s is for common use because there was a typo in the datasheet. We did not connect the resistors to the Ground in the beginning so that our circuit did not work. A TA helped us find this mistake and the other parts of our circuit worked perfectly. We tested the result by comparing the code and the actual lighting up order.

b) Digital Input Using Switches: In this part, we added A DIP switch including 8 independent toggle switches (Alco ADE08) but we only used 2 toggle switches. We learned how to identify a resistor value as well. We selected a sequence of colors to be displayed whenever the appropriate switches states are set. For example, if we have switch 2 ON and switch 3 OFF then the expected output (sequence displayed) is Red Green Blue. We learned that the pins are with high impedance. For that reason, without connecting the switches to the pins through resistors, we were getting an inappropriate output. Tested the result by observation.

c) Analog Input Using Photocell and PWM: For the hardware part, we added a single-color LED connected to the Arduino through a Pulse Width Modulation (PWM) pin and a photocell which was our analog input (had to be connected to one of the analog Arduino pins). The reason behind using such analog input is to sense the level of the ambient light and then decide on how dim the LED must be. The new used function was analogWrite. Using the photocell allowed us to experiment the fading example that demonstrates the use of analog output (PWM). We encountered some difficulties in making it gradually light up because we did not understand the instruction. It turned out that the LED should be off when the light level is above the threshold we set instead of fading out gradually.

d) Using Serial Monitor: We wrote code that prints the output of the LEDs and the switches in the serial monitor. The function Serial.println( ) was used to print the values. For the switches, the value on the serial monitor is ON if the switch is high and vice versa. For the photocell, the value on the serial monitor changes depending on the light level on it. The monitor prints the value light controlled LED depending on its value. We finally found that we need “/r” to print all the results in one line. Lots of adjustment were made to our code.

- Our design procedures were mostly based on the circuits that have been shown on the Arduino files given to us. Alternatively, instead of connecting an LED in the first part, we used a built-in LED which came in the Arduino board.

- The testing procedure for the first part entailed checking whether the LED blinks every after two seconds. The results were positive with the LED being ON for two seconds and the OFF for the next two seconds and the series continues. For the second part, in part a) we tested whether the LED lights every 200ms and the results were positive and reacted as indicated in the code below. For part b) we tested whether the pattern of the lights followed the combination of the switches. The results were positive and followed the chosen pattern indicated in the code below. For part c) we tested whether the photocell responded by dimming after a light with higher threshold is shown on it. We also tested if the LED brightens up when a light of lower threshold is shone on it. This responded positively and followed the code indicated below.

- Problems encountered included the breadboard being too small and therefore the circuit connections becoming too complex to fit on a small breadboard. Also, another problem was making a fritzing copy of the circuit after it is all done and complex wasn’t too easy. We learnt that the best practice would be fritzing as we connect the circuit instead of leaving it till the end. We also learnt that we could use a bigger breadboard than the one we used.

**C. Conclusions**

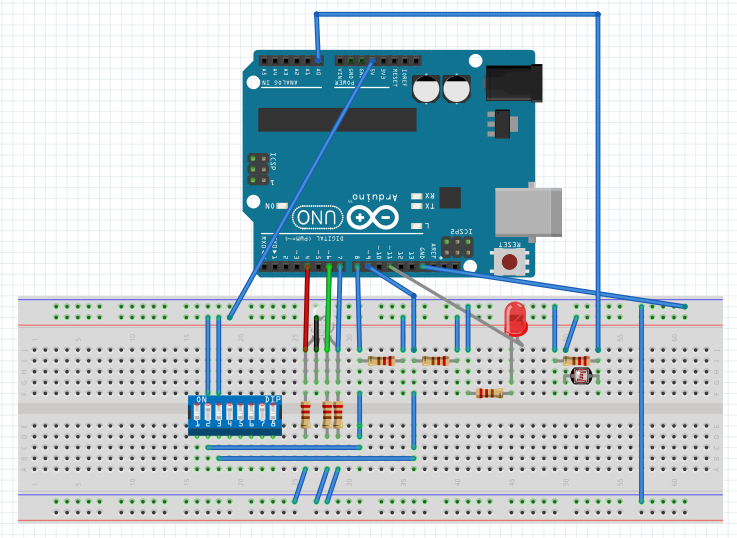
- In conclusion, this lab taught us the basics of Arduino and familiarized us with the environments and the functions that can be used in Arduino IDE. It was also a good foundation to refresh our C programming skills.

**D. References**

- Some references that we consulted during our experiments apart from the links provided in the handouts were, [these](https://www.google.ca/search?q=led+and+photocell+arduino&source=lnms&tbm=isch&sa=X&ved=0ahUKEwino8T6lqXKAhWCkh4KHZvGC3cQ_AUIBygB&biw=1188&bih=664#imgrc=Ddv3xDVKf_uxrM%3A)

**APPENDIX**

**Appendix I**



Fritzing Breadboard Diagram

**Appendix II**

**1.)**

void setup() {

// put your setup code here, to run once:

pinMode(13, OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)

delay(2000); // wait for a second

digitalWrite(13, LOW); // turn the LED off by making the voltage LOW

delay(2000); // wait for a second

}

**2a.)**

void setup() {

// put your setup code here, to run once:

pinMode(4, OUTPUT);

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(4, HIGH); // turn the BLUE LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(4, LOW); // turn the BLUE LED off by making the voltage LOW

delay(1000); // wait for a second

digitalWrite(6, HIGH); // turn the RED LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(6, LOW); // turn the RED LED off by making the voltage LOW

delay(1000); // wait for a second

digitalWrite(7, HIGH); // turn the GREEN LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(7, LOW); // turn the GREEN LED off by making the voltage LOW

delay(1000); // wait for a second

}

**2b.)**

//initialize the switches to the pin values

const int switch2 = 8;

const int switch3 = 9;

// variables will change:

int switchState2 = 0;

int switchState3 = 0;

void setup() {

// put your setup code here, to run once:

pinMode(switch2, INPUT); // FOR SWITCH 2

pinMode(switch3, INPUT); // FOR SWITCH 3

pinMode(4, OUTPUT); // FOR THE BLUE PIN OF LED

pinMode(6, OUTPUT); // FOR RED PIN OF LED

pinMode(7, OUTPUT); // FOR GREEN PIN OF LED

}

void loop() {

switchState2 = digitalRead(switch2);

switchState3 = digitalRead(switch3);

delay(200); //a delay of 200ms for each color change in the LED

//OFF

if (switchState2 == LOW && switchState3 == LOW)

{

digitalWrite(4, LOW);

digitalWrite(6, LOW);

digitalWrite(7, LOW);

}

delay(200);

//BRG

if (switchState2 == HIGH && switchState3 == HIGH) {

digitalWrite(4, HIGH); // turn the BLUE LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(4, LOW); // turn the BLUE LED off by making the voltage LOW

delay(1000); // wait for a second

digitalWrite(6, HIGH); // turn the RED LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(6, LOW); // turn the RED LED off by making the voltage LOW

delay(1000); // wait for a second

digitalWrite(7, HIGH); // turn the GREEN LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(7, LOW); // turn the GREEN LED off by making the voltage LOW

delay(1000); // wait for a second

}

delay(200);

//GBR

if (switchState2 == LOW && switchState3 == HIGH) {

digitalWrite(7, HIGH); // turn the GREEN LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(7, LOW); // turn the GREENED off by making the voltage LOW

delay(1000); // wait for a second

digitalWrite(4, HIGH); // turn the BLUE LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(4, LOW); // turn the BLUE LED off by making the voltage LOW

delay(1000); // wait for a second

digitalWrite(6, HIGH); // turn the RED LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(6, LOW); // turn the RED LED off by making the voltage LOW

delay(1000); // wait for a second

Serial.println(BRG);

}

delay(200);

//RGB

if (switchState2 == HIGH && switchState3 == LOW)

{

digitalWrite(6, HIGH); // turn the RED LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(6, LOW); // turn the RED LED off by making the voltage LOW

delay(1000); // wait for a second

digitalWrite(7, HIGH); // turn the GREEN LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(7, LOW); // turn the GREEN LED off by making the voltage LOW

delay(1000); // wait for a second

digitalWrite(4, HIGH); // turn the BLUE LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(4, LOW); // turn the BLUE LED off by making the voltage LOW

delay(1000); // wait for a second

}

delay(200);

}

**2c.)**

int ledPin = 11; // LED connected to digital pin 11

int analogPin = A0;

void setup() {

pinMode(ledPin, OUTPUT); // nothing happens in setup

pinMode(analogPin, INPUT);

}

void loop() {

int val = analogRead(analogPin); // read the input pin

analogWrite(ledPin, val / 4);

if (val <= 900)

{

// fade in from min to max in increments of 60 points:

for (int fadeValue = 0 ; fadeValue <= 1000; fadeValue += 60) {

// sets the value (range from 0 to 255):

analogWrite(ledPin, fadeValue);

// wait for 2 milliseconds to see the dimming effect

delay(200);

}

}

if (val > 900) {

analogWrite(ledPin, 0);

// wait for 2 milliseconds to see the dimming effect

delay(200);

}

}

**2d.)**

const int switch2 = 8; //switch2 connected to digital pin 8

const int switch3 = 9; //switch3 connected to digital pin 9

int ledPin = 11; // LED connected to digital pin 11

int analogPin = A0; //analogPin connected to the pin A0

//initialize the variables to 0

int switchState2 = 0;

int switchState3 = 0;

int LIGHTControlledLED =0;

int LEDsB = 0;

int LEDsR = 0;

int LEDsG = 0;

int sensorValue = 0;

int RGB =0;

int BRG =0;

int GBR =0;

//the code that will run once

void setup() {

Serial.begin(9600);

pinMode(switch2, INPUT); // FOR SWITCH 2

pinMode(switch3, INPUT); // FOR SWITCH 3

pinMode(4, OUTPUT); //FOR LEDsB

pinMode(6, OUTPUT); // FOR LEDsR

pinMode(7, OUTPUT); // FOR LEDsG

pinMode(ledPin, OUTPUT); //FOR LEDPIN

pinMode(analogPin, INPUT); //FOR ANALOGPIN

}

void loop() {

//READING THE STATES OF THE SWITCHES AND LEDs

switchState2 = digitalRead(switch2);

switchState3 = digitalRead(switch3);

LIGHTControlledLED = digitalRead(ledPin);

LEDsB = digitalRead(4);

LEDsR = digitalRead(6);

LEDsG = digitalRead(7);

sensorValue = digitalRead(analogPin);

//printing the output of switches on the serial monitor

if (switchState2 == LOW && switchState3 == HIGH) {

digitalWrite(7, HIGH); // turn the RED LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(7, LOW); // turn the RED LED off by making the voltage LOW

delay(1000); // wait for a second

digitalWrite(4, HIGH); // turn the GREEN LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(4, LOW); // turn the GREEN LED off by making the voltage LOW

delay(1000); // wait for a second

digitalWrite(6, HIGH); // turn the BLUE LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(6, LOW); // turn the BLUE LED off by making the voltage LOW

delay(1000); // wait for a second

Serial.print("\rSWITCH:OFF ON" );

Serial.print(" ");

}

//IF STATEMENTS TO PRINT THE OUTPUTS ON SERIAL MONITOR

if (switchState2 == 0 && switchState3 == 1)

Serial.print("\rSWITCH:OFF ON" );

Serial.print(" ");

if (switchState2 == 1 && switchState3 == 0)

Serial.print("\rSWITCH:ON OFF" );

Serial.print(" ");

if (switchState2 == 1 && switchState3 == 1)

Serial.print("\rSWITCH:ON ON" );

Serial.print(" ");

if (switchState2 == 0 && switchState3 == 0)

Serial.print("\rSWITCH:OFF OFF" );

Serial.print(" ");

if (switchState2 == 0 || switchState3 == 0)

Serial.print("LEDs:0FF" );

Serial.print(" ");

if (switchState2 == 1 || switchState3 == 0)

Serial.print("0N");

Serial.print(" ");

if (switchState2 == 1 || switchState3 == 1)

Serial.print("0N");

Serial.print(" ");

if (switchState2 == 0 || switchState3 == 1)

Serial.print("0N");

Serial.print(" ");

int sensorValue = analogRead(A0);

// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):

float voltage = sensorValue \* (5.0 / 1023.0);

// print out the value you read:

Serial.print(" ");

Serial.print("Photocell value: ");

Serial.print(voltage);

Serial.print("mV");

Serial.print(" ");

int val = analogRead(analogPin); // read the input pin

analogWrite(ledPin, val / 4);

Serial.println("Light-controlled LED: OFF");

int val = analogRead(analogPin); // read the input pin

analogWrite(ledPin, val / 4);

if (val <= 900)

{

// fade in from min to max in increments of 60 points:

for (int fadeValue = 0 ; fadeValue <= 1000; fadeValue += 60) {

// sets the value (range from 0 to 1000):

analogWrite(ledPin, fadeValue);

// wait for 200 milliseconds to see the dimming effect

delay(200);

}

}

if (val > 900) {

analogWrite(ledPin, 0);

// wait for 200 milliseconds to see the dimming effect

delay(200);

}

}

}

**Appendix III**

**General application notes**

1) Capacitor use: The Arduino Uno is a microcontroller board. A microcontroller is a digital device with fast switching edges which uses a large amount of current for a short period of time at each transition. The capacitors serve the amount of current needed to prevent the power supply from sagging during that time(avoid creating noise and malfunction of the logic circuit).From the Arduino UNO Schematic, capacitors are built-in and connected to the pins since the frequency of the switching can vary depending on the load on each pin and the internal logic circuits.

2) Transistor/op-amp use: The UNO is a microcontroller board based on the ATmega328 chip. The chip is limited how much power can sync. It gives 20mg per pin and a 50 mg for all the pins combined. In order to run high current devices like motors and hyper LEDs, we need somehow to amplify the power from the chip and that is for the transistor to come in. When the switch is driving a much higher current load, the pins take control of high current to not to go over the max current. In cases we need to drive an inductive load like motor, transformer, we have to add a protection diode to protect both the transistor and the microcontroller from the high current voltage when they are turned off.

3) - Maximum range of power supply voltage is 6 V - 36 V, with a tolerance of ±3 V -18 V

- Range of allowed input voltage is 3mV

- VCC- is pin 4 and VCC+ is pin 8