INTRODUCTION

IIn this project, we've created a complete GSM and Arduino Gas Leakage Detector system with SMS warning capabilities and a window-opening mechanism for when gas leaks occur. An Arduino microcontroller, a MQ135 gas sensor, a Sim900 GSM module, and a window opening control mechanism are all interfaced by the system.

In order to ensure safety and avert tragedies, smoke and gas leak detectors are essential for spotting possible threats like fire or gas leaks in buildings. Events like unintentional fires and burst cylinders have historically seriously damaged economies, underscoring the significance of these safety precautions.

This circuit is intended to detect the presence of smoke or gas leaks and activate an alert system. The MQ135 Smoke/Gas sensor is the main device it uses, and

WORKING OF THE PROJECT

Gas Leakage Detector using GSM & Arduino with SMS Alert

- As a complete safety solution, the gas leak detecting system and window opening mechanism work together. The system continuously checks the surroundings for the presence of dangerous gases using a MQ135 Smoke/Gas sensor. An Arduino microcontroller receives the sensor's output and uses it to process data in real time. Instant feedback on gas levels is given to passengers via an LCD display, which indicates whether or not they have surpassed a certain threshold or are within safe bounds. The Sim900 GSM module is triggered by the Arduino to send an SMS alert to a designated mobile number alerting the recipient to the potential hazard of a gas leak. To initiate the windows opening mechanism, the Arduino and an ESP32 microcontroller interact simultaneously via Bluetooth. After receiving the signal, the ESP32 operates servo motors

SOURCE CODE (ARDUINO)

```
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
SoftwareSerial BTSerial(2, 3);
SoftwareSerial mySerial(9, 10);
int gasValue = A0;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
void setup() {
Serial.begin(9600);
 BTSerial.begin(9600);
 mySerial.begin(9600);
 lcd.begin(16, 2); }

 void loop() {
int gasLevel = analogRead(gasValue);
Serial.print("Gas Level: ");
 Serial.println(gasLevel);
 lcd.clear();
lcd.setCursor(0, 0);
 lcd.print("Gas Level: ");
 lcd.print(gasLevel);
 LCDif (gasLevel > 500) { SendMessage();

  BTSerial.println("OPEN_WINDOWS");
delay(1000);  }
delay(1000);  }
void SendMessage() {
mySerial.println("AT+CMGF=1");
delay(1000);
mySerial.println("AT+CMGS=\"+91900xxxxxxx\""); // Replace with your
phone number
delay(1000); mySerial.println("Gas leak detected. Open windows!");
delay(100);
mySerial.println((char)26);
delay(1000); }
```

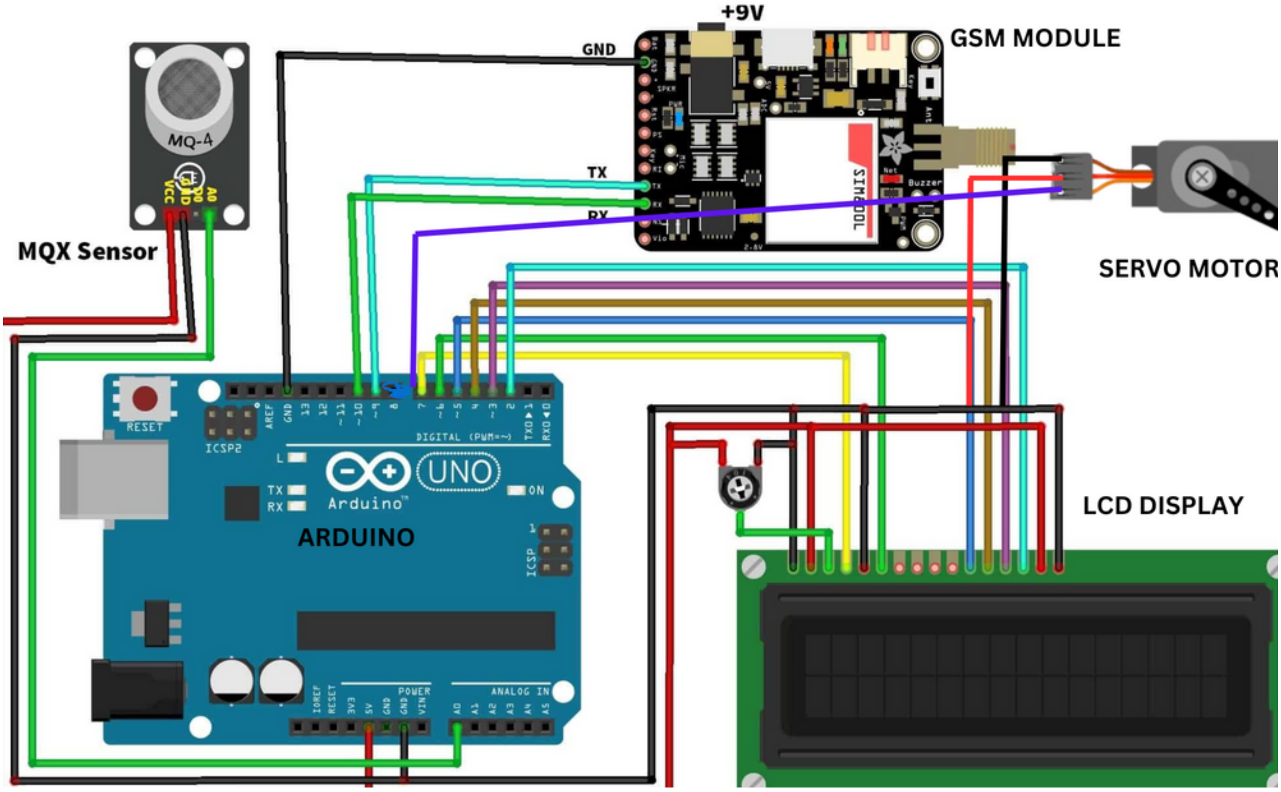
ESP32 CODE

```
#include <Servo.h>
#include "BluetoothSerial.h"
 BluetoothSerial SerialBT;
Servo windowServo;
void setup() { Serial.begin(9600);
SerialBT.begin("ESP32_BT");
 windowServo.attach(8); // Pin for servo control
if (!pairWithArduino()) {
Serial.println("Failed to pair with Arduino!");
 } }
 void loop() { if (SerialBT.available()) {
String command = SerialBT.readStringUntil('\n');
 if (command.equals("OPEN_WINDOWS")) {
 windowServo.write(90);
delay(2000);
} } }
bool pairWithArduino() {
Serial.println("Scanning for nearby devices...");
int devicesFound = SerialBT.scanForUuid("00001101-0000-1000-8000-
00805F9B34FB");
if (devicesFound > 0) { Serial.println("Arduino Bluetooth module found.
Initiating pairing...");
bool paired = SerialBT.pair();
if (paired) { Serial.println("Pairing successful!");
 return true; }
else { Serial.println("Pairing failed."); } }
else { Serial.println("No Arduino Bluetooth module found."); }
return false; }
```

COMPONENTS USED

1) Arduino board (e.g., Arduino Uno)
2) ESP32 development board
3) MQ135 Gas Sensor
4) Servo Motor
5) Bluetooth Module (e.g., HC-05)
6) GSM 900 Module
7) LCD Display
8) Jumper wires
9) Power source (USB cable, battery, etc.)
10) Breadboard (optional, for prototyping)

CIRCUIT DIAGRAM



CONNECTIONS FOR THE CIRCUIT

STEP 1: Gas Sensor (MQ135) Connection
1) Identify the analog pin on your Arduino board (labeled A0, A1, etc.).
2) Connect the analog output of the MQ135 gas sensor to any available analog pin on the Arduino.

STEP 2: Bluetooth Module Connection
1) Identify pin 2 and pin 3 on your Arduino board for serial communication (hardware serial pins).
2) Connect the RX pin of the Bluetooth module to pin 2 (TX) on the Arduino.
3) Connect the TX pin of the Bluetooth module to pin 3 (RX) on the Arduino.
4) Ensure that the Bluetooth module's power and ground pins are connected to the corresponding power and ground pins on the Arduino.

STEP 3: GSM 900 Module Connection
1) Identify the hardware serial pins on your Arduino board for serial communication.
2) Connect the TX pin of the GSM 900 module to the RX pin of the Arduino.
3) Connect the RX pin of the GSM 900 module to the TX pin of the Arduino.
4) Connect the GSM module's power and ground pins to the corresponding power and ground pins on the Arduino.

STEP 4: LCD Display Connection
1) Connect the RS (Register Select) pin of the LCD display to digital pin 7 on the Arduinc
2) Connect the EN (Enable) pin of the LCD display to digital pin 6 on the Arduino.
3) Connect the D4, D5, D6, and D7 pins of the LCD display to digital pins 5, 4, 3, and 2 respectively on the Arduino.
4) Connect the LCD display's power and ground pins to the corresponding power and ground pins on the Arduino.

STEP 5: Power Up
Power up your Arduino board using a suitable power source (USB cable, battery, etc.).

STEP 6: Servo Motor Connection to ESP32
1) Identify the appropriate pins on your ESP32 board for connecting the servo motor.
2) Connect the signal wire of the servo motor to the chosen pin on the ESP32.
3) Connect the servo motor's power and ground wires to the corresponding power and ground pins on the ESP32.

CONCLUSION

the gas leakage detection system with an integrated windows opening mechanism serves as a vital safety measure in various environments prone to gas hazards. By combining the MQ135 Smoke/Gas sensor, Arduino microcontroller, GSM module (Sim900), ESP32 microcontroller, and LCD display, the system provides comprehensive monitoring, alerting, and mitigation functionalities.

Throughout the project, we successfully developed a robust system capable of detecting gas leakages in real-time. The MQ135 sensor's excellent sensitivity and quick response time allow for timely detection of hazardous gases, ensuring the safety of occupants and property. The Arduino microcontroller processes sensor data, displaying gas levels on the LCD display and triggering alerts when gas levels exceed predefined thresholds.

The integration of the GSM module enables the system to send SMS alerts to predefined mobile numbers, notifying users of gas leakages even when they are not on-site. Additionally, the communication between the Arduino and ESP32 via Bluetooth facilitates the automatic opening of windows upon detecting excessive gas levels. This ventilation mechanism helps dissipate the gas, mitigating potential risks and enhancing overall safety.