Objective:

The objective of this experiment is to get an idea how to do mini projects using Arduino Uno Board. From this experiment we will be able to simulate with Arduino Uno Board using Arduino IDE and Proteus. Besides, we will be able to learn how does Gas Sensor (MQ-2), ultrasonic Sensor work and show result in LCD display.

Apparatus/Instruments used in the Lab:

The components used in this experiment are:

Hardware:

1. Arduino Uno
2. Gas Sensor (MQ-2)
3. Buzzer
4. LCD Display
5. Motorized Exhaust Fan
6. Jumper Wires
7. Ultrasonic Sensor
8. Push button
9. LED
10. Resistor

Software:

1. Arduino IDE
2. Proteus

DO NOT COPY

Theory:

Now, lets learn and understand the hardware components and software that will be used in this experiment theoretically.

Hardware:

1. **Arduino Uno:**

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. [1]

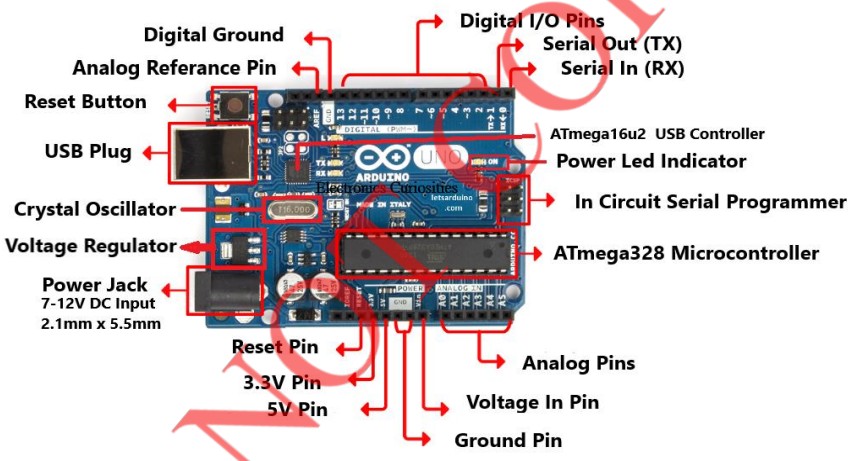

Fig 1: Pin Diagram of Arduino Uno

2. **Gas Sensor (MQ-2):**

MQ-2 is a sensor used to detect gases such as LPG (Liquefied Petroleum Gas), CO (Carbon Monoxide), and smoke. However, this sensor is often used as a gas leak detector both at home and in the industry. The MQ-2 sensor is made of an aluminum tube surrounded by silicon and there are electrodes made of gold. When detecting gas, a heating process occurs so that the ceramic SnO2 becomes a conductor so that it releases electrons to produce an analog voltage output. The MQ-2 sensor module has 4 pins, namely: VCC, GND, DO, and A0**.** [2]

Fig 2: Pin Diagram of Gas Sensor (MQ-2)

DO NOT COPY

3. Ultrasonic Sensor:

The ultrasonic sensor is an electronic device that calculates distance by emitting sound waves and collecting their echoes. The sensor is composed of two ultrasonic transducers. One is transmitter which outputs ultrasonic sound pulses and the other is receiver which listens for reflected waves. It measures the amount of time it takes for the sound to bounce off an object and then calculate the distance. [3]

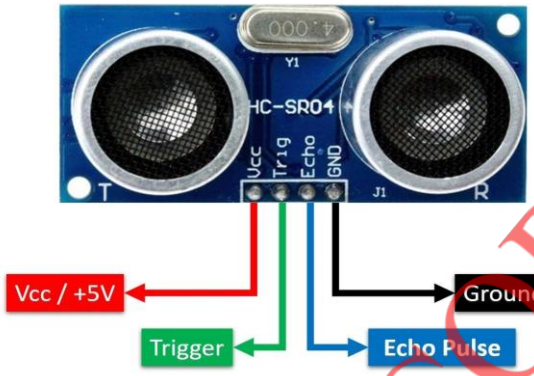

Fig 3: Pin diagram of Ultrasonic Sensor

DO NOT COPY

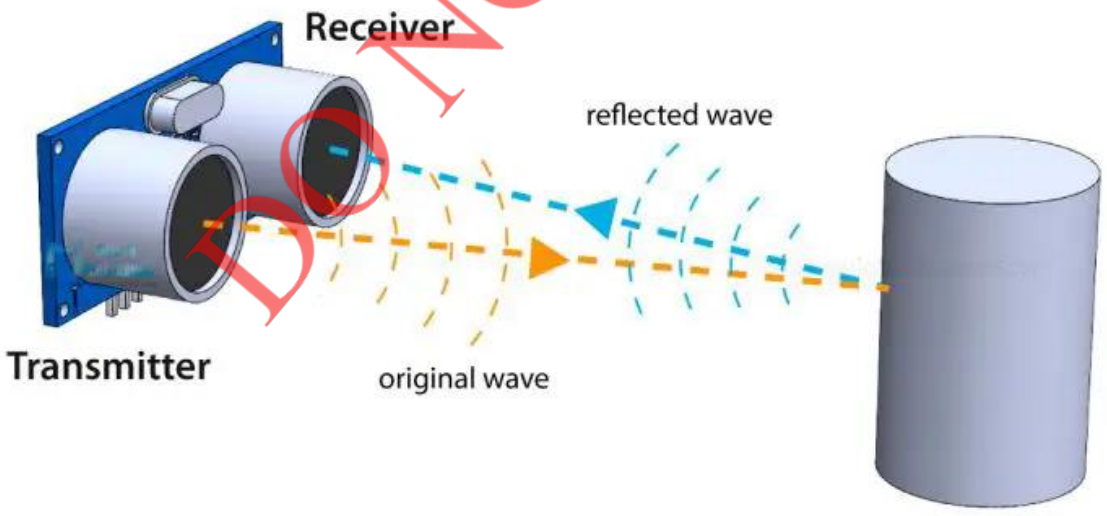

Fig 4: How the Ultrasonic Sensor works

4. Buzzer:

Buzzer is a electronic component that generates sound through the transmission of electrical signals. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '-' symbol or short terminal and it is connected to the GND terminal. [4]

Fig 5: Pin diagram of Buzze

5. LCD Display:

An LCD character display is a unique type of display that can only output individual ASCII characters with fixed size. Using these individual characters then we can form a text. The number of the rectangular areas define the size of the LCD. The most popular LCD is the 16×2 LCD, which has two rows with 16 rectangular areas or characters. It has 16 pins and the first one from left to right is the Ground pin. The second pin is the VCC which we connect the 5 volts pin on the Arduino Board. The V0 pin control the contrast of display. The RS pin send commands to the LCD. The 8 pins (D0 – D8) are control 8 bits data and convert it to ASCII. [5]

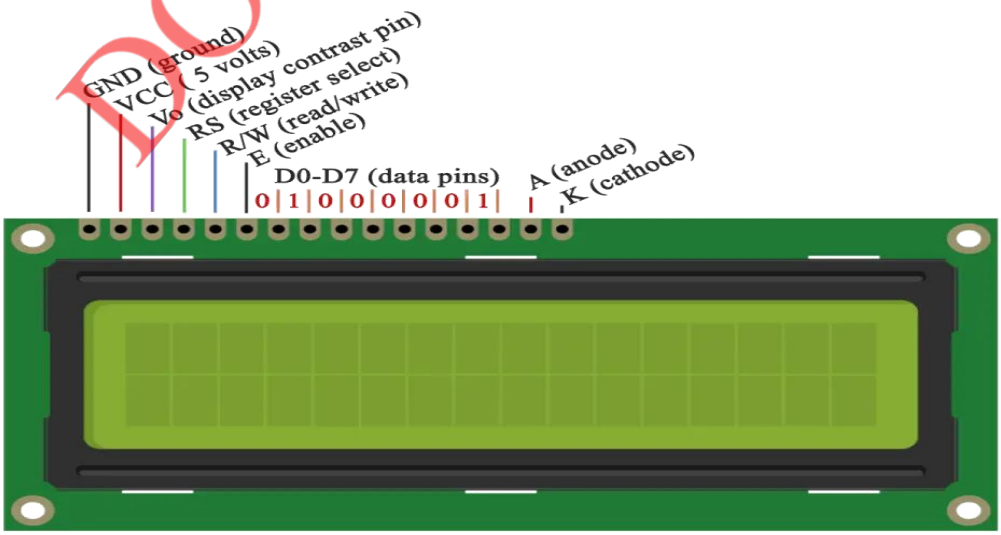

Fig 6: Pin diagram of LCD Display

DO NOT COPY

Solution of the mentioned problems:

1. Use a push button to toggle an LED using Proteus.

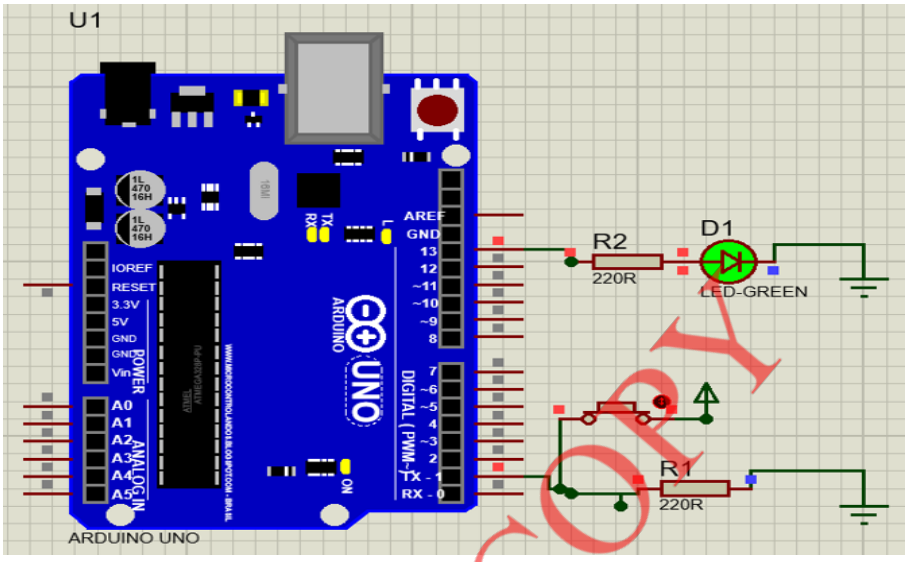

Fig 7: Toggle LED using push button

```
void setup() {
  Serial.begin(9600);
  pinMode(1, INPUT_PULLUP);
  pinMode(13, OUTPUT);
}
void loop() {
  int sensorVal = digitalRead(1);
  Serial.println(sensorVal);
  if (sensorVal == HIGH) {
    digitalWrite(13, HIGH);
  } else {
    digitalWrite(13, LOW);
   }
}
```

DO NOT COPY

2. LDR sensors can sense the change of intensity of light. Suppose, you have an LDR sensor and an LED. Interface an LDR sensor with Arduino in Proteus and turn on the LED when the intensity of light is low and turn it off when the intensity is high.

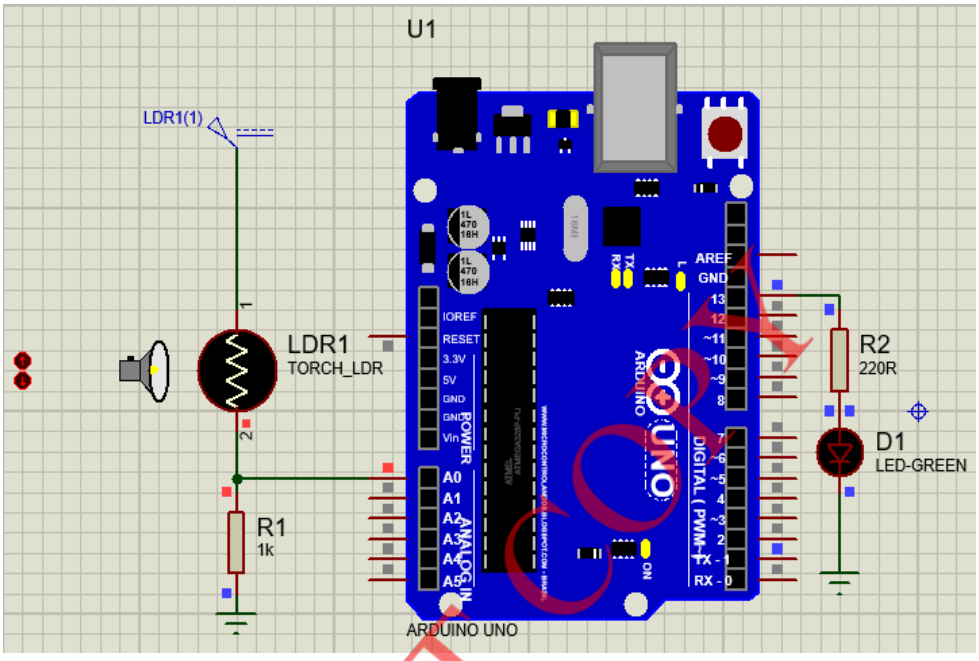

Fig 8: The intensity of light (Touch) is high and LED is OFF.

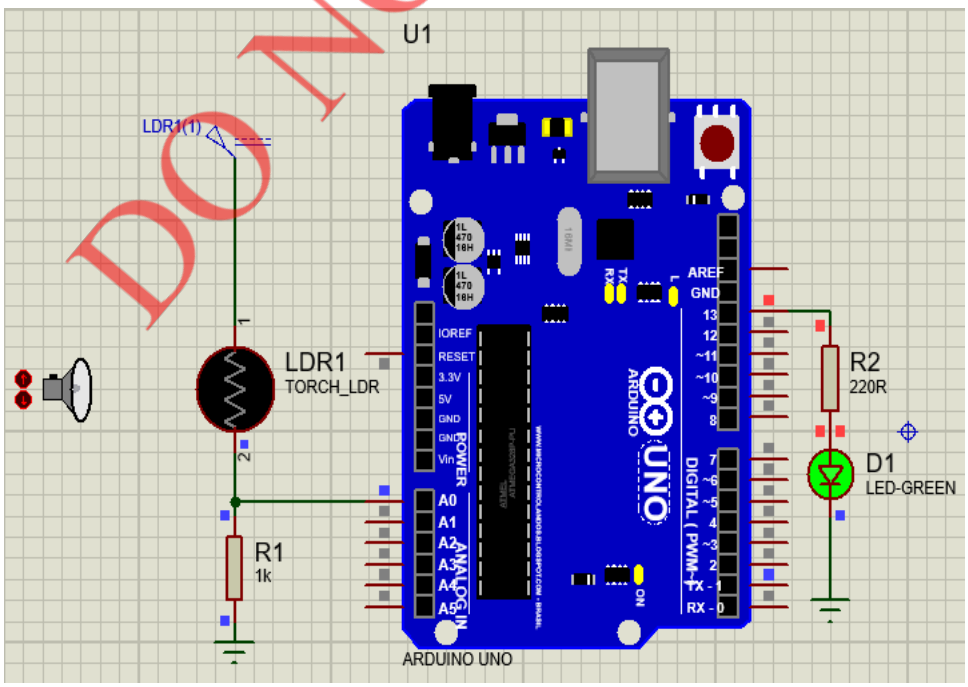

Fig 9: The intensity of light (Touch) is low and LED is ON.

DO NOT COPY

```
const int LEDPin = 13;// Initialize Pin11 for connecting LED
const int LDRPin = A0;// Initialize PIN A0 LDR PIN
void setup() {
  Serial.begin(9600);
  pinMode(LEDPin, OUTPUT);// Define LED pin as output
  pinMode(LDRPin, INPUT);// Define LDR pin as input
}
void loop() {
  int ldrStatus = analogRead(LDRPin); // read LDR lignt intensity as analog
value
// Control LED based on LDR light intensity
  if (ldrStatus <= 100) {
      digitalWrite (LEDPin, HIGH);
     Serial.print(" LIGHT ON : ");
      Serial.println(ldrStatus);
  } else {
     digitalWrite(LEDPin, LOW);
     Serial.print("LIGHT OFF: ");
     Serial.println(ldrStatus); // Print LDR analog value on serial port
  }
}
```

DO NOT COPY

3. Imagine you are doing a project on obstacle avoidance. You want to measure the distance of an object from your device. Interface a SONAR sensor with Arduino and measure distance using it. Show the distance in meter.

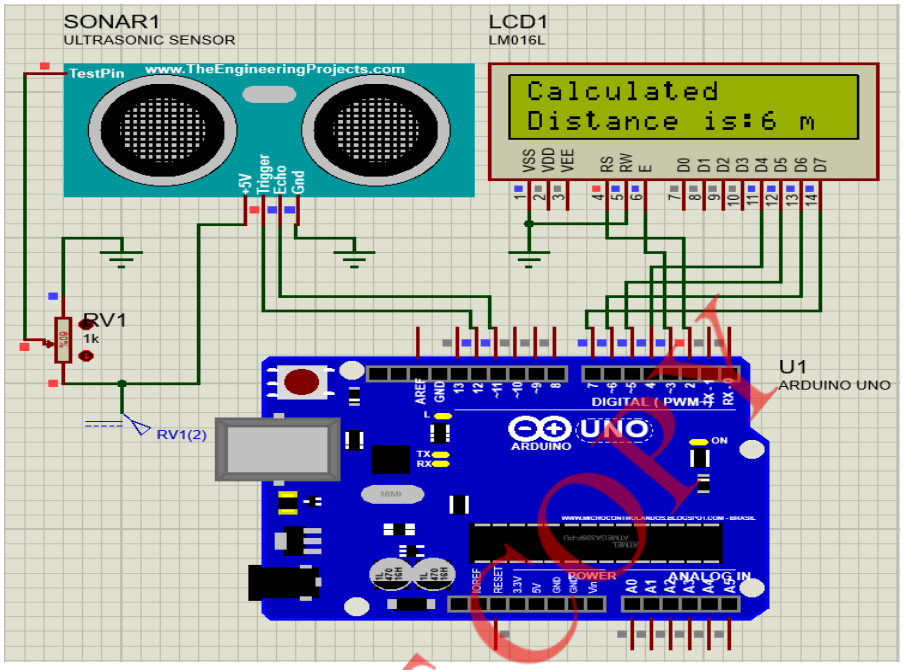

Fig 10: Calculate distance using Ultrasonic Sensor.

```
#include <LiquidCrystal.h>  //lcd libary
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);   //LCD object Parameters: (rs, enable, d4, d5, d6, d7)
const int trigPin = 12; //trig pin connection
const int echoPin = 11;  //echopin connection
long duration;
int distanceCm;
int distanceM;
float liquid;


void setup() {
lcd.begin(16,2);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
lcd.setCursor(0,0);
lcd.print("  Distance    ");
```

DO NOT COPY

```
lcd.setCursor(0,1);
lcd.print("  Measurement  ");
delay(2000);
lcd.clear();
}

void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distanceCm = duration*0.034/2;
distanceM = distanceCm/100;
lcd.setCursor(0,0);
lcd.print("Calculated ");
delay(10);
lcd.setCursor(0,1);
lcd.print("Distance is:");
lcd.print(distanceM);
lcd.print(" m ");
delay(10);
}
```

DO NOT COPY

4. A gas sensing based system should be implemented such that if gas is detected, an alarm/buzzer is triggered instantly to notify the incident.

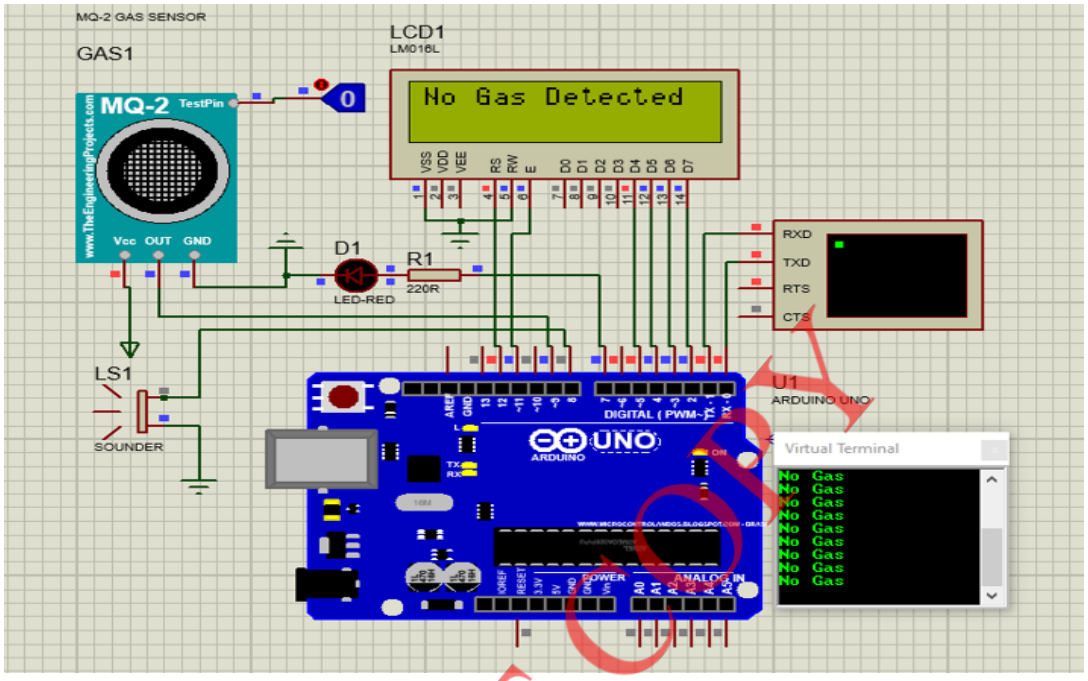

Fig 11: Gas detection using MQ-2 Sensor and play alarm.

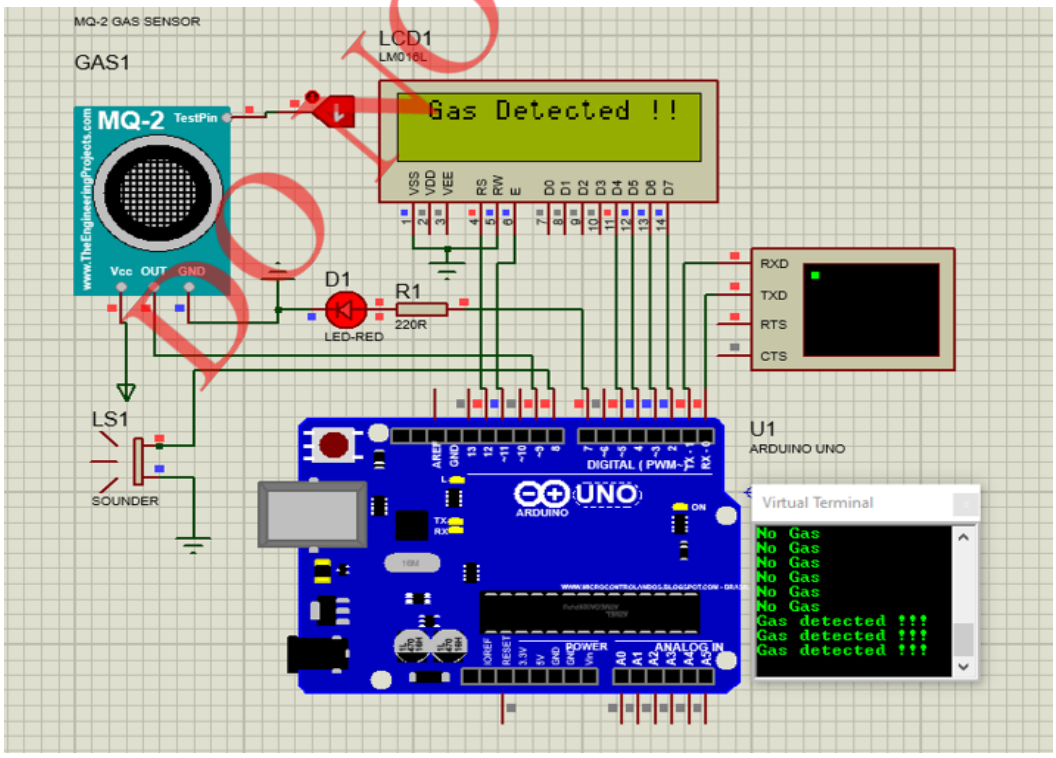

Fig 12: Gas detection using MQ-2 Sensor and play alarm.

DO NOT COPY

In **Fig-11**, the logic switch is 0, which means no gas. So, the LCD display shows “No Gas Detected” and the LED is OFF. Also, the buzzer is off.

In **Fig-12**, the logic switch is 1, which means gas Leakage. So, the MQ-2 sensor detects the gas and the LCD display shows “Gas Detected !!!”. Also, the LED is ON and the buzzer plays a sound.

```
#include<LiquidCrystal.h> //LCD library
LiquidCrystal lcd(12 , 11 , 5 ,4 ,3 ,2);  //lcd pins declarations
int Gas = 9;
int redLed = 7; //presense of gas
int greenLed = 6; //normaml mode absense of gas
int alarm = 8; // ring incase of gas
void setup() {
Serial.begin(9600);
pinMode(Gas , INPUT);
}
void loop() {
  if(digitalRead(Gas) == HIGH){
    lcd.setCursor(0,0);
    lcd.print(" Gas Detected !!!");
    Serial.println("Gas detected !!!");
    digitalWrite(7 , HIGH);
    digitalWrite(6, LOW);
    digitalWrite(8,HIGH);
}
  else{
  lcd.setCursor(0,0);
  lcd.print("No Gas Detected !");
  Serial.println("No Gas  ");
  digitalWrite(6, HIGH);
  digitalWrite(7 ,LOW);
  digitalWrite(8,LOW);
}
delay(500);
lcd.clear();
}
```

DO NOT COPY

5. A gas sensing based system should be implemented such that if gas is detected, a motorized exhaust fan is triggered to keep out the fumes.

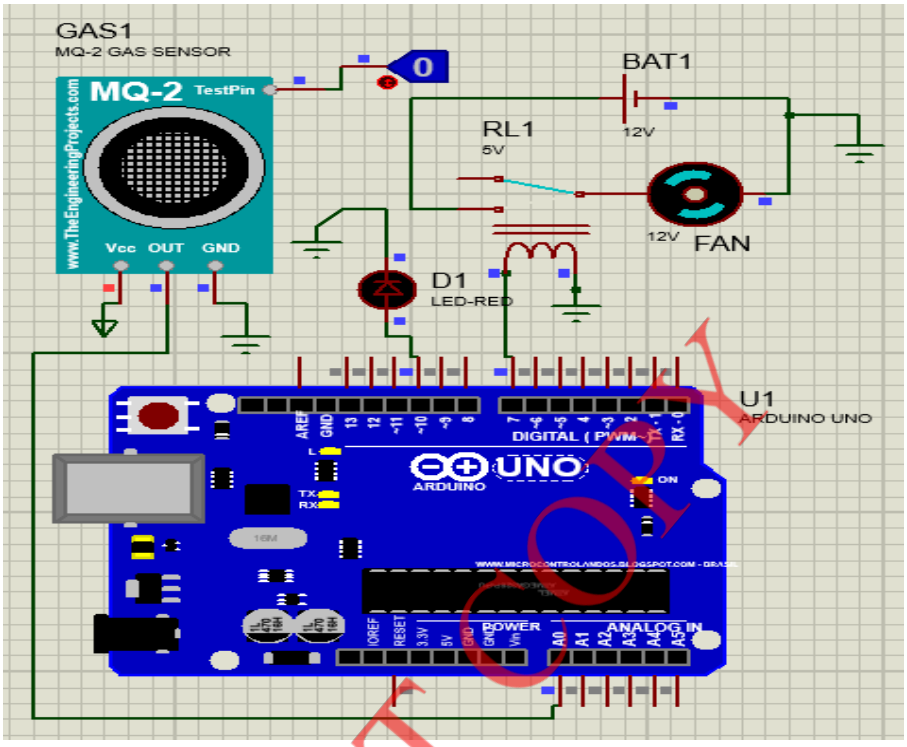

Fig 13: Gas detection using MQ-2 Sensor and trigger exhaust fan.

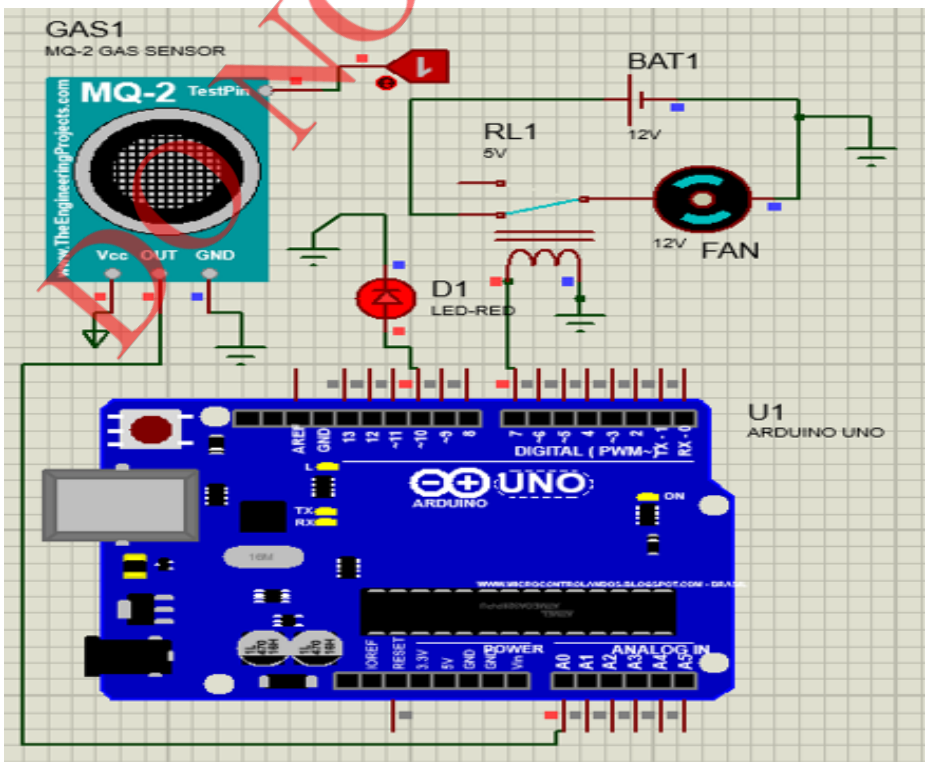

Fig 14: Gas detection using MQ-2 Sensor and trigger exhaust fan.

DO NOT COPY

In **Fig-13**, the logic switch is 0, which means no gas. So, the exhaust fan and LED are both off.

In **Fig-14**, the logic switch is 1, which means gas leakage. So, the MQ-2 sensor detects the gas and the exhaust fan and LED are ON.

```
const int Sesnsor_Pin = A0; // Sensor Pin
const int Fan_Pin = 7; // Fan Pin
const int RedLED = 10; // Red LED Pin

void setup()
{
 pinMode(Sesnsor_Pin,INPUT);  // Sensor Pin as a INPUT
 pinMode(Fan_Pin,OUTPUT);  // Fan Pin as a OUTPUT
 pinMode(RedLED,OUTPUT);  // Red LED Pin as a OUTPUT
}
void loop()
{
 if(digitalRead(Sesnsor_Pin) == HIGH)  // If Sesnsor Detect the Smoke
 {
  digitalWrite(Fan_Pin,HIGH);  // Turn ON the Fan
  digitalWrite(RedLED,HIGH);  // Turn ON the Red LED
 }
 else
 {
  digitalWrite(Fan_Pin,LOW);  // Turn OFF the Fan
  digitalWrite(RedLED,LOW);  // Turn OF the Red LED
 }
}
```

DO NOT COPY

Discussion:

Throughout the project, I developed some useful systems that simplify our life. Each project presented unique challenges and learning opportunities.

1: Push Button to Toggle an LED

In this project, I learned how to use a push button to control an LED. The primary challenge was ensuring the connections of all components. This project demonstrated the fundamental concept of digital input handling.

2: LDR Sensor to Control LED Based on Light Intensity

The LDR sensor project taught me how to use analog inputs to monitor environmental conditions. The main difficulty was calibrating the LDR to accurately respond to different light levels. This project has real-life applications in automatic lighting systems, where lights can be controlled based on ambient light levels to save energy.

3: SONAR Sensor for Distance Measurement

Integrating a SONAR sensor with Arduino was a step toward understanding distance measurement techniques. This project has practical applications in obstacle avoidance systems for robots and automated vehicles.

4: Gas Sensing System with Alarm

Implementing a gas sensing system with an alarm involved understanding the gas sensor's characteristics and interfacing it with Arduino. This system is vital for safety applications, such as detecting gas leaks in homes and industries, where immediate notification can prevent critical situations.

5: Gas Sensing System with Motorized Exhaust Fan

Building on the previous project, this system added an actuator (motorized exhaust fan) to mitigate detected gas presence. The challenge was synchronizing the sensor input with the motor control to ensure the timely activation of the exhaust fan. This project can be implemented to maintain safety in residential, commercial, and industrial settings.

In conclusion, all these projects provided a comprehensive understanding of sensor and actuator integration with Arduino, highlighting the importance of reliable sensor readings and timely actuator responses in developing effective and safe embedded systems.

DO NOT COPY

References:

[1] https://docs.arduino.cc/hardware/uno-rev3/#compatibility

[2] https://saptaji.com/2022/02/14/how-to-get-mq-2-gas-sensor-data-using-nodemcu/

[3] https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

[4] https://www.elprocus.com/buzzer-working-applications/

[5] https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/