LAB 5: Arduino-Digital and Analog

Arduino Digital

Objective

This lab aims to extend your understanding of controlling LEDs using Arduino.

Components Needed

- Arduino UNO and USB cable
- Breadboard
- 3 LEDs (red, green, blue)
- 220/330 ohm resistors (3)
- Connecting jumper wires

Procedure

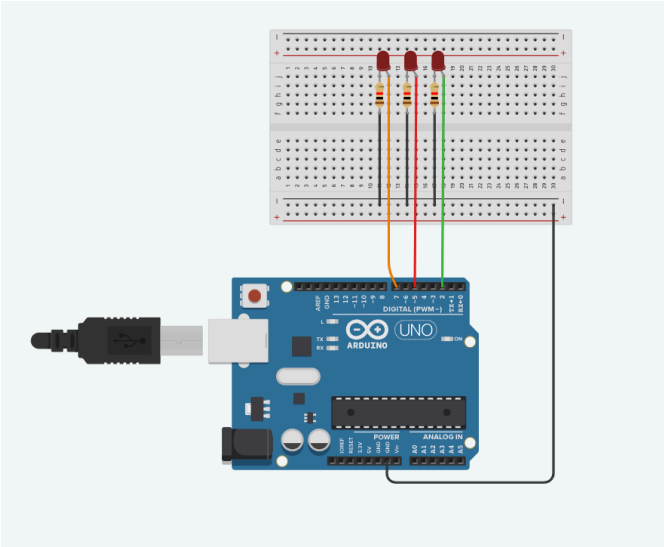
Task 1: LED Sequential Control

Connect the three LEDs to the breadboard

→ Attach a resistor (220/330 ohm) to the cathode (short leg) of each LED.

→ Connect the free end of each resistor to ground rail on the breadboard

→ Connect the cathodes (long legs) of the LEDs to the digital pin on the Arduino.



Arduino Code

→ Open the Arduino IDE on your computer.

→ Start a new sketch and write the code such that you get three different LEDs to turn on and turn off in a simple sequence (say with a gap of 1 sec).

Upload the Code

→ Connect your Arduino UNO to your computer using the USB cable.

→ Select the correct board and port in the Arduino IDE.

→ Click on the “Upload” button to transfer the code to the Arduino.

Observe the Output

→ After uploading, observe the LEDs as they light up in sequence with a 1-second interval.

Task 2: RGB LED control using pushbutton

Objective

Use a pushbutton to control an RGB LED, allowing each button press to change the LED's color in a predefined sequence.

Components Needed

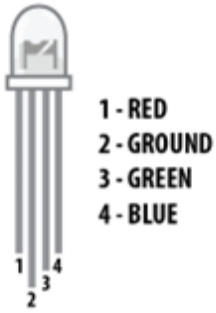
- 1 RGB LED
- 1 Pushbutton
- 3 Resistors (220/330 ohm for RGB LED)
- 1 Resistor (10k ohm for pushbutton)
- Connecting jumper wires
- Breadboard

Procedure

Circuit Setup

→ Place the RGB LED: Insert the RGB LED into the breadboard, noting the longer leg and the individual color legs (red, green, blue).

Pinout of RGB LED (Common Cathode):

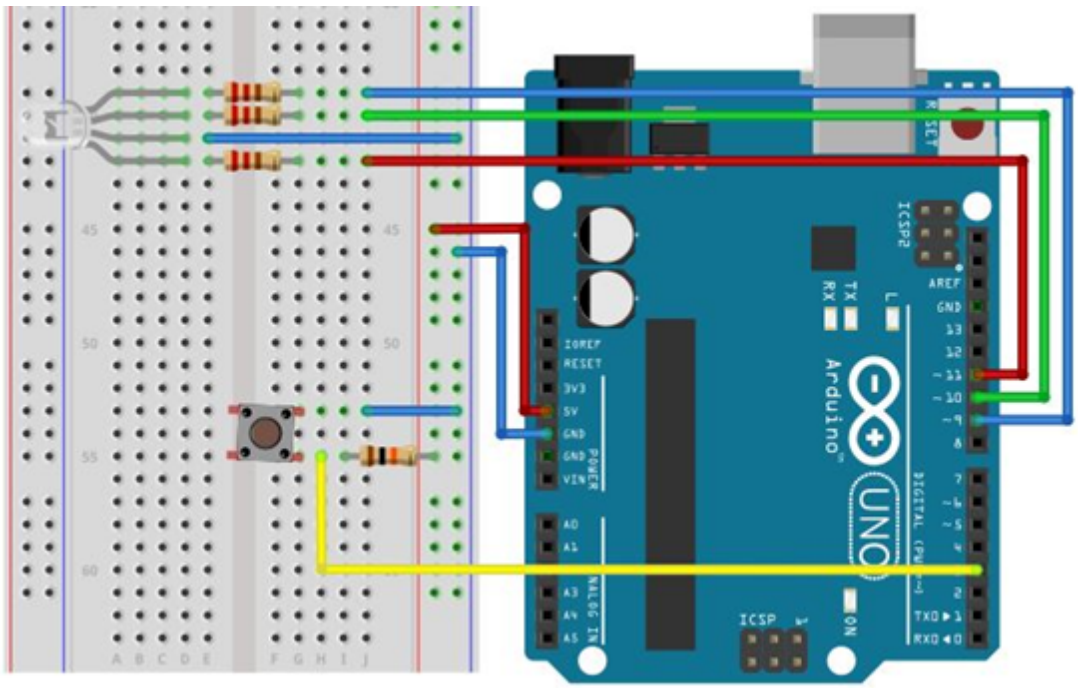


→ Connect Resistors: Attach a resistor (220/330 ohm) to each of the color legs (red, green, and blue) of the RGB LED. Connect the other end of each resistor to the digital pins on the Arduino

→ Connect the Common Leg: If it's a common cathode RGB LED, connect the common leg to the ground rail on the breadboard; for a common anode, connect it to the positive rail.

→ Set Up the Pushbutton:

- Connect one terminal of the pushbutton to a digital pin on the Arduino.
- Connect the other terminal to the ground rail.
- Use a 10k ohm resistor to connect the same terminal of the pushbutton to the positive rail (this acts as a pull-up resistor).



Write the Code

→ Open the Arduino IDE and create a new sketch. Write the code.

Upload the Code

→ Connect your Arduino UNO to your computer using the USB cable.

→ Select the correct board and port in the Arduino IDE.

→ Click on the “Upload” button to transfer the code to the Arduino.

Test the Setup

→ After uploading the code, press the pushbutton.

→ Each press should cycle the RGB LED through red, green, and blue.

Arduino Analog:

Objective:

In this experiment, we will learn to program simple applications using an Ultrasonic sensor attached to the Arduino board.

Components Required:

- Arduino UNO
- USB cable
- Breadboard
- HC-SR04
- Ultrasonic sensor
- Buzzer
- Red/White LED
- connecting jumper wires.

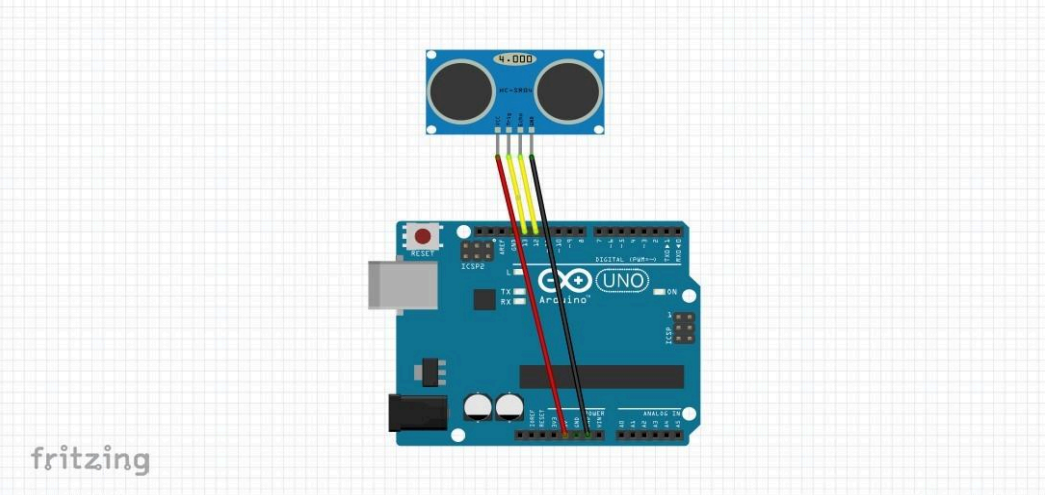
TASK 1:

Ultrasonic Sensor Setup and Testing

1. Connect the Circuit:
    - Echo pin of the HC-SR04 ultrasonic sensor to pin 12 on the Arduino UNO.

- Trigger pin of the sensor to pin 13 on the Arduino.
- Vcc pin of the sensor to the 5V power pin on the Arduino.
- GND pin of the sensor to the GND pin on the Arduino.

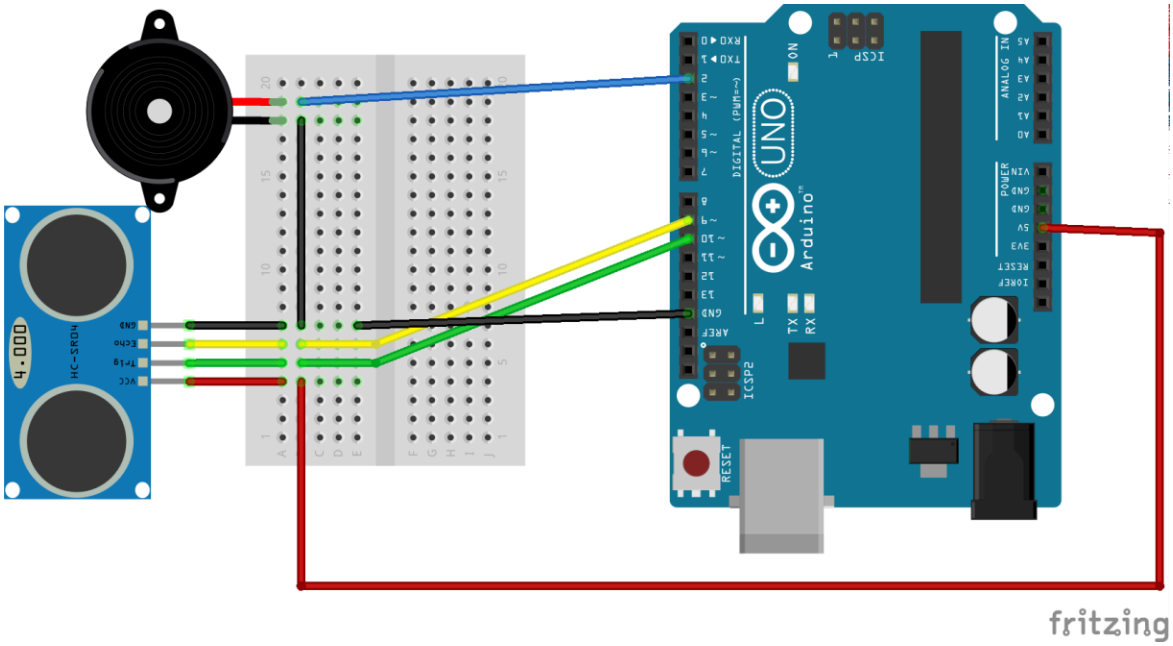
2. Upload the Code:
   - Upload the provided code for the HC-SR04 ultrasonic sensor to the Arduino using the Arduino IDE.
3. Test Obstacle Distance:
   - Place an object at 1 inch from the sensor and observe the reading in the Serial Monitor.
   - Move the object to 2 inches and note down the reading.
   - Place the object at 3 inches and verify the values again. Check if they match the actual distance.
4. Point the Sensor at the Ceiling:
   - Aim the sensor upwards towards the ceiling.
   - Observe and note the reading in the Serial Monitor.



TASK 2:

Adding an LED for Proximity Warning.

Modify your circuit as shown. (Please replace the Buzzer with an LED).



TASK 3:

Arduino Sensing- ***MQ-2 Smoke Sensor***

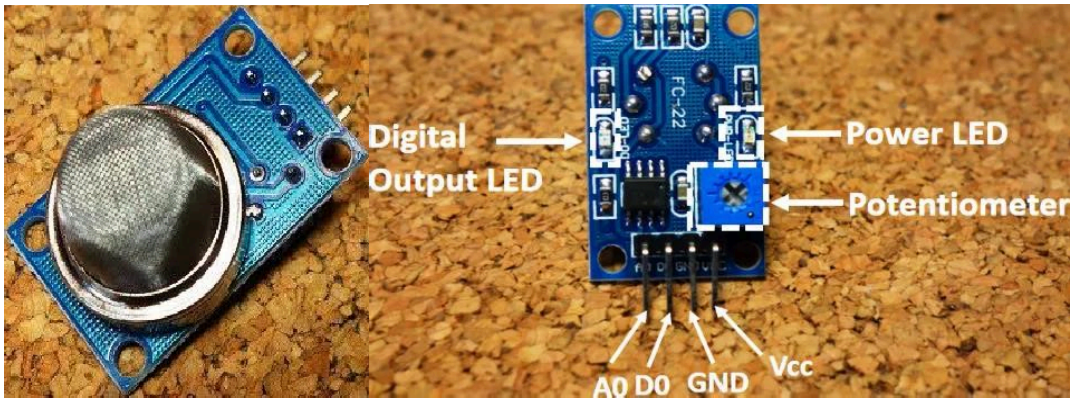
Objective:

In this section , you will have to read the sensor analog output voltage, and when the smoke reaches a certain level, it will make a buzzer and a red LED will turn on. A green LED will be on when the output voltage is below that level.

What is an MQ-2 Smoke Sensor?

The MQ-2 smoke sensor is sensitive to smoke and to the following flammable gases:
• LPG
• Butane
• Propane
• Methane
• Alcohol
• Hydrogen

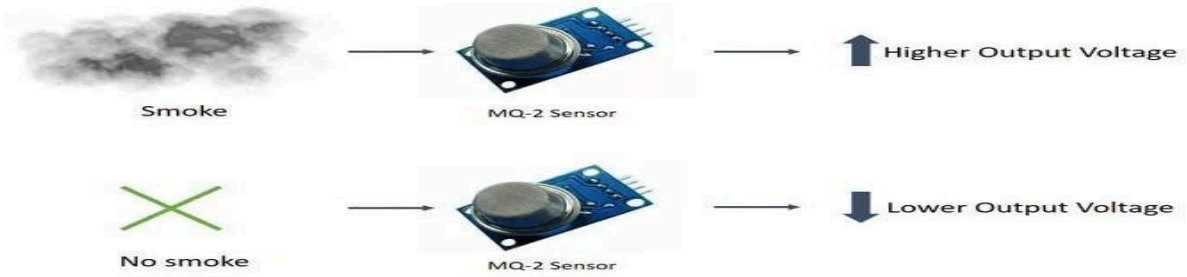
The resistance of the sensor is different depending on the type of the gas.
The smoke sensor has a built-in potentiometer that allows you to adjust the sensor sensitivity according to how accurate you want to detect gas.



How does it Work?

The voltage that the sensor outputs changes accordingly to the smoke/gas level that exists in the atmosphere. The sensor outputs a voltage that is proportional to the concentration of smoke/gas. In other words, the relationship between voltage and gas concentration is the following:

- The greater the gas concentration, the greater the output voltage
- The lower the gas concentration, the lower the output voltage



The output can be an analog signal (A0) that can be read with an analog input of the Arduino or a digital output (D0) that can be read with a digital input of the Arduino.

Componenets required

- Arduino uno.
- MQ-2 Smoke detection sensor.
- Male and Female jumper wires.
- 2 leds
- Buzzer
- Resistors.(220)

Pin Wiring

The MQ-2 sensor has 4 pins.

- Pin - Wiring to Arduino Uno
- A0 - Analog pins
- D0 - Digital pins
- GND - GND
- VCC - V

Here you don't need to perform the **Buzzer part.**