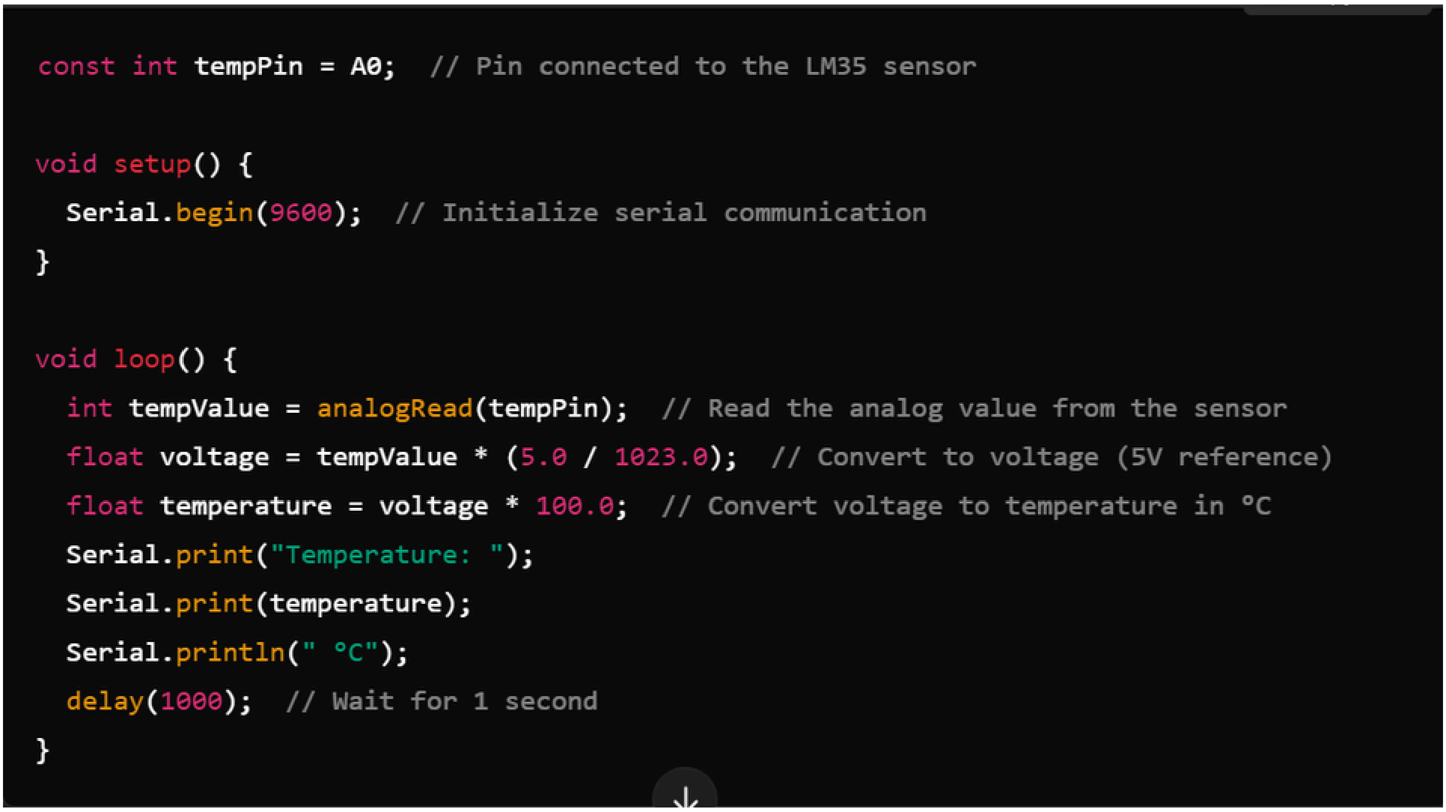
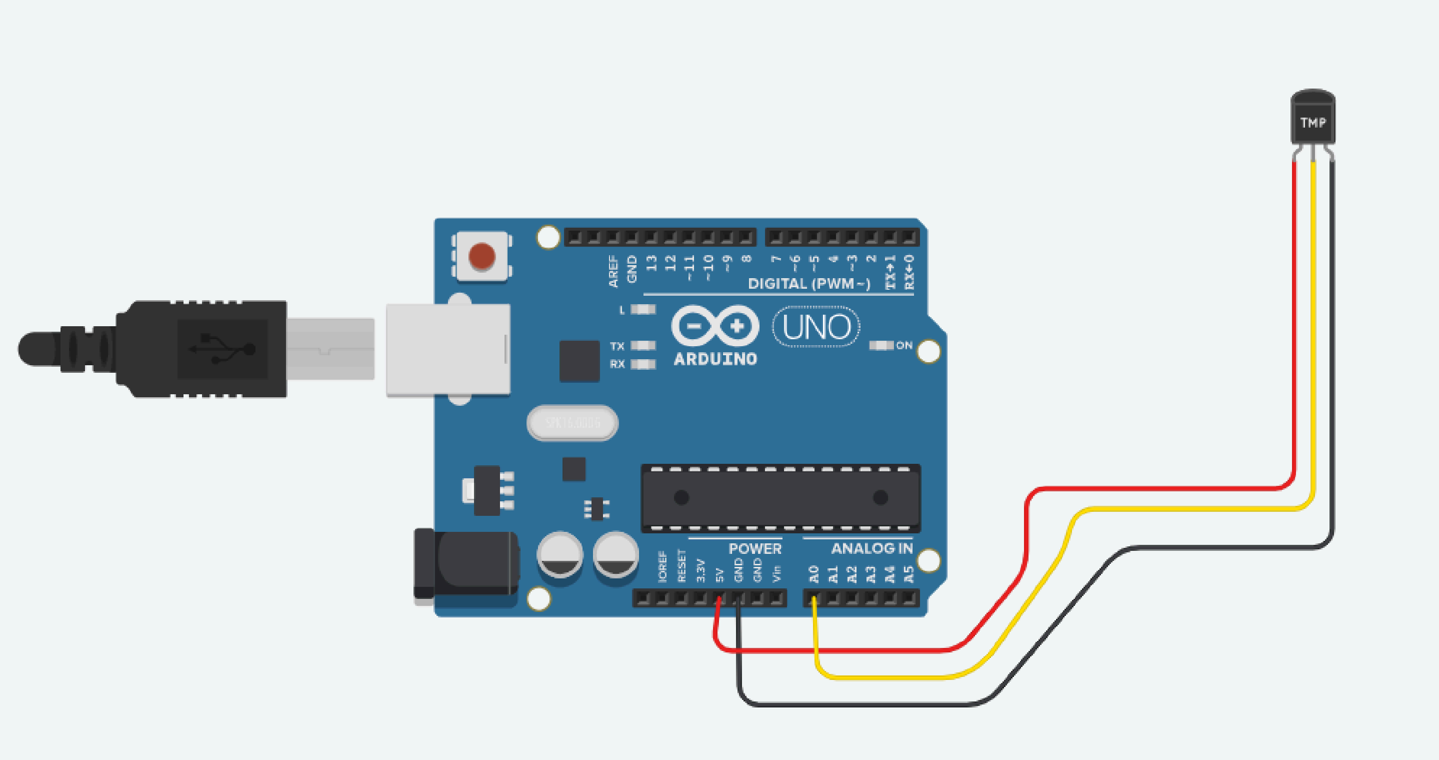
MORE PROJECTS

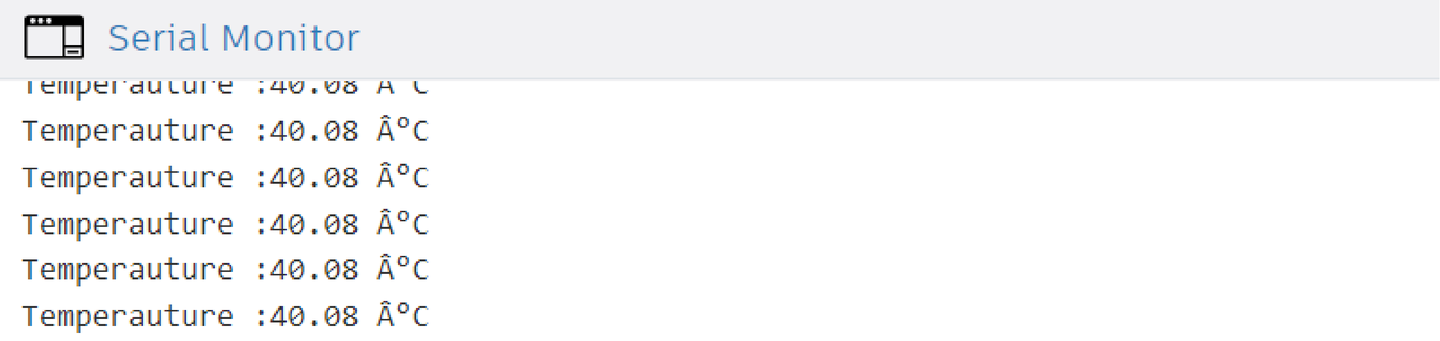
6.Temperature Sensor (LM35)

This program reads the temperature from an LM35 sensor and sends the data to the serial monitor.



```
const int tempPin = A0;  // Pin connected to the LM35 sensor

void setup() {
  Serial.begin(9600);  // Initialize serial communication
}

void loop() {
  int tempValue = analogRead(tempPin);  // Read the analog value from the sensor
  float voltage = tempValue * (5.0 / 1023.0);  // Convert to voltage (5V reference)
  float temperature = voltage * 100.0;  // Convert voltage to temperature in °C
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");
  delay(1000);  // Wait for 1 second
}
```

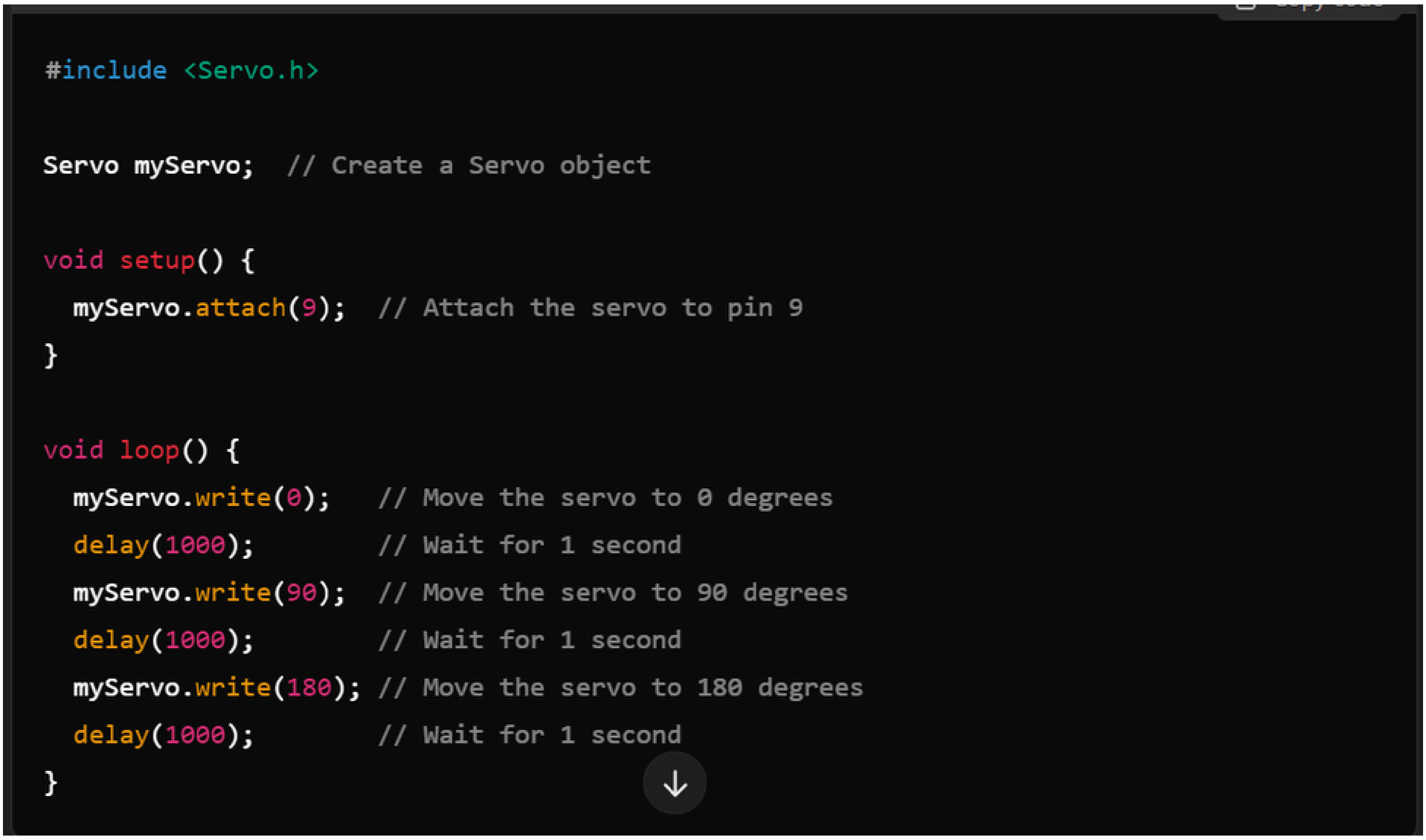




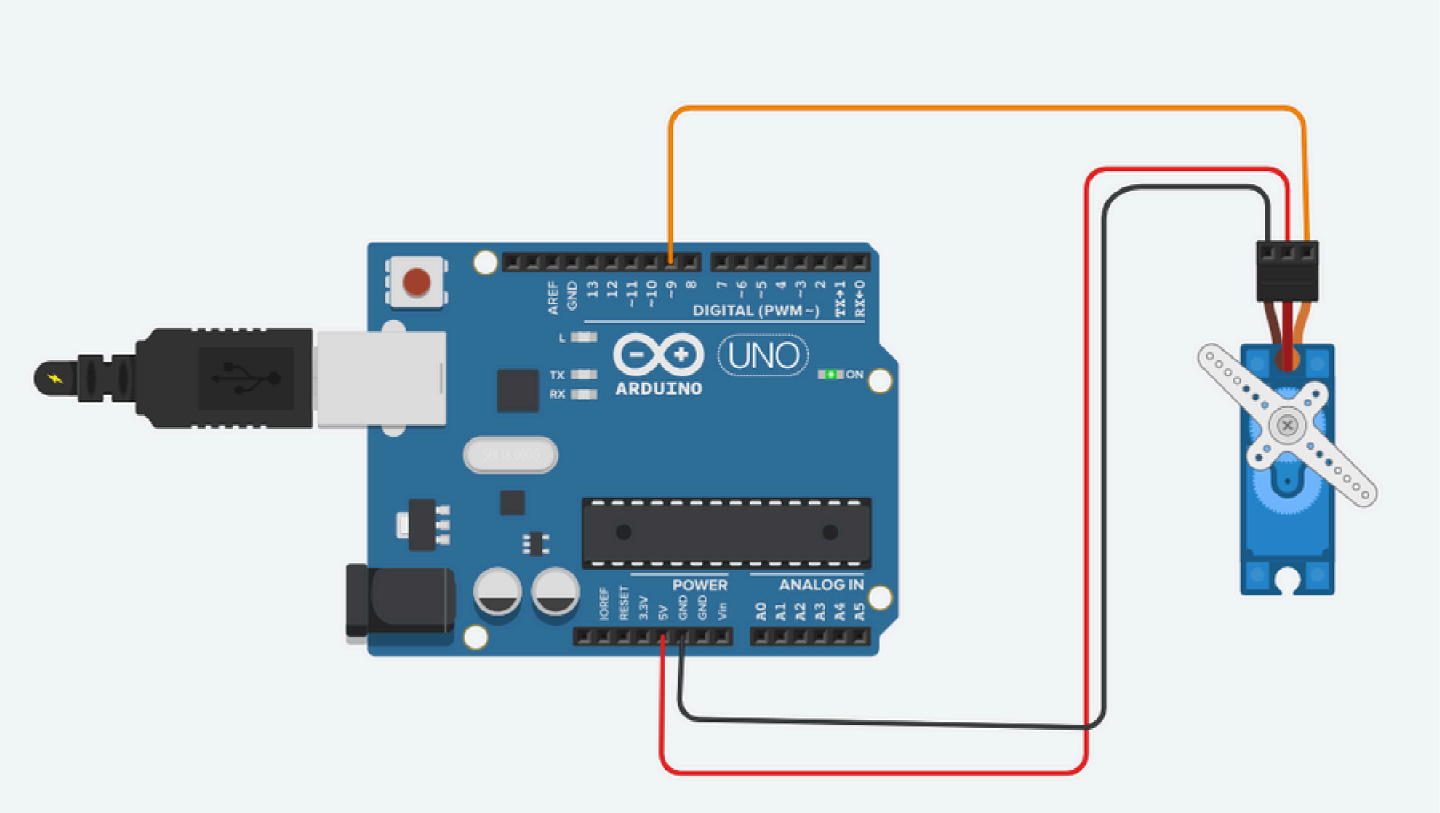
```
Temperauture :40.08 Â°C
Temperauture :40.08 Â°C
Temperauture :40.08 Â°C
Temperauture :40.08 Â°C
Temperauture :40.08 Â°C
Temperauture :40.08 Â°C
```

7.Controlling a Servo Motor

This program controls a servo motor to move it to different angles using the Servo library.

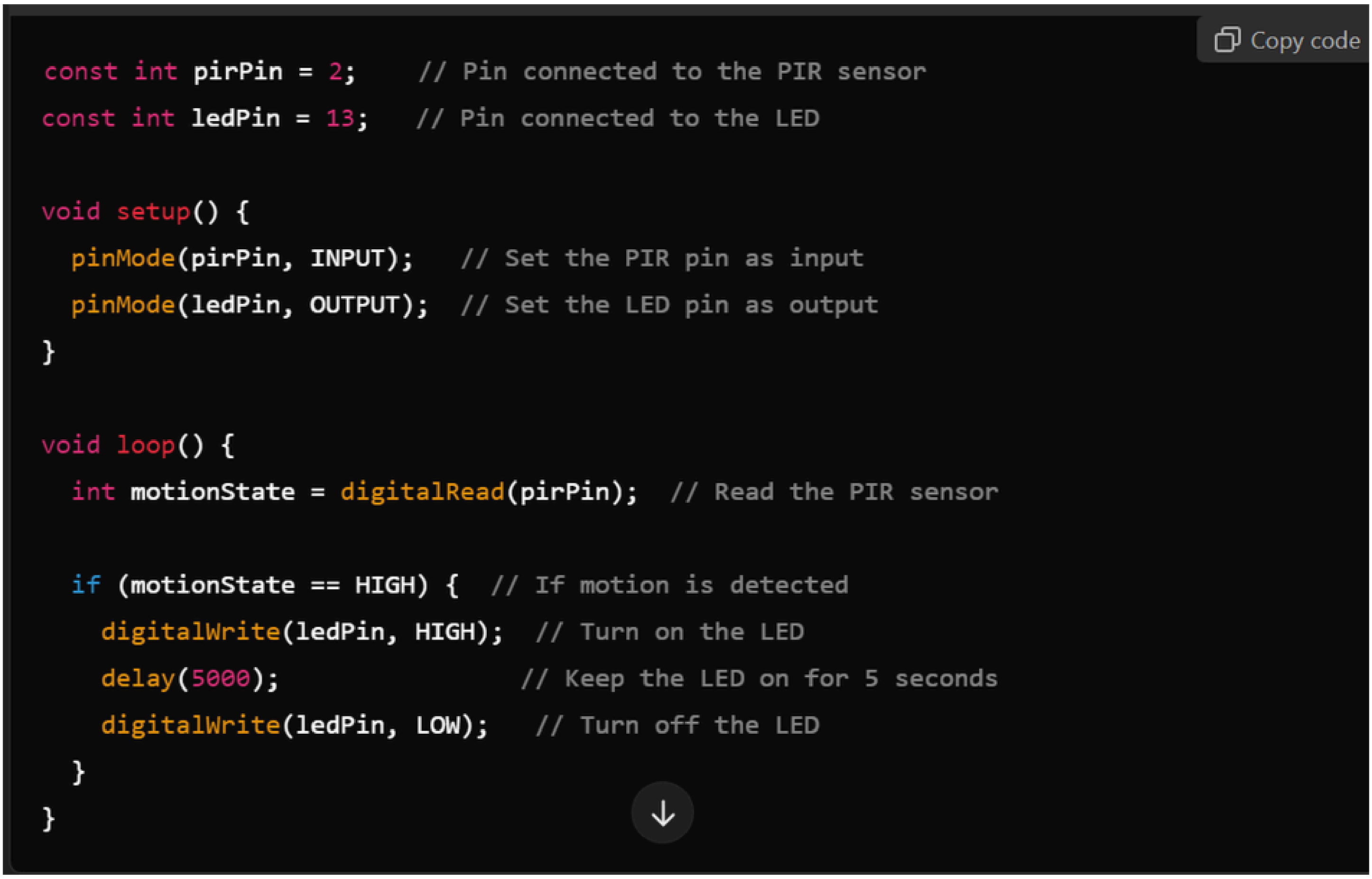


```
#include <Servo.h>

Servo myServo;  // Create a Servo object

void setup() {
  myServo.attach(9);  // Attach the servo to pin 9
}

void loop() {
  myServo.write(0);   // Move the servo to 0 degrees
  delay(1000);        // Wait for 1 second
  myServo.write(90);  // Move the servo to 90 degrees
  delay(1000);        // Wait for 1 second
  myServo.write(180); // Move the servo to 180 degrees
  delay(1000);        // Wait for 1 second
}
```

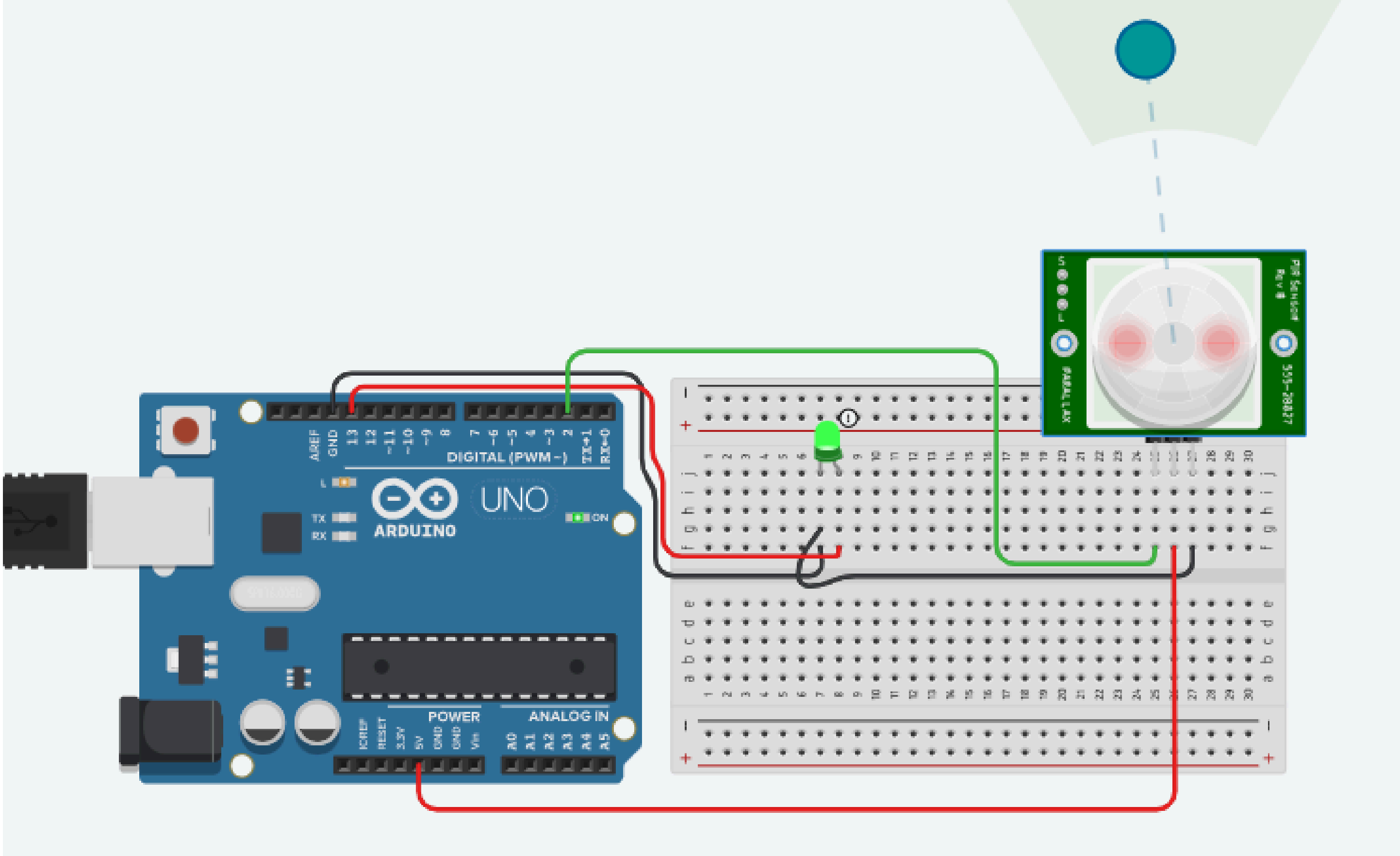


8.Motion Detection using PIR Sensor

This program uses a Passive Infrared (PIR) sensor to detect motion and control an LED.

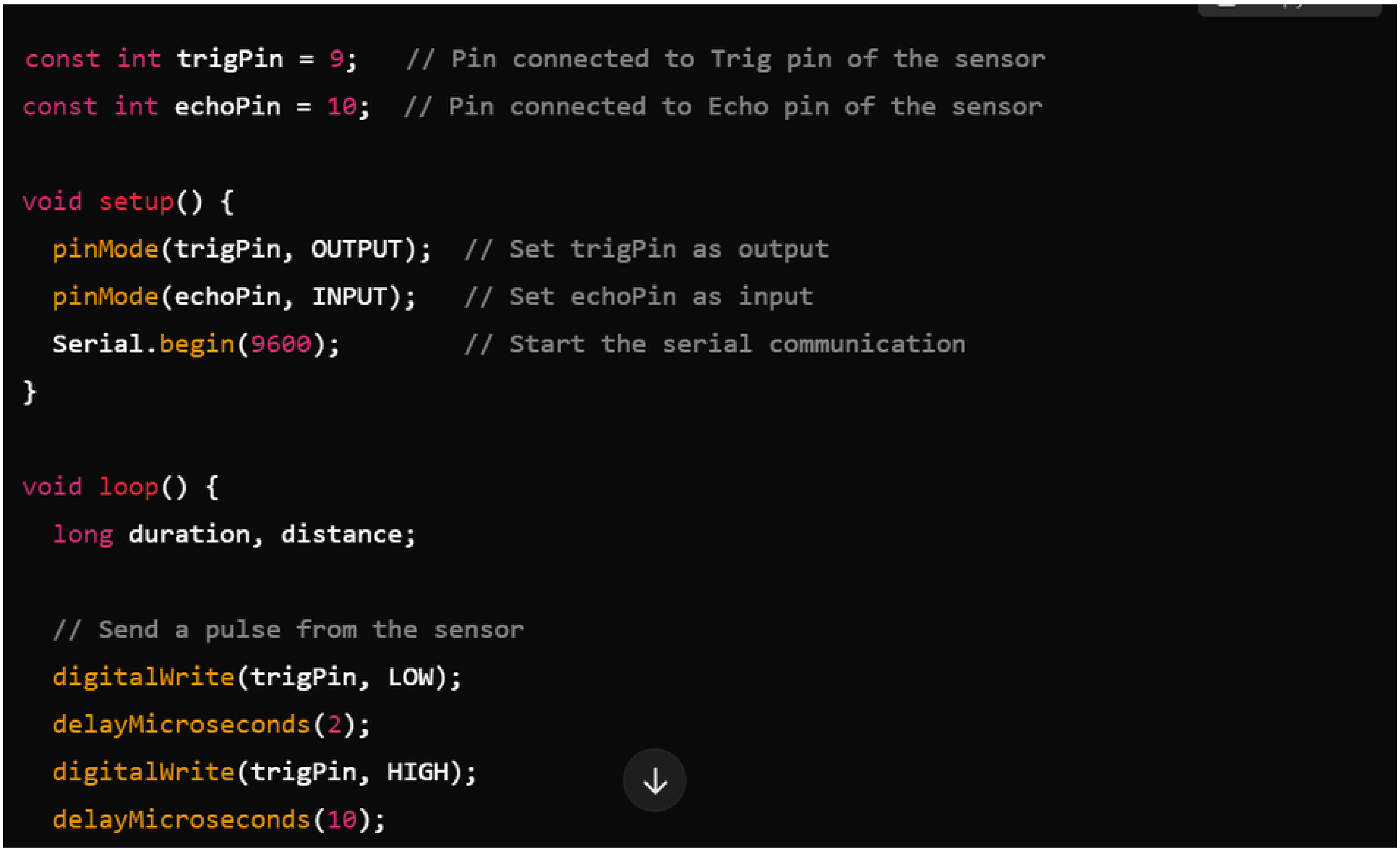


```
const int pirPin = 2;     // Pin connected to the PIR sensor
const int ledPin = 13;    // Pin connected to the LED

void setup() {
  pinMode(pirPin, INPUT);    // Set the PIR pin as input
  pinMode(ledPin, OUTPUT);   // Set the LED pin as output
}

void loop() {
  int motionState = digitalRead(pirPin);   // Read the PIR sensor

  if (motionState == HIGH) {   // If motion is detected
    digitalWrite(ledPin, HIGH);   // Turn on the LED
    delay(5000);                   // Keep the LED on for 5 seconds
    digitalWrite(ledPin, LOW);    // Turn off the LED
  }
}
```

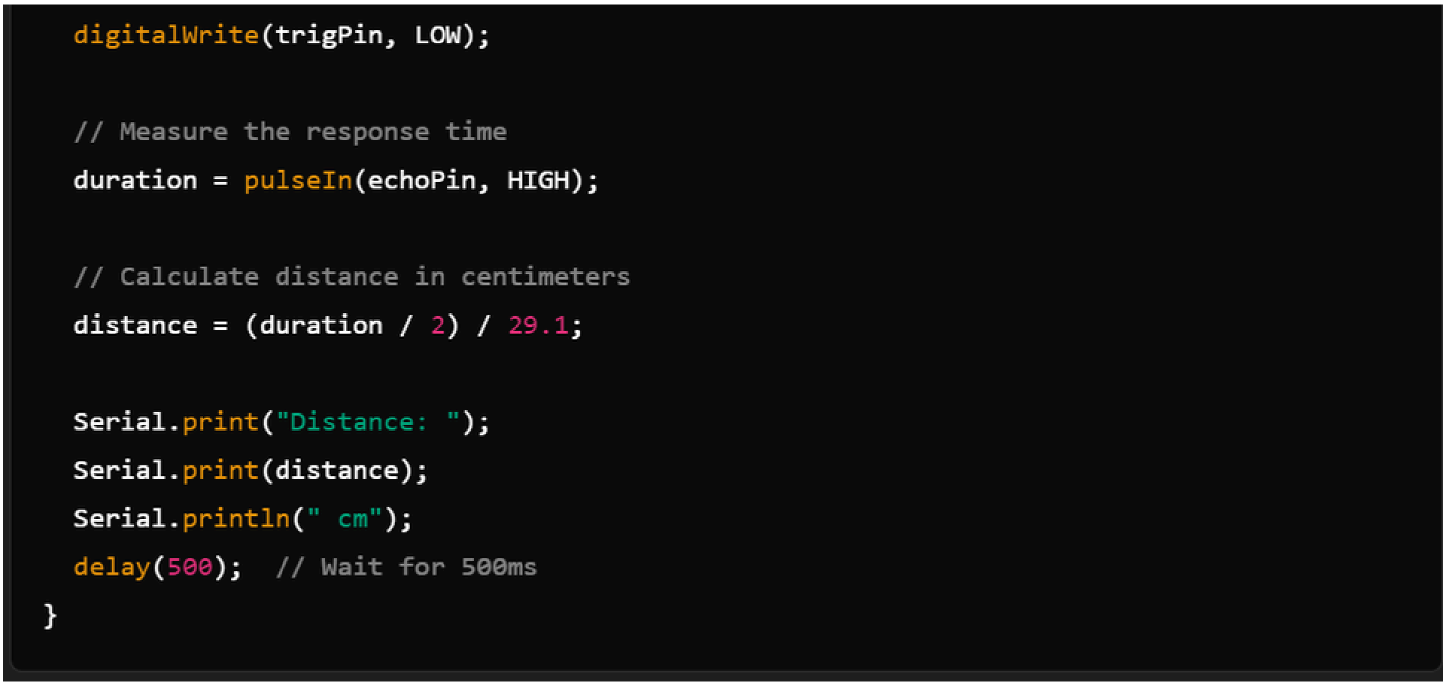


9.Ultrasonic Sensor (HC-SR04) for Distance Measurement

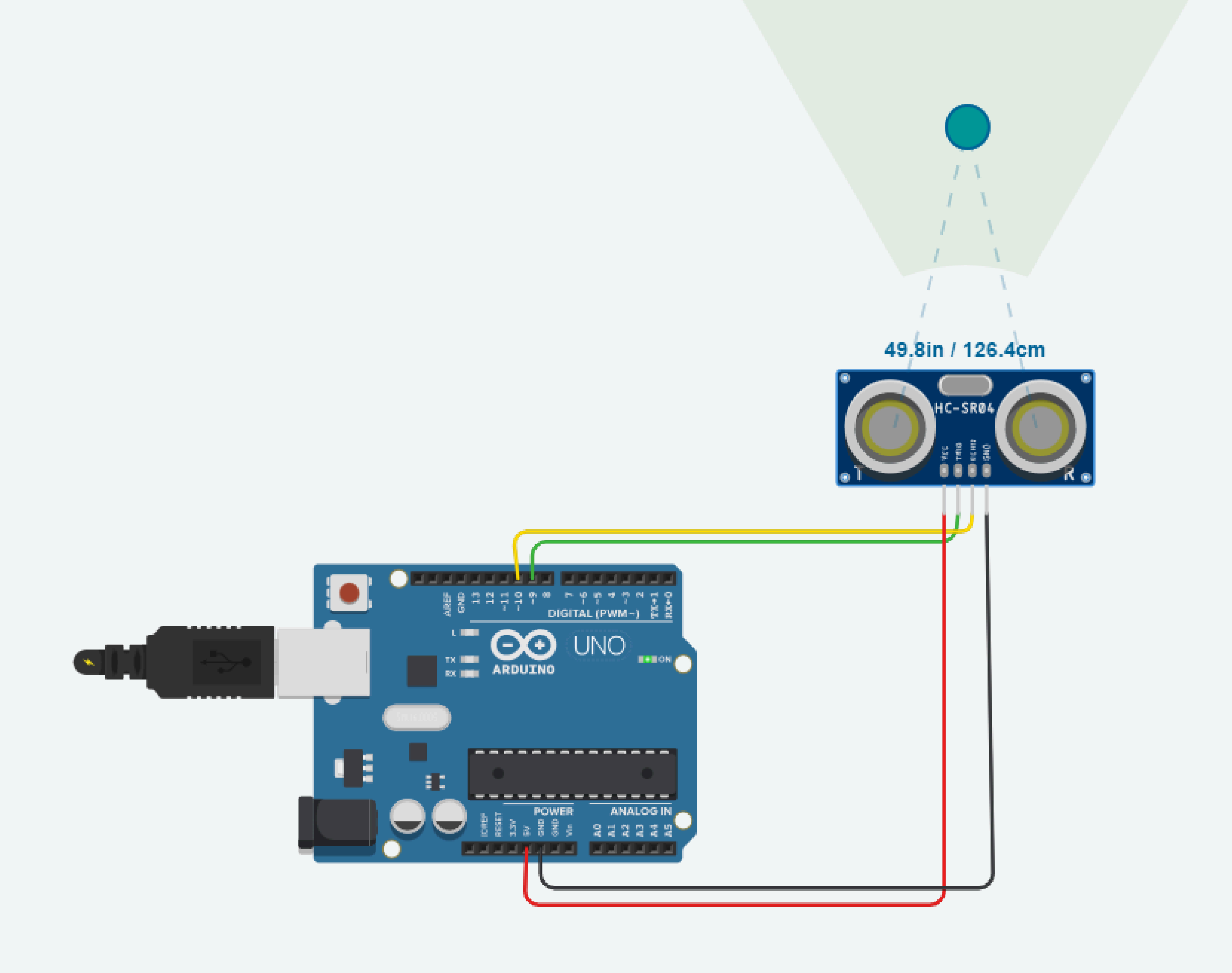
This program measures the distance using an ultrasonic sensor and displays it on the serial monitor.

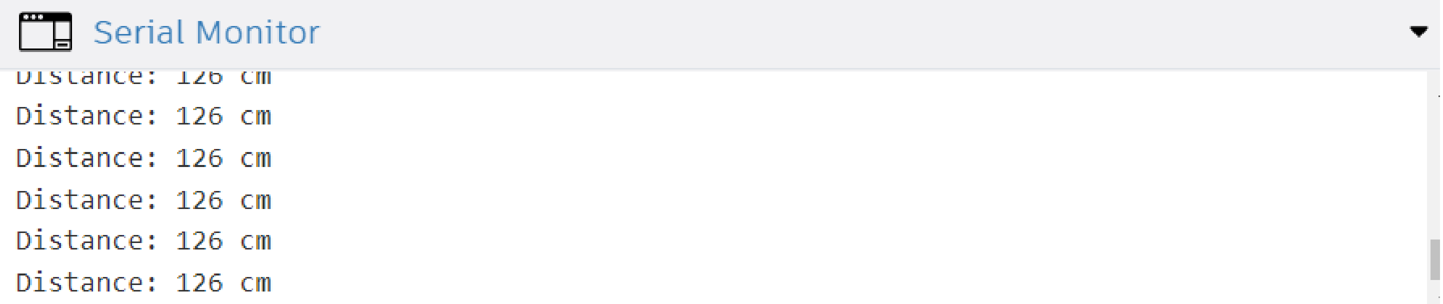


```
const int trigPin = 9;   // Pin connected to Trig pin of the sensor
const int echoPin = 10;  // Pin connected to Echo pin of the sensor

void setup() {
  pinMode(trigPin, OUTPUT);  // Set trigPin as output
  pinMode(echoPin, INPUT);   // Set echoPin as input
  Serial.begin(9600);        // Start the serial communication
}

void loop() {
  long duration, distance;

  // Send a pulse from the sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
```



```
  digitalWrite(trigPin, LOW);

  // Measure the response time
  duration = pulseIn(echoPin, HIGH);

  // Calculate distance in centimeters
  distance = (duration / 2) / 29.1;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);  // Wait for 500ms
}
```

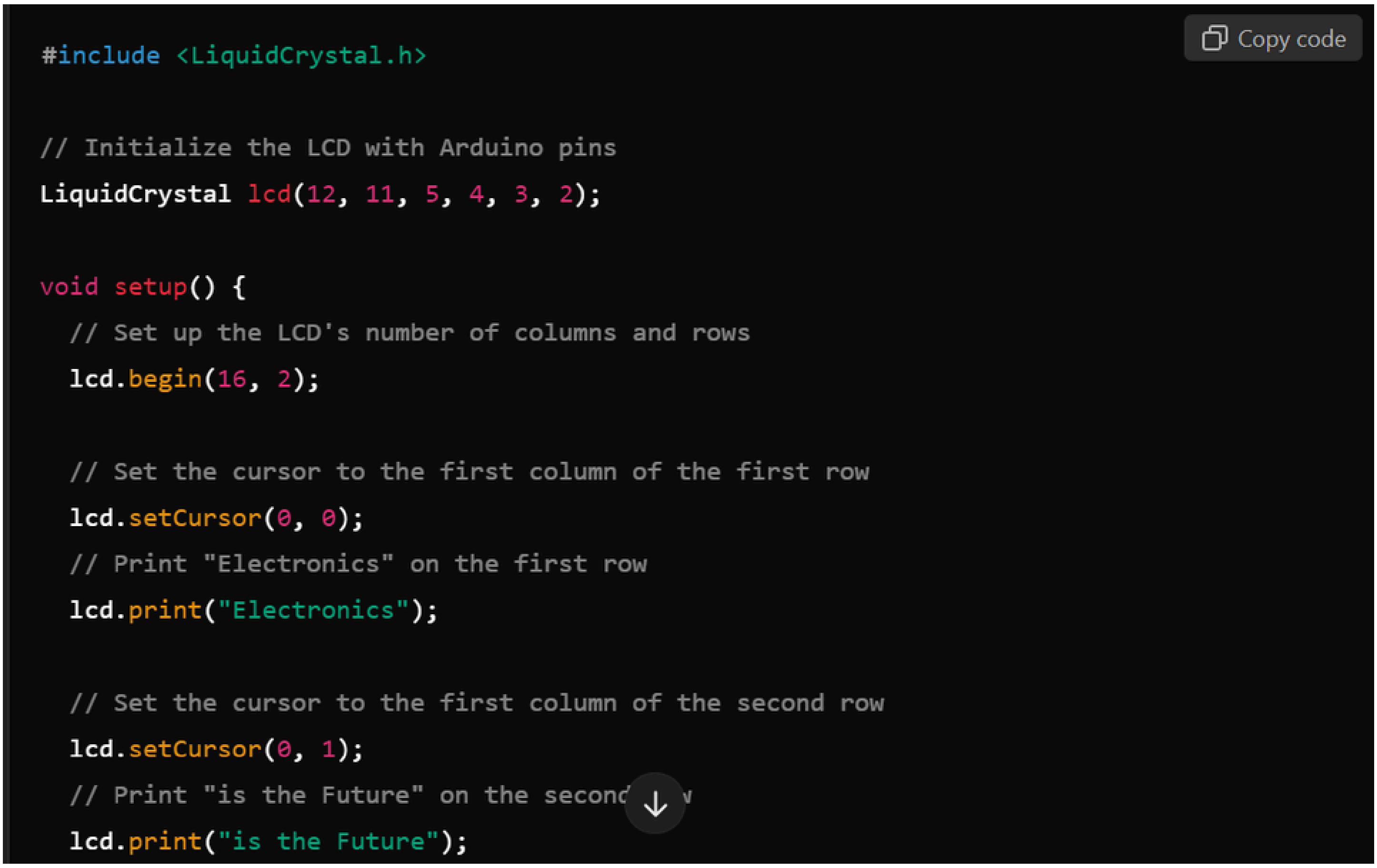




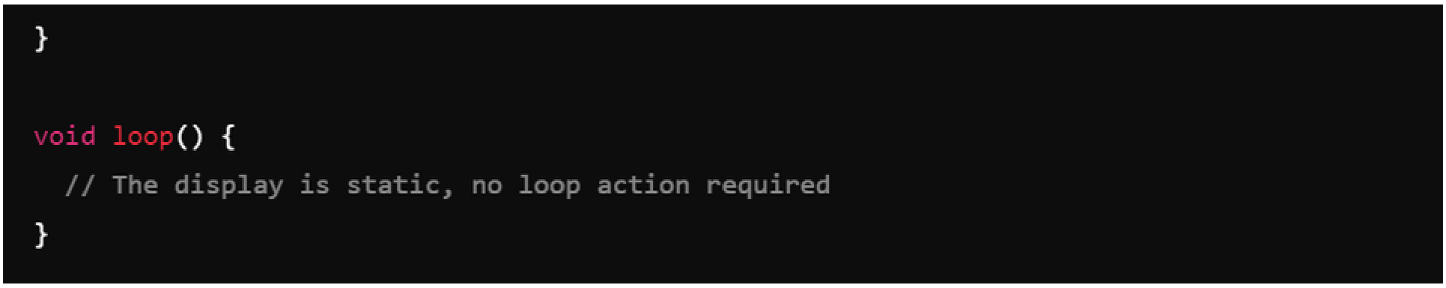
```
Distance: 126 cm
Distance: 126 cm
Distance: 126 cm
Distance: 126 cm
Distance: 126 cm
Distance: 126 cm
```

10.Displaying Text on 16x2 LCD

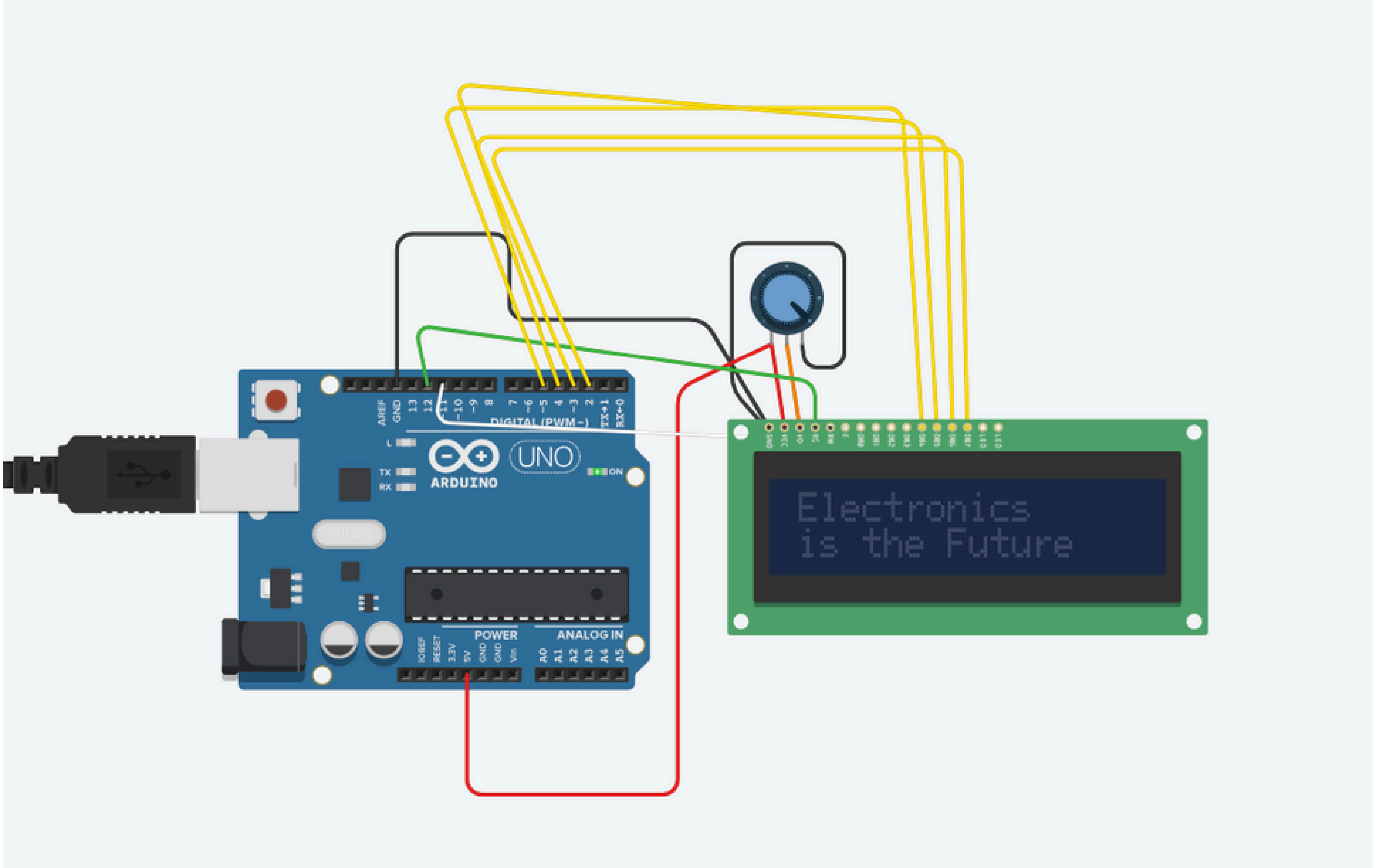
This program displays the text "Electronics" on the first row and "is the Future" on the second row of a 16x2 LCD.



```
#include <LiquidCrystal.h>

// Initialize the LCD with Arduino pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // Set up the LCD's number of columns and rows
  lcd.begin(16, 2);

  // Set the cursor to the first column of the first row
  lcd.setCursor(0, 0);
  // Print "Electronics" on the first row
  lcd.print("Electronics");

  // Set the cursor to the first column of the second row
  lcd.setCursor(0, 1);
  // Print "is the Future" on the second row
  lcd.print("is the Future");
```



```
}

void loop() {
  // The display is static, no loop action required
}
```



Note :- In Program -> 7 and 10 we have used header files which is most important part of Arduino Software we will Explain header files and some examples of header files in next part.

THANK YOU

HAVE A BRIGHT FUTURE