Lab 5: Arduino-Digital and Analog

Team Members:
Name: Chanda Akshay Kumar
Roll No.: 2024102014

Name: Samadhi Venkata Santhosh Kumar Yadav
Roll No.: 2024102054

Team ID: 42

Objective:

This lab aims to deepen the understanding of controlling LEDs and using sensors with Arduino through practical tasks involving digital and analog components.

Components Needed:

- **Arduino UNO** and **USB cable**
- **Breadboard**
- **3 LEDs** (red, green, blue)
- **220/330 ohm resistors** (3 for LEDs, 3 for RGB LED)
- **10k ohm resistor** (for pushbutton)
- **RGB LED**
- **Pushbutton**
- **HC-SR04 Ultrasonic sensor**
- **Buzzer**
- **MQ-2**
- **Red/White LED**
- **Connecting jumper wires**

Procedure:

Task 1: LED Sequential Control

- Place the 3 LEDs on the breadboard.
- Attach a 220/330 ohm resistor to the cathode (short leg) of each LED.
- Connect the free end of each resistor to the ground rail on the breadboard.
- Connect the anodes (long legs) of the LEDs to the digital pins on the Arduino.
- Open the Arduino IDE and write a code that turns on and off each LED in a sequence with a 1-second delay between each.

Upload the Code:

- Connect the Arduino UNO to the computer via USB.
- Select the correct board and port in the Arduino IDE.
- Upload the code to the Arduino.
- After uploading, observe the LEDs light up in sequence with a 1-second interval.

Task 2: RGB LED Control using Pushbutton

- Insert the RGB LED into the breadboard.
- Attach a 220/330 ohm resistor to each color leg (red, green, and blue) of the RGB LED, and connect the other end of each resistor to a digital pin on the Arduino.
- For a common cathode RGB LED, connect the common leg to the ground rail.
- Connect one terminal of the pushbutton to a digital pin on the Arduino and the other to the ground rail, with a 10k ohm pull-up resistor connected between the terminal and the positive rail.

Write the Code:

- Open the Arduino IDE and write a code to change the RGB LED's color with each button press.
- Follow the same upload procedure as Task 1.
- After uploading the code, press the pushbutton. Each press should cycle through red, green, and blue colors.

Task 3: Arduino Analog - Ultrasonic Sensor Setup and Testing

1. Connect the **Echo pin to pin 9** and **Trigger pin to pin 10** on the Arduino.
2. Connect the VCC and GND to the power and ground rails, respectively.
3. Upload the ultrasonic sensor code.
4. Observe distance readings for objects placed at 1, 2, and 3 inches in the Serial Monitor. Point the sensor at the ceiling to verify the distance.

Task 4: Adding an LED for Proximity Warning

- Replace the buzzer with an LED and connect it to the ultrasonic sensor setup.
- Adjust the code to light the LED when the object is within a specified range.

Task 5: MQ-2 Smoke Sensor

1. **Objective**:
   - Read the sensor's analog output voltage and trigger a red LED when the smoke concentration exceeds a certain level, with a green LED for lower concentrations.
2. **Circuit Setup**:
   - Connect the A0 pin of the MQ-2 sensor to an analog pin on the Arduino, GND to ground, and VCC to the 5V power supply.
   - Connect the red and green LEDs to digital pins, along with the necessary resistors (220 ohms).

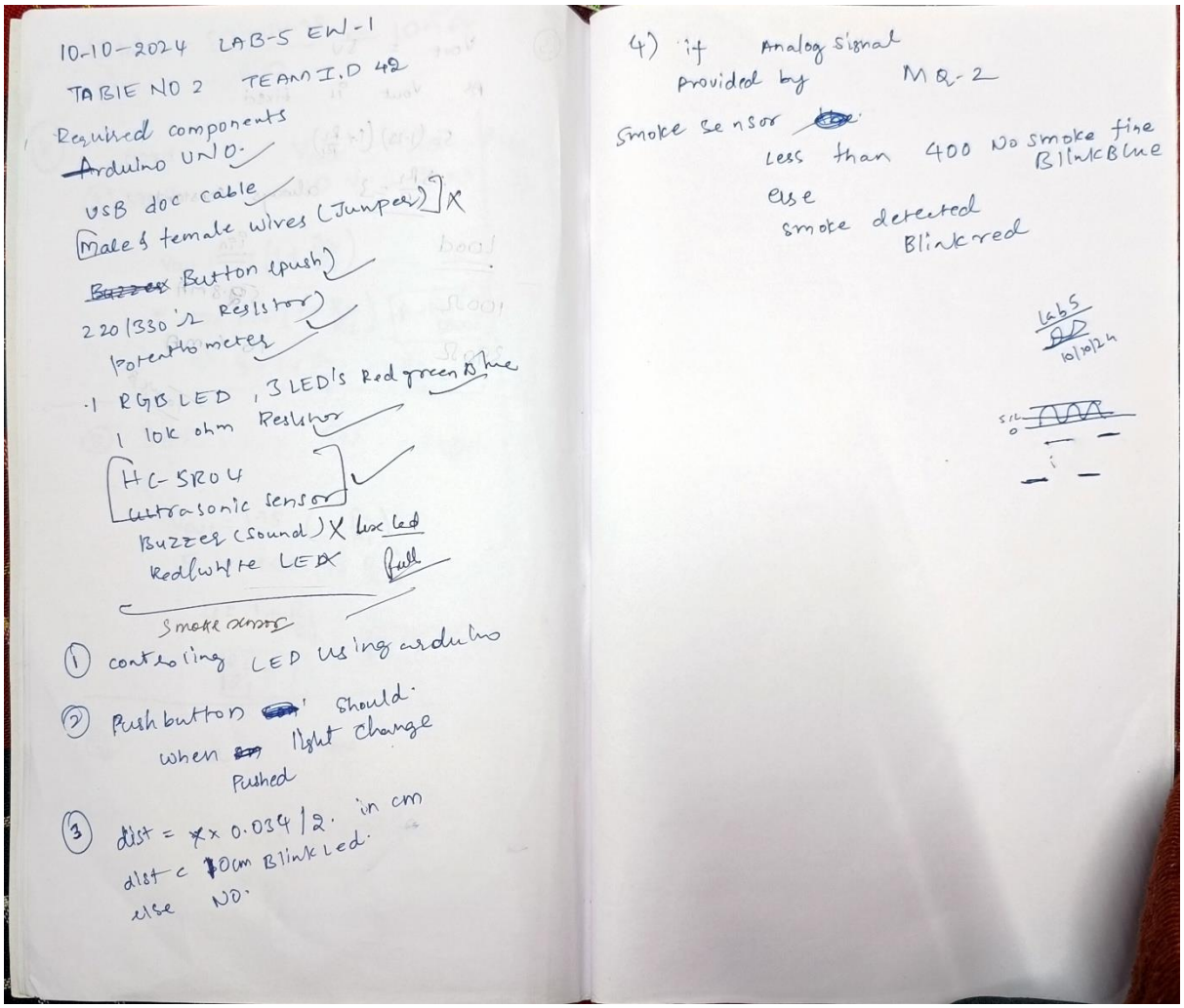
Observation:

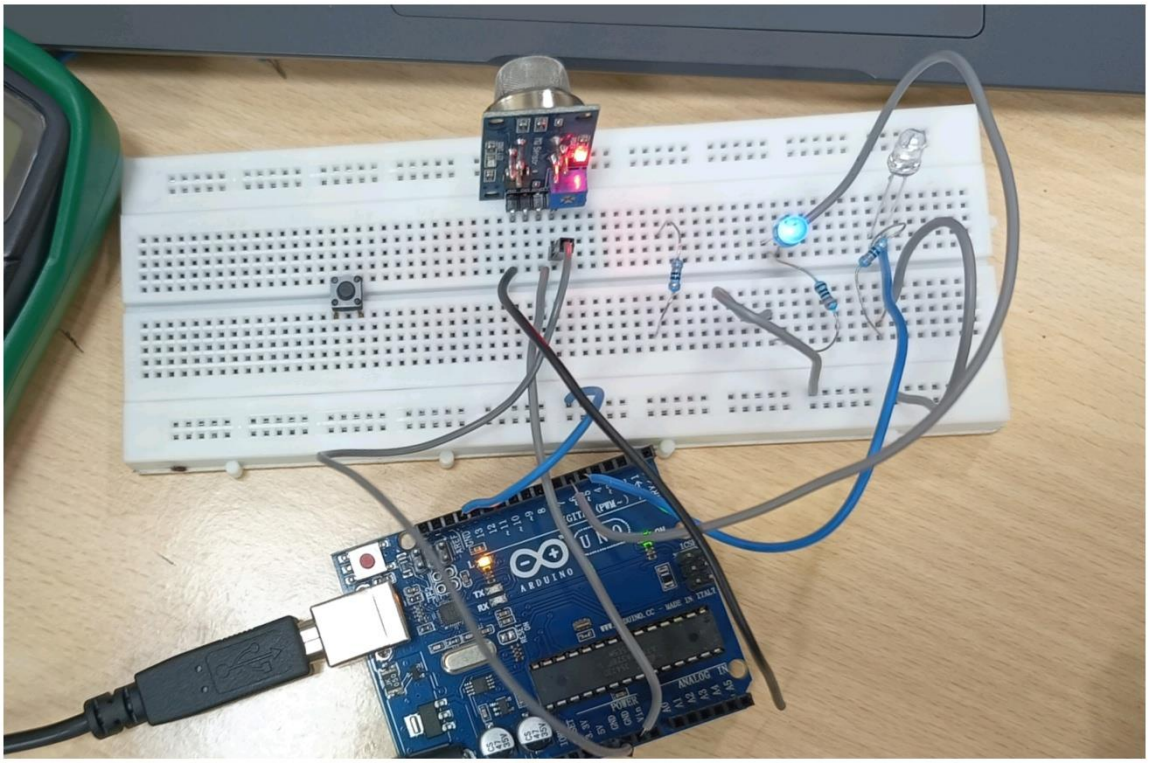
Component	Task	Output
LEDs	Task 1: Sequential Control	LEDs light up in sequence with 1-sec delay
RGB LED	Task 2: Pushbutton Control	RGB LED cycles through red, green, blue colours
Ultrasonic Sensor	Task 3: Distance Measurement	Accurate distance readings for objects at 1, 2, and 3 inches
MQ-2 Smoke Sensor	Task 5: Smoke Detection	Red LED on for high smoke concentration, Green LED for low

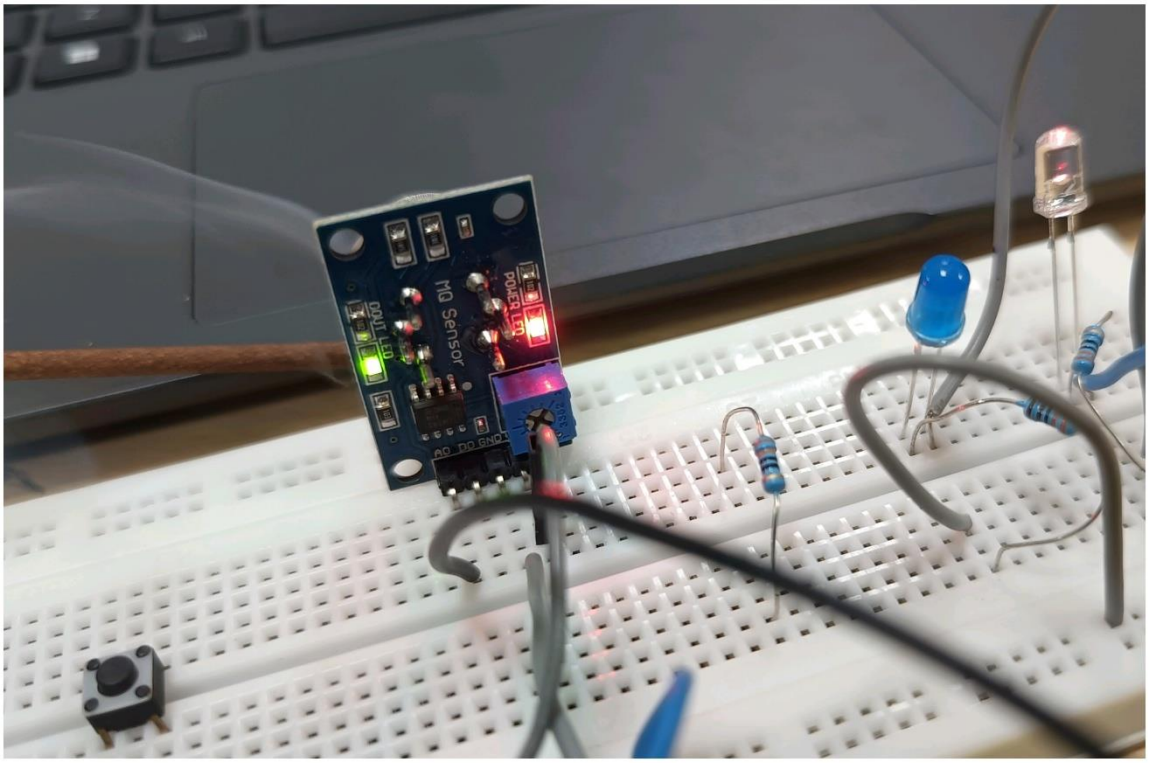
---

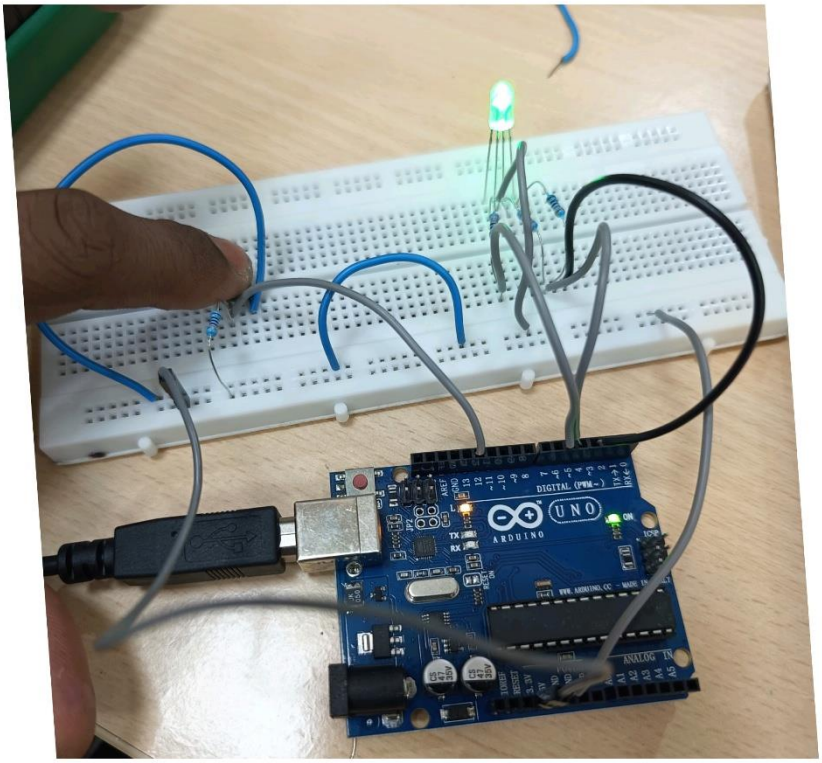
Conclusion:

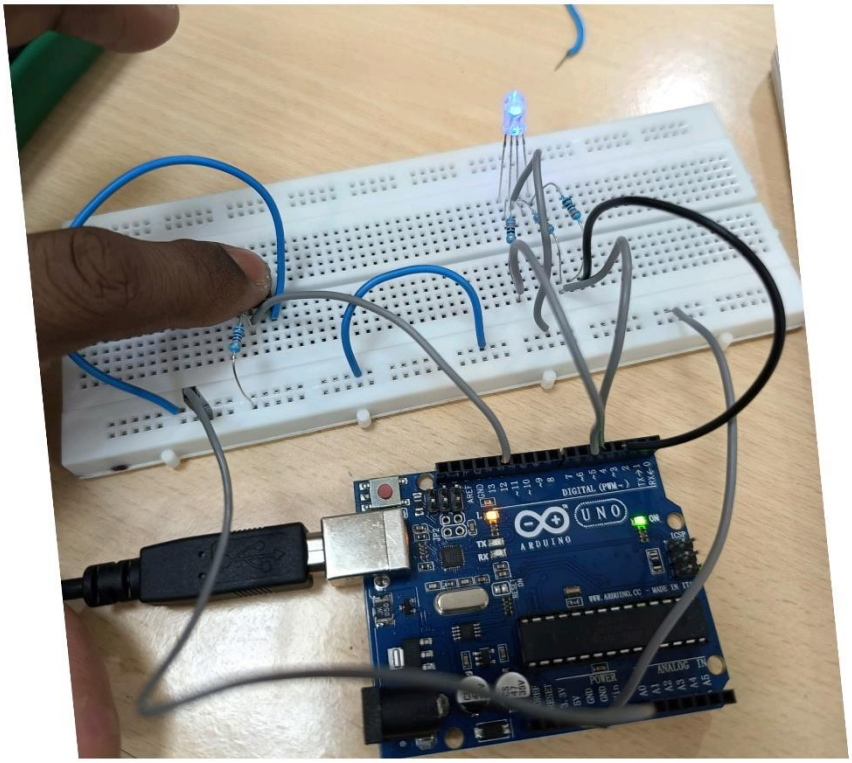
This lab demonstrated effective control of LEDs using Arduino through digital signals, the use of a pushbutton for RGB LED control, and the practical application of analog sensors like the ultrasonic and smoke sensors. These tasks helped build a foundation for further exploration of digital and analog input/output in embedded systems.

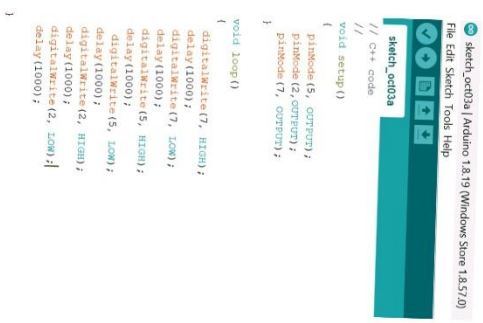



ARDUINO

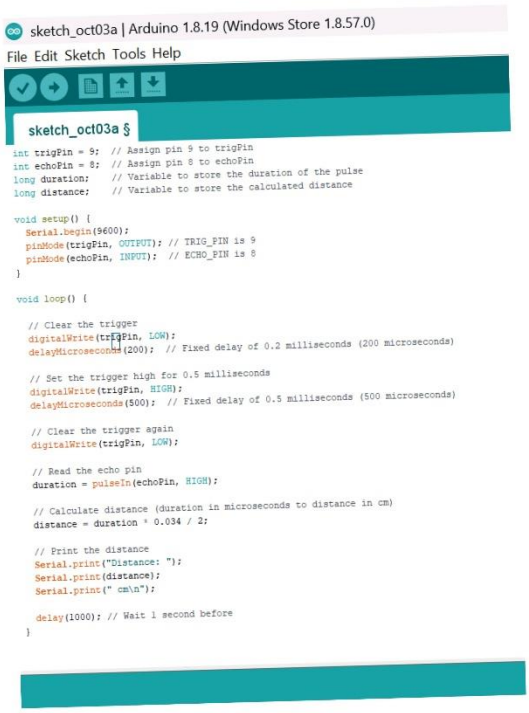

MQ Sensor
POWER LED
DOUT LED
AO DO GND

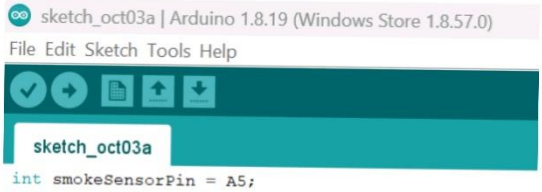

DIGITAL (PWM~)
UNO
ARDUINO
ANALOG IN


DIGITAL (PWM~)
UNO
ARDUINO
ANALOG IN

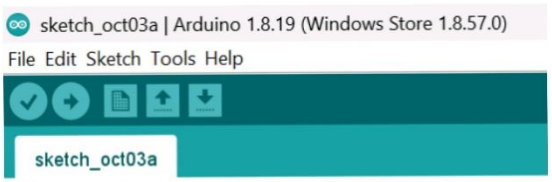



```
if(distance<10){
    digitalWrite(7,HIGH);
    //delay(1000);
  }
  else{
        digitalWrite(7,LOW);

  }
```






```
int smokeSensorPin = A5;
int sensorValue = 0;

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

void loop() {
  sensorValue = analogRead(smokeSensorPin);


  Serial.println(sensorValue);

  if(sensorValue>400){
    digitalWrite(5,HIGH);
    digitalWrite(6,LOW);
  }
  else{
    digitalWrite(6,HIGH);
    digitalWrite(5,LOW);
  }

  delay(1000);
}
```




```
// C++ code
//
void setup()
{
  pinMode(5, OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(3, OUTPUT);
}
int c=3;
void loop(){
  if(!(digitalRead(12))){
    digitalWrite(c,HIGH);
    delay(100);
    digitalWrite(c,LOW);
    c++;
    if(c>5){
      c=3;
    }
  }
}
```