An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا المعلومات

Computer Engineering Department
Microcontroller Lab (10636496)
Report Grading Sheet

Instructor Name: Abdallah Rashed	**Experiment #: Arduino Part 2**
Academic Year: 2024	**Performed on: 18-12-2024**
Semester: First semester	**Submitted on: 24/12/2024**
Student Names:	
1- Sama Abusair	2- Marah Hanini
3-	4-
5-	6-

Evaluation Criterion	CLO	Grade	Points
Abstract and Aims Aims and idea of the experiment are clearly stated in simple words		10	
Introduction, Apparatus and Procedures Introduction is complete and well-written, all grammar/spelling correct, Appropriate background information related to the principles of the experiment is provided. The list of apparatus and procedures are also provided		15	
Experimental Results, Calculations and Discussion Results analyzed correctly. Experimental findings adequately and specifically summarized, in graphical, tabular, and/or written form. Comparison of theoretical predictions to experimental results, including discussion of accuracy and error analysis as needed.		50	
Conclusions Conclusions summarize the major findings from the experimental results with adequate specificity. Highlighting the most important results		15	
Appearance Title page is complete, page numbers applied, content is well organized, correct spelling, fonts are consistent, good visual appeal. You have also to use reference for the information you provide		10	
Total		100	

❖ Abstract:

This lab experiment investigates using different interfacing techniques to use the capabilities of microcontrollers. Using three separate modules—a 4-digit, 7-segment LED display, an accelerometer-gyroscope (GY-521), and a liquid crystal display (LCD) with an I2C adapter—the experiment focuses on using Arduino for serial communication tasks. Every assignment tries to show off useful uses and coding methods for connecting these modules to Arduino.

❖ Introduction:

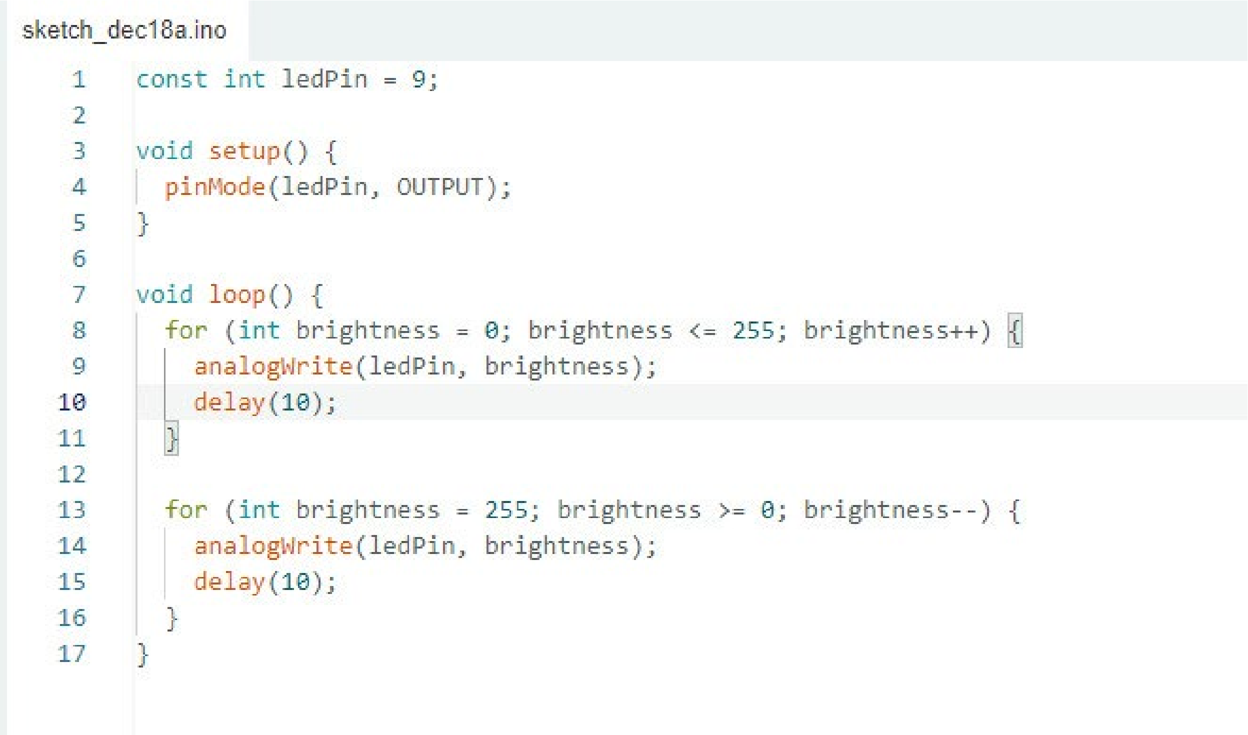
To show the flexibility of Arduino microcontrollers in interacting with external modules, the experiment explores three different tasks. First, a 4-digit, 7-segment LED display is controlled to show the clock timer and BCD counter capabilities using the TM1637 module and we make a Clock. To detect tilting along the x and y axes, the second focus switches to an I2C serial connection with the MPU-6050 accelerometer-gyroscope module (GY-521). Lastly, text display is achieved by using an I2C LCD with a PCF8574 adaptor board, which interfaces with an ultrasonic sensor to calculate distance. Coding using Arduino, making use of pertinent libraries, comprehending module requirements, and putting in place suitable communication protocols are all required for each assignment. Through these exercises, participants can increase their understanding of microcontroller applications in various projects by gaining practical experience in interfacing different modules.

Materials and Methods:

- *Arduino Mega with USB Cable*
- *LCD*
- *4-digit 7-segment LED display*
- *Accelerometer- gyroscope GY-521*
- *Arduino IDE*
- *Liquid Crystal Display with I2C*

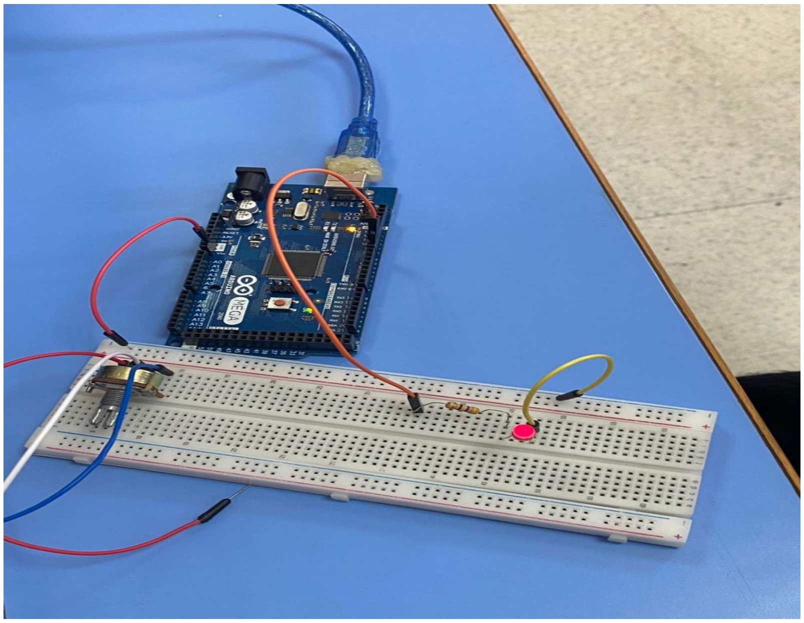
Part1 :

Automatically increase the LED light intensity and then decrease it continuously.

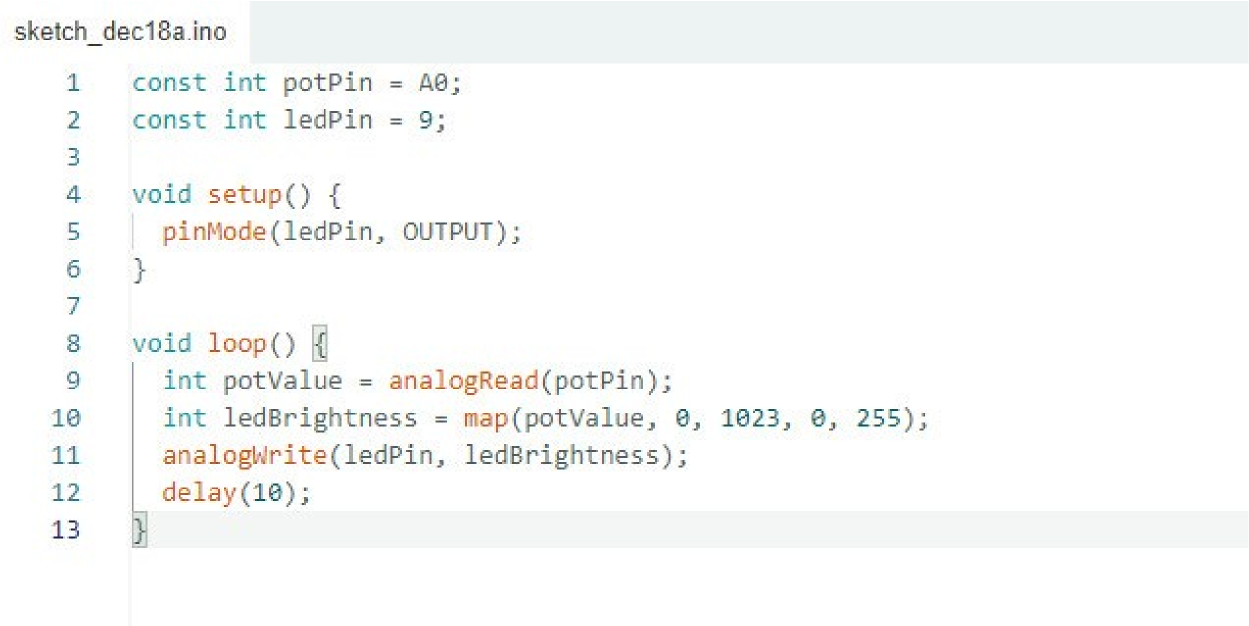

sketch_dec18a.ino

```
const int ledPin = 9;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  for (int brightness = 0; brightness <= 255; brightness++) {
    analogWrite(ledPin, brightness);
    delay(10);
  }

  for (int brightness = 255; brightness >= 0; brightness--) {
    analogWrite(ledPin, brightness);
    delay(10);
  }
}
```

Result:



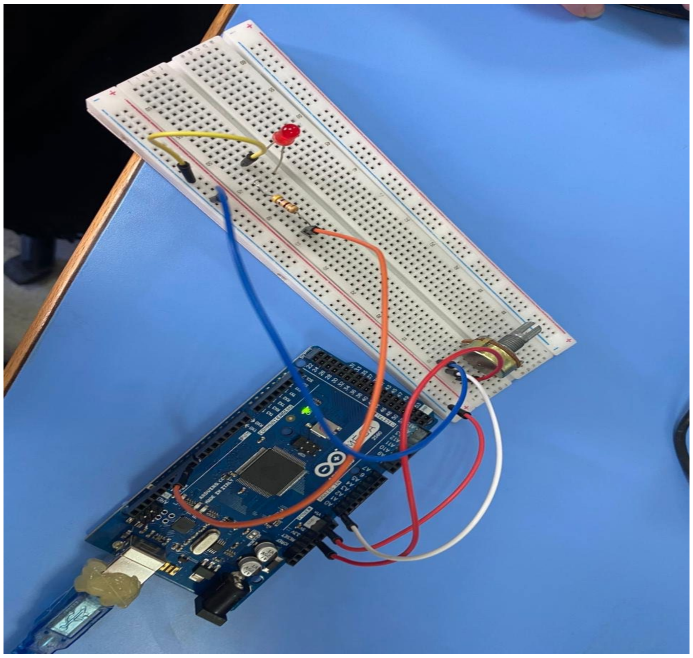
Part 2: Control the intensity of light by a Potentiometer



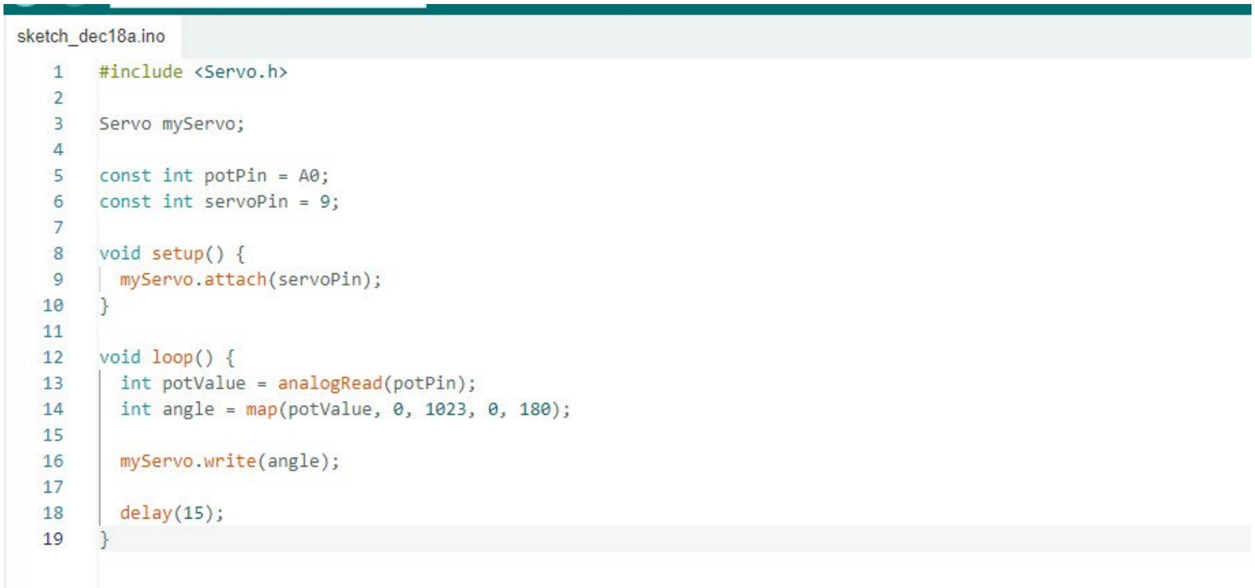
sketch_dec18a.ino

```
const int potPin = A0;
const int ledPin = 9;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int potValue = analogRead(potPin);
  int ledBrightness = map(potValue, 0, 1023, 0, 255);
  analogWrite(ledPin, ledBrightness);
  delay(10);
}
```

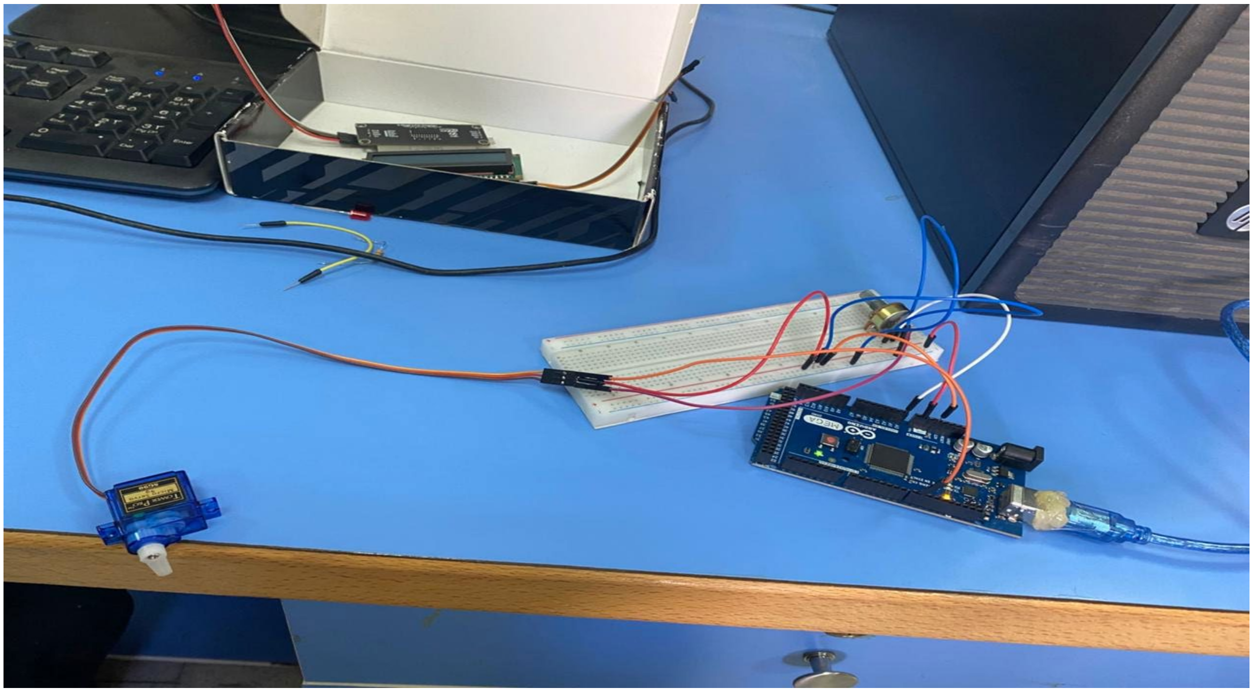
Result:



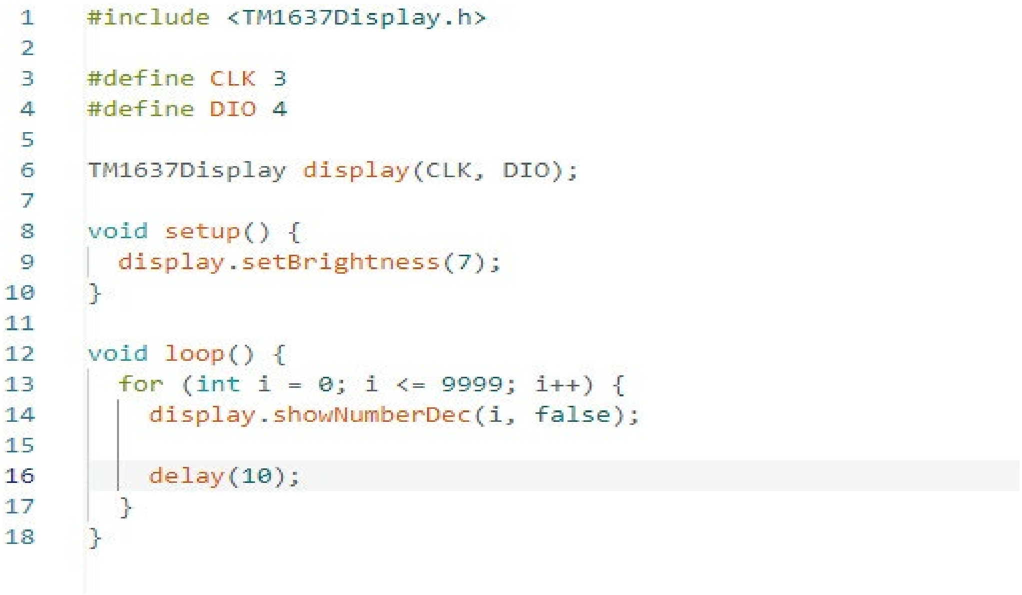
Part 3: Use the potentiometer to move the servo. If it moves to the right, the servo moves with it. If it moves to the left, the servo moves with it.



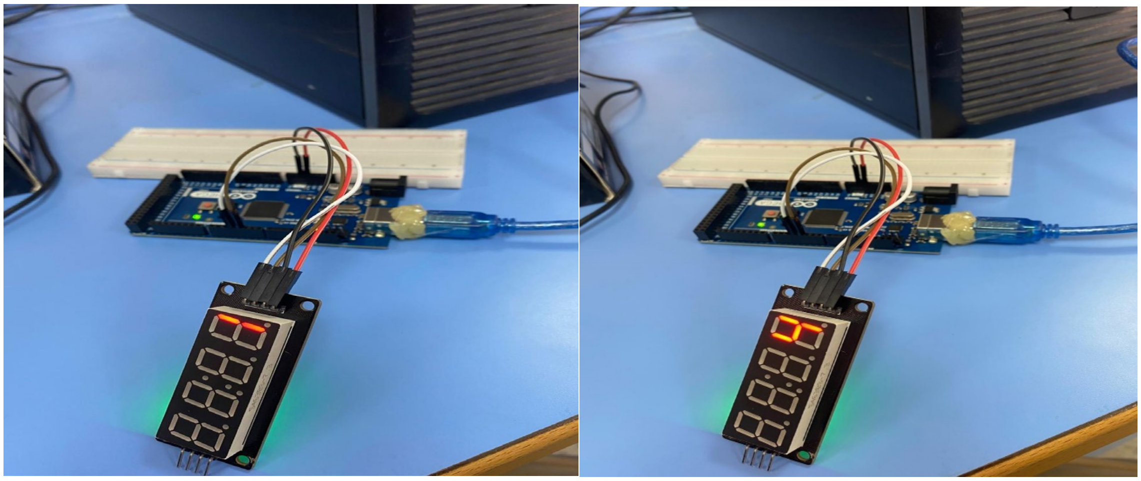
```
#include <Servo.h>

Servo myServo;

const int potPin = A0;
const int servoPin = 9;

void setup() {
  myServo.attach(servoPin);
}

void loop() {
  int potValue = analogRead(potPin);
  int angle = map(potValue, 0, 1023, 0, 180);

  myServo.write(angle);

  delay(15);
}
```



Task 1 : display a 0-9999 BCD counter on the 4-digit LED display.

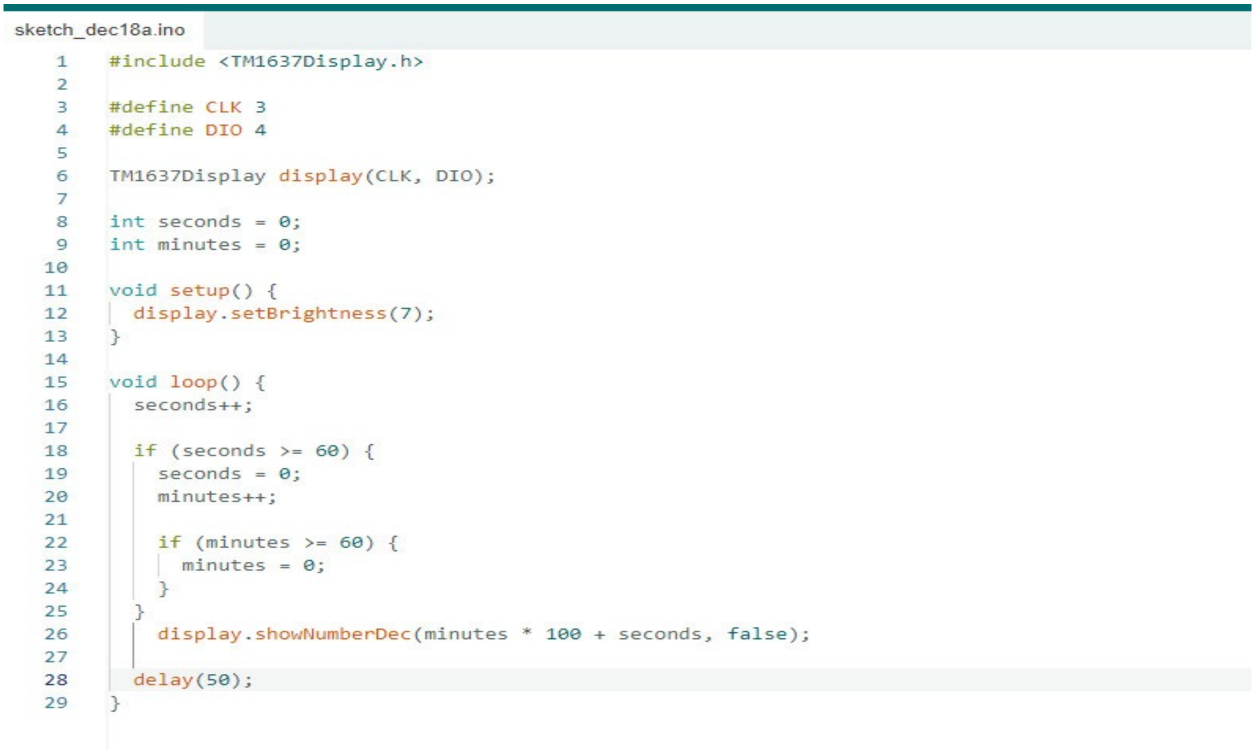


```
#include <TM1637Display.h>

#define CLK 3
#define DIO 4

TM1637Display display(CLK, DIO);

void setup() {
  display.setBrightness(7);
}

void loop() {
  for (int i = 0; i <= 9999; i++) {
    display.showNumberDec(i, false);

    delay(10);
  }
}
```

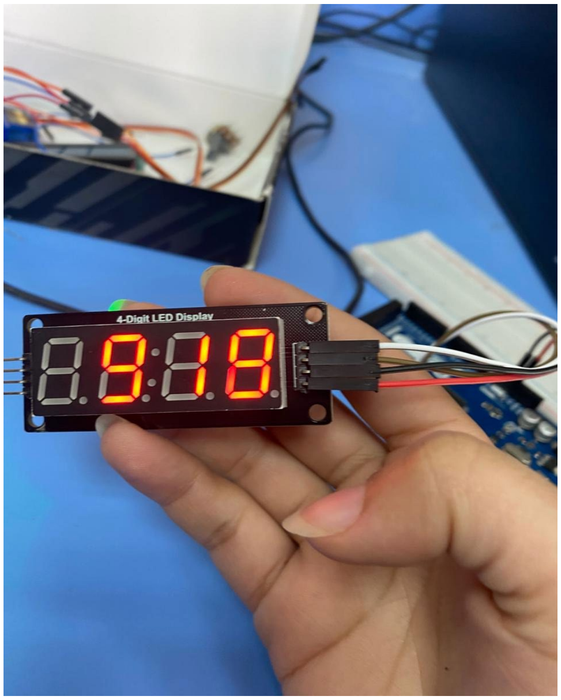


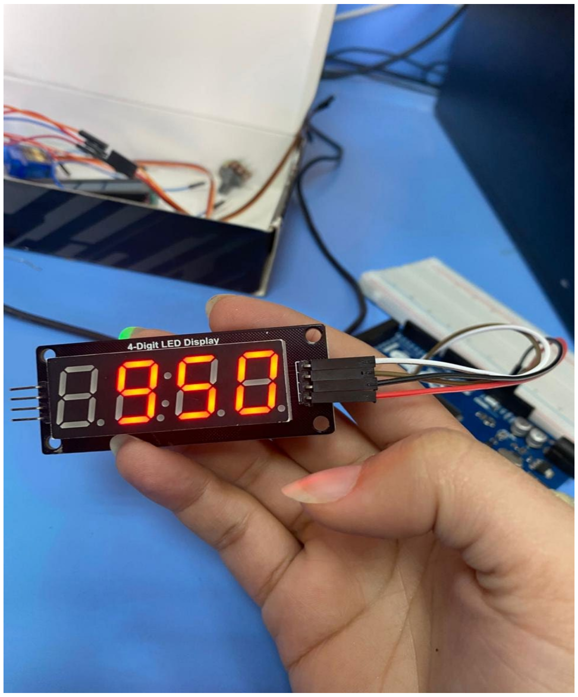
Task2 :

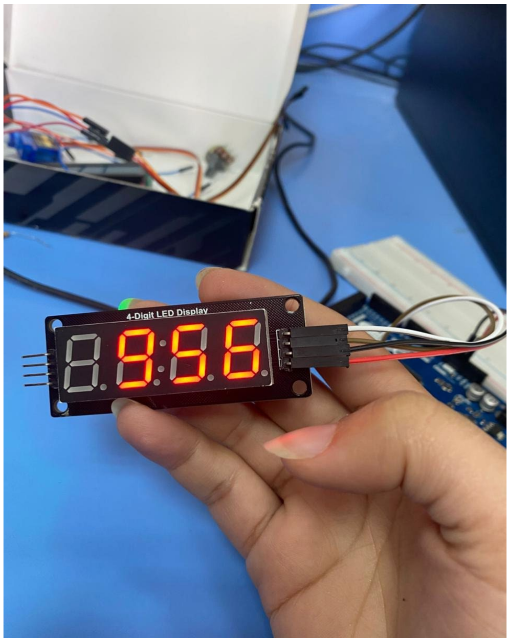
display a clock timer on the display to count minutes and seconds. Each count from 0 to 59. The seconds are displayed on rightmost digits and the minutes on the leftmost digits.

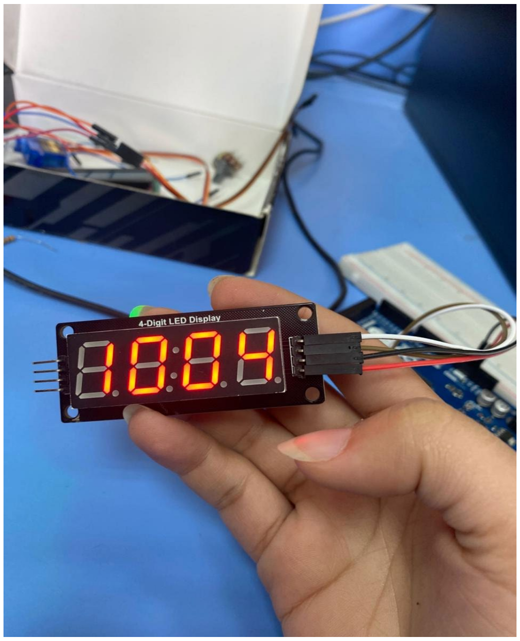


```
#include <TM1637Display.h>

#define CLK 3
#define DIO 4

TM1637Display display(CLK, DIO);

int seconds = 0;
int minutes = 0;

void setup() {
  display.setBrightness(7);
}

void loop() {
  seconds++;

  if (seconds >= 60) {
    seconds = 0;
    minutes++;

    if (minutes >= 60) {
      minutes = 0;
    }
  }
    display.showNumberDec(minutes * 100 + seconds, false);

  delay(50);
}
```

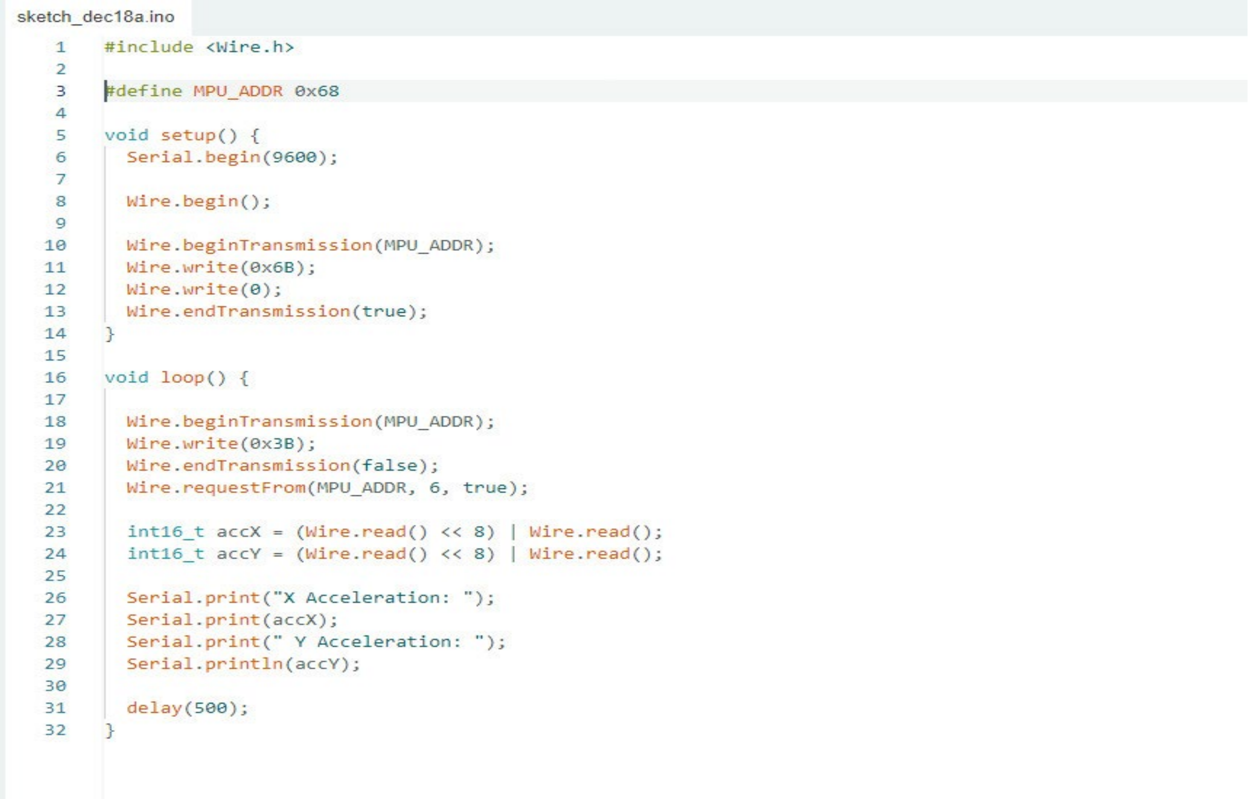






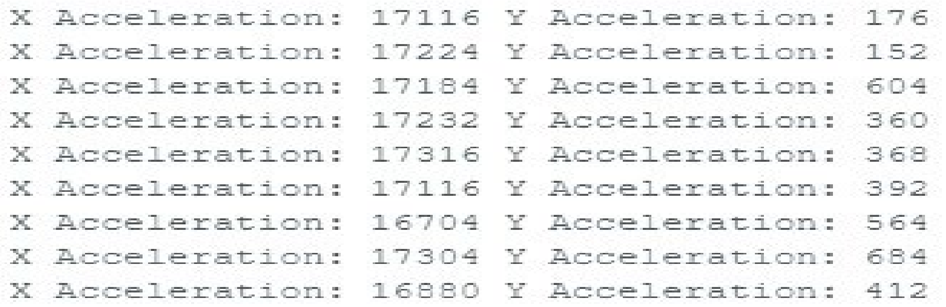


Task 3: detect the tilting of the module over the x and y axes using the GY-521 sensor.

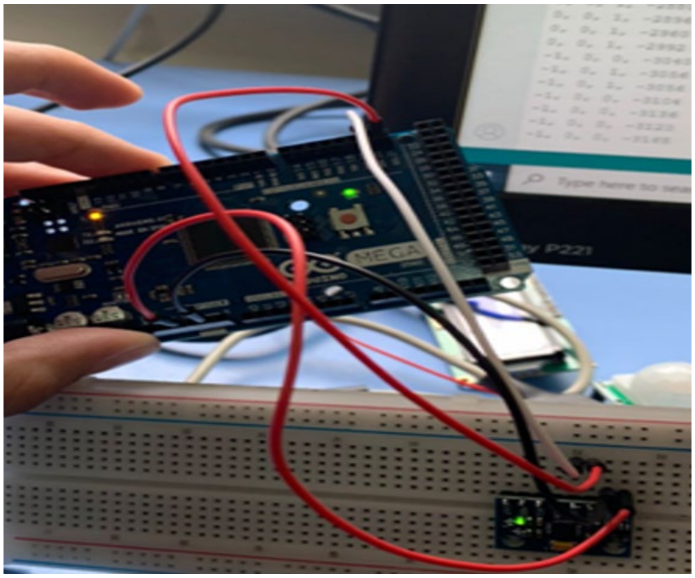


```
#include <Wire.h>

#define MPU_ADDR 0x68

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

  Wire.begin();

  Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
}

void loop() {

  Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_ADDR, 6, true);

  int16_t accX = (Wire.read() << 8) | Wire.read();
  int16_t accY = (Wire.read() << 8) | Wire.read();

  Serial.print("X Acceleration: ");
  Serial.print(accX);
  Serial.print(" Y Acceleration: ");
  Serial.println(accY);

  delay(500);
}
```

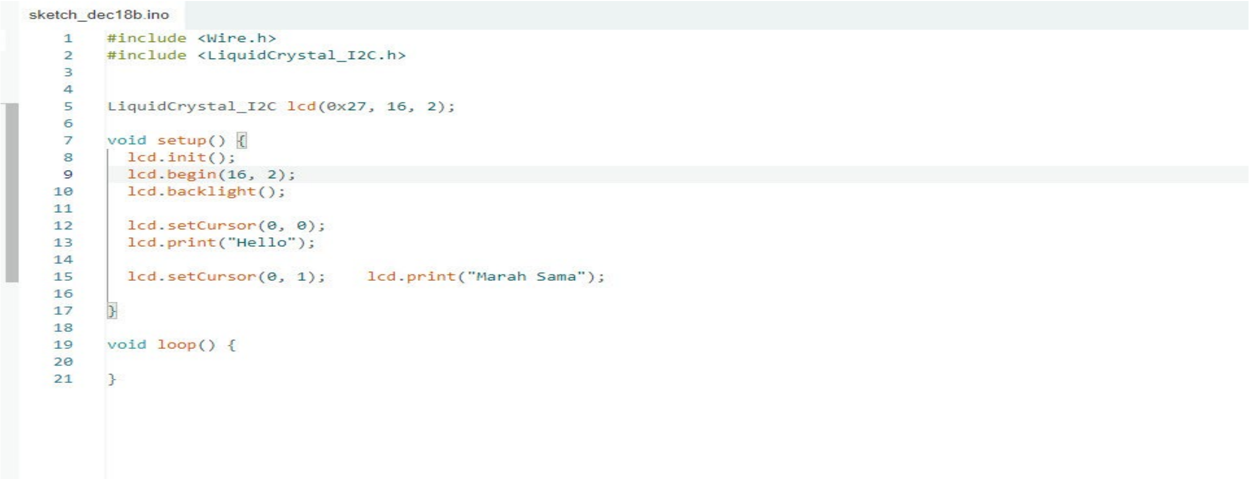
Result:



```
X Acceleration: 17116 Y Acceleration: 176
X Acceleration: 17224 Y Acceleration: 152
X Acceleration: 17184 Y Acceleration: 604
X Acceleration: 17232 Y Acceleration: 360
X Acceleration: 17316 Y Acceleration: 368
X Acceleration: 17116 Y Acceleration: 392
X Acceleration: 16704 Y Acceleration: 564
X Acceleration: 17304 Y Acceleration: 684
X Acceleration: 16880 Y Acceleration: 412
```

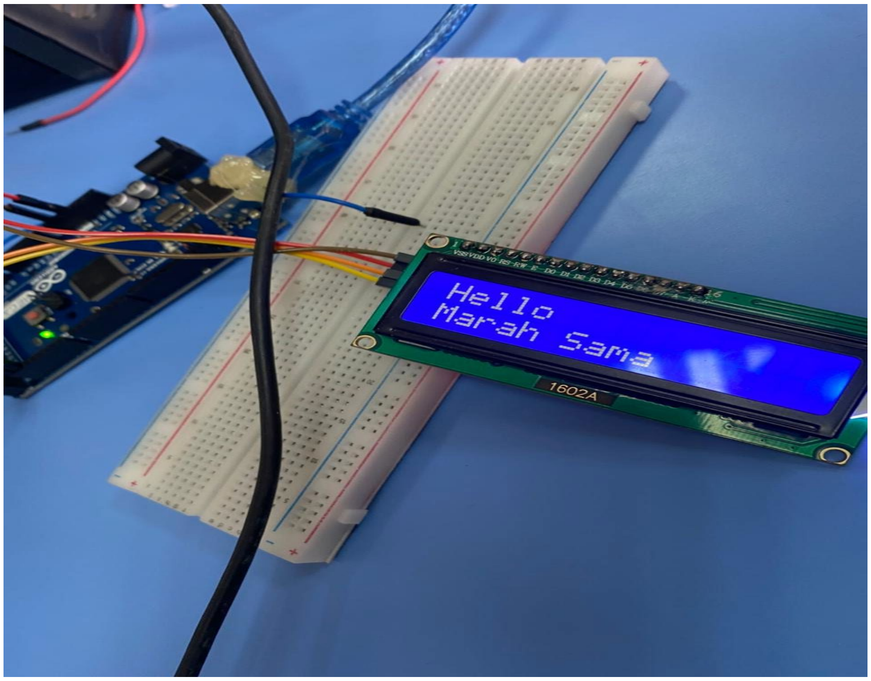


Task 4: Display Hello message on the first line and your name on the second line.

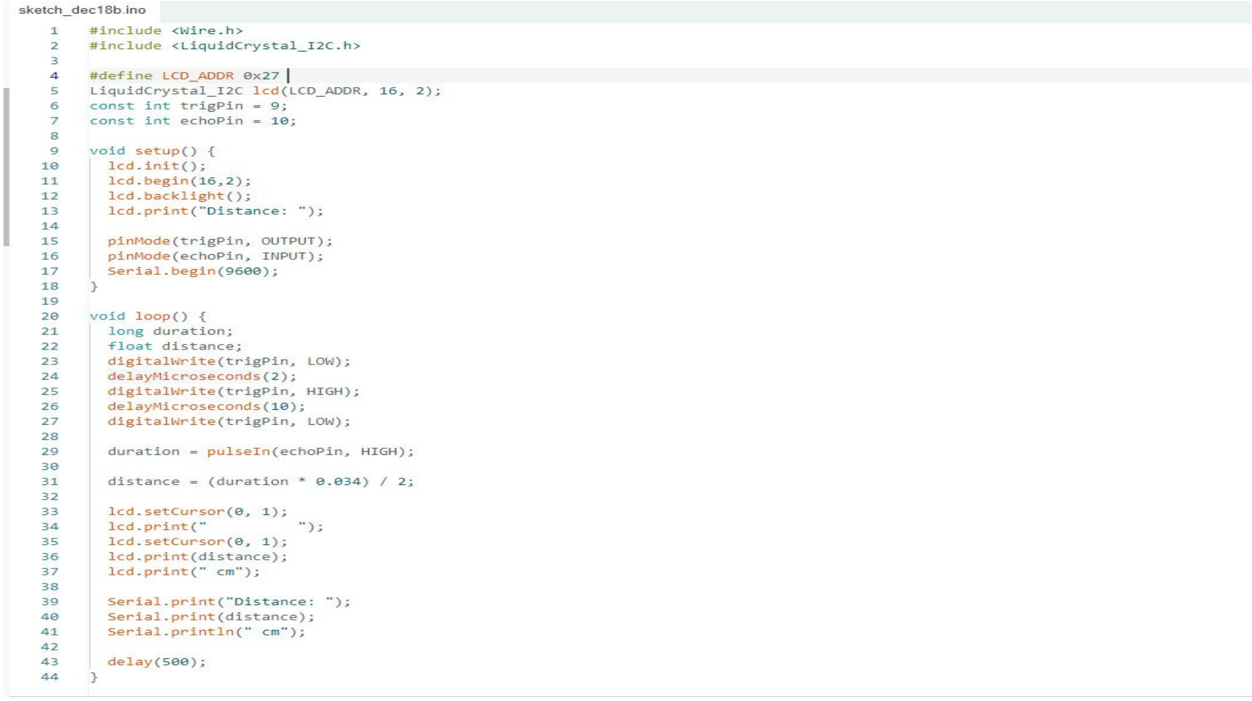


sketch_dec18b.ino

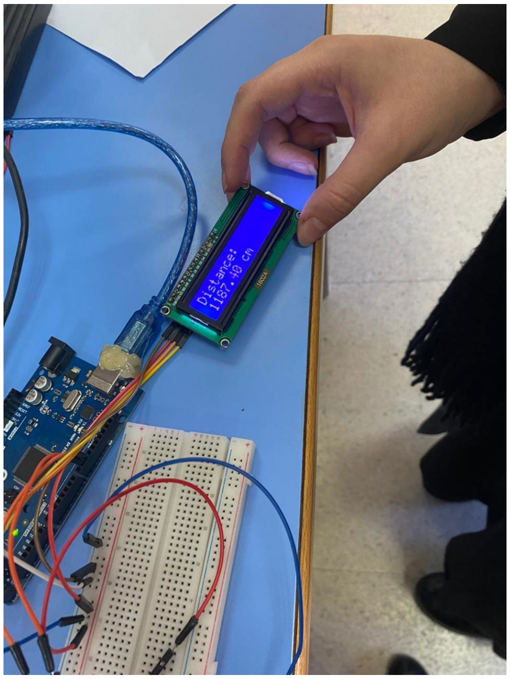
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>


LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.init();
  lcd.begin(16, 2);
  lcd.backlight();

  lcd.setCursor(0, 0);
  lcd.print("Hello");

  lcd.setCursor(0, 1);    lcd.print("Marah Sama");

}

void loop() {

}
```

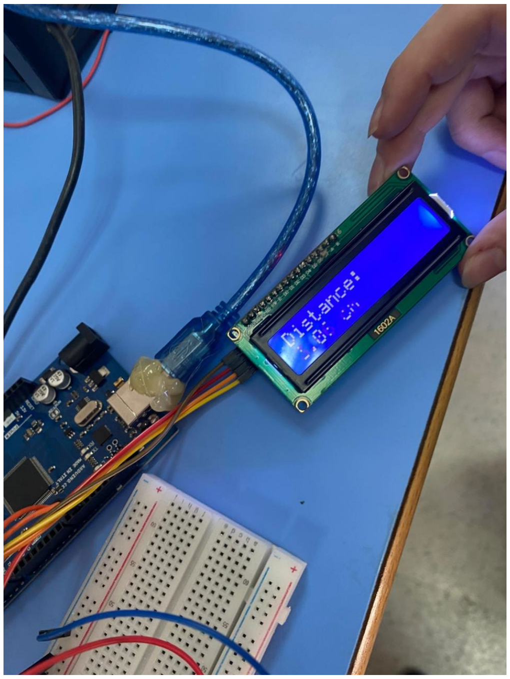


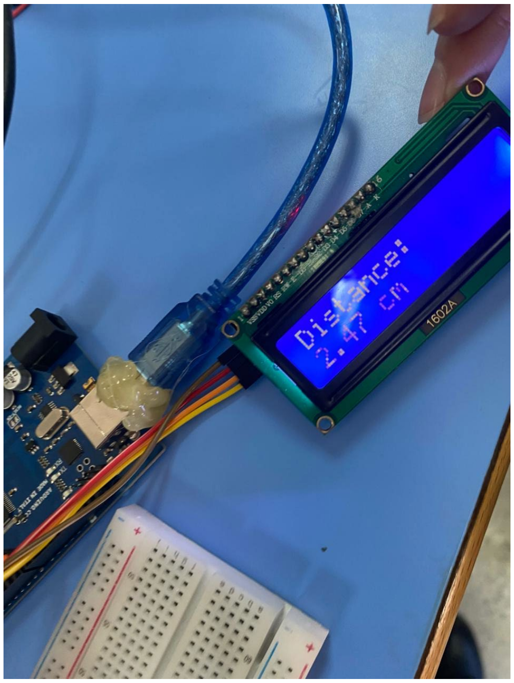
Task 5: Display distance on LCD by using the ultrasonic sensor.

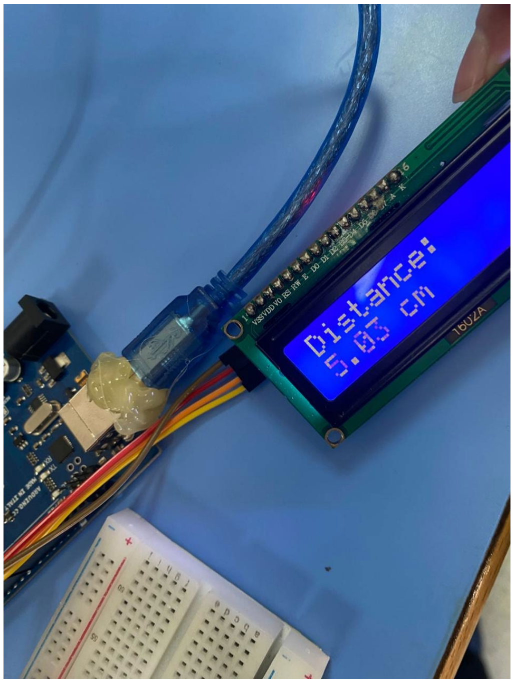


```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define LCD_ADDR 0x27
LiquidCrystal_I2C lcd(LCD_ADDR, 16, 2);
const int trigPin = 9;
const int echoPin = 10;

void setup() {
  lcd.init();
  lcd.begin(16,2);
  lcd.backlight();
  lcd.print("Distance: ");

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  long duration;
  float distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  distance = (duration * 0.034) / 2;

  lcd.setCursor(0, 1);
  lcd.print("          ");
  lcd.setCursor(0, 1);
  lcd.print(distance);
  lcd.print(" cm");

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  delay(500);
}
```









❖ Conclusions:

In conclusion, this lab demonstrated how Arduino microcontrollers can communicate with a variety of modules, including LCDs, accelerometer-gyroscopes, and LED displays. Students learned about coding and communication protocols through hands-on projects, highlighting the many uses of microcontrollers in practical applications.