Cindy Zulani (001202000137)
Mulya Fajar (001202000101)
Tito Alexsta (001202000098)
M-IT 2020 Class 3

Report of Final Project Internet of Things

Project Name: Gas Leakage Monitoring System

Project Description

In this final project, we make a gas leakage monitoring system using the NodeMCU board and the Blynk app. This project mainly uses the MQ2 gas sensor and it's based on IoT technology. That is, we can monitor everything in this security system on the mobile phone using the Internet.

When power ON this security system, The NodeMCU board connects to the interface created in the Blynk app via the Blynk cloud. Then the system should be activated by the button created in the Blynk app interface. In this case, when a gas leak occurs, it is detected by the MQ2 sensor. Then the buzzer and red LED are activated. At the same time, it notifies our smartphone via a push notification. Also, we can see the gas values on the LCD and Blynk app interface. In the absence of a gas leak, the green LED bulb turns ON and operates normally this security system.

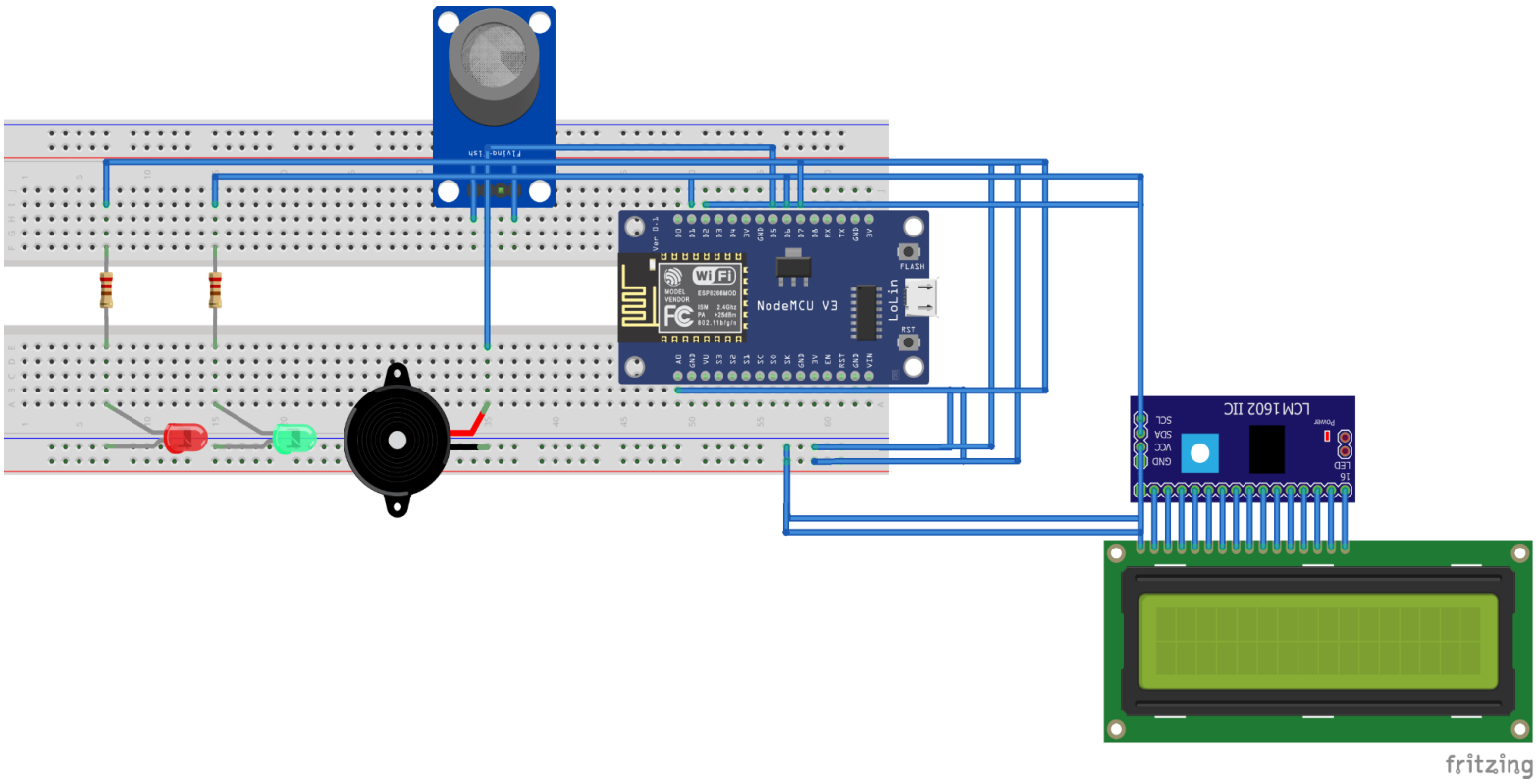
Project Requirements

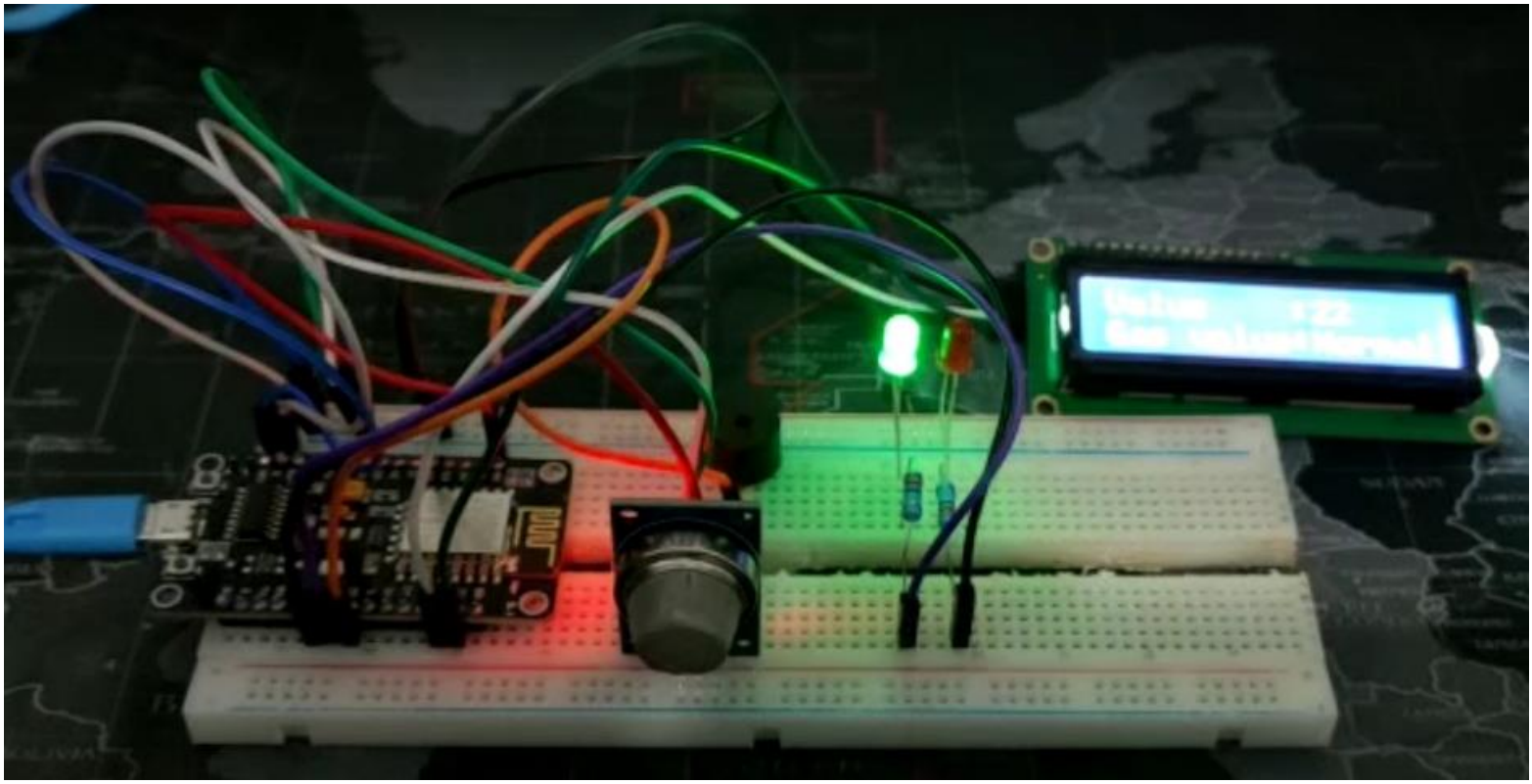
1. **Hardware:**
   - NodeMCU board ESP8266
   - MQ-2 gas sensor
   - LCD display 16x2
   - I2C module
   - Buzzer
   - Red LED
   - Green LED
   - Resistor
   - Breadboard
   - Jumper wires (Male to Male)
   - Jumper wires (Male to Female)
   - Micro USB Cable

2. **Software:**
   - Arduino IDE
   - Blynk app

***Note: pictures for each hardware and software are in the attachment section.**

Circuit Diagram





The Sketch

List of Codes:

```
/*GAS detector security system.
 */

#include <LiquidCrystal_I2C.h>  //The library allows to control I2C displays
#define BLYNK_PRINT Serial //It is just a redirection of built in Blynk "status prints"
#include <ESP8266WiFi.h>  //This library provides ESP8266 specific Wi-Fi routines
#include <BlynkSimpleEsp8266.h>
LiquidCrystal_I2C lcd(0x27, 16, 2); //set address I2C and the size of lcd 16×2 char

//char auth[] = " ";// Enter your Auth token
char auth[] = "…";
char ssid[] = "…";   //Wi-Fi name
char pass[] = "…"; //Wi-Fi password
BlynkTimer timer;  //to send data in intervals and keep the void loop()
int pinValue = 0;

#define Buzzer D5
#define Green D6
#define Red D7
#define Sensor A0

void setup() {
  Serial.begin(115200);//Sets the data rate in bits per second (baud) for serial data transmission.
  lcd.backlight();  //turn on the backlight
  lcd.begin();  //Initializes the interface to the LCD screen
  pinMode(Green, OUTPUT);
  pinMode(Red, OUTPUT);
  pinMode(Buzzer, OUTPUT);
  pinMode(Sensor, INPUT);
  Blynk.begin(auth, ssid, pass); //connect to set auth token, Wi-Fi
  timer.setInterval(100L, notifiaction);
}
BLYNK_WRITE(V0) { //The device can send data to the App using Blynk.
  pinValue = param.asInt();
}

void notifiaction() { //function for notification
  int sensor = analogRead(Sensor);
  Serial.println(sensor);
  sensor = map(sensor, 0, 1024, 0, 100);
  if (pinValue == 1) { //if button on
    if (sensor <= 50) { //gas level below 50
      digitalWrite(Green, HIGH);  //green led on
      digitalWrite(Red, LOW);  //red led off
```

```
    digitalWrite(Buzzer, LOW);  //buzzer off
    lcd.setCursor(0, 1);   //set character in second row
    lcd.print("Gas value:Normal"); //display in lcd
   } else if (sensor > 50) {
    Blynk.notify("Warning! Gas leak detected");
    digitalWrite(Green, LOW);
    digitalWrite(Red, HIGH);
    digitalWrite(Buzzer, HIGH);
    lcd.setCursor(0, 1);
    lcd.print("Gas value:High  ");
   }
   lcd.setCursor(0, 0);   //set character in first row
   lcd.print("Value    :");
   lcd.print(sensor); //call the sensor value and display
   Blynk.virtualWrite(V1, sensor);
  } else { //if the button off
   digitalWrite(Red, LOW);
   digitalWrite(Buzzer, LOW);
   digitalWrite(Green, LOW);
   lcd.clear();
  }
}

void loop() {
  Blynk.run();
  timer.run();
}
```

How it Works

In this project, we will measure the quantity of gas in numbers and send it over the internet using the Blynk App. With this system, the data can be monitored remotely staying in any part of the world. We just need a gas/smoke/LPG sensor which is MQ2 that is directly connected to Nodemcu ESP8266-12E Module. The Gas Sensor (MQ2) module is excellent for detecting gas leaks (in home and industry). It has the ability to detect combustible gas and smoke. When the gas concentrations increase, the Gas sensor's output voltage rises. Rotating the potentiometer changes the sensitivity. Firstly, the BLYNK_PRINT Serial will define which serial port will be used for the Blynk Debug later in the background. Then header file <ESP8266Wifi> will utilize all the functions needed for operating the NodeMCU over the internet. Then the header <BlynkSimpleEsp8266> is added for the working of the Blynk app with the NodeMCU(Esp8266) over Wifi. Then, the authentication token is inserted in the code, it is required to ensure security to make sure that the hardware connected with the PC is communicating with the correct Blynk Project and its user.

Afterward, our network credentials (SSID and password) are needed to establish a link between the hardware and the Blynk app over our network. The pin of the NodeMCU where the Positive (+) of the Buzzer is connected (called buzzer), and the pin of the NodeMCU where the pin of the MQ-2 is connected. If the sensor's output value (which is proportional to the amount of gas present around it) exceeds this value, the buzzer/alarm goes off. This must be set keeping in view the sensor's output value in a normal atmosphere. And the red LED will be on if the gas value is over, if everything is fine the green LED will be on. Also under this condition, the Blynk app is used to notify the message to the user.

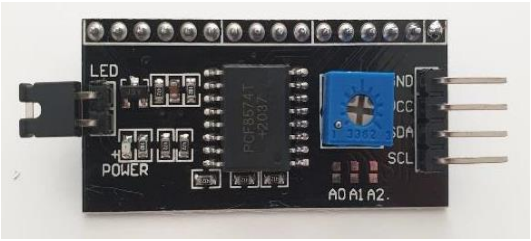
***Note: for more detail, please check the Google Drive link below (our presentation video).**

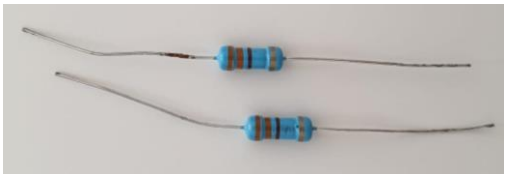
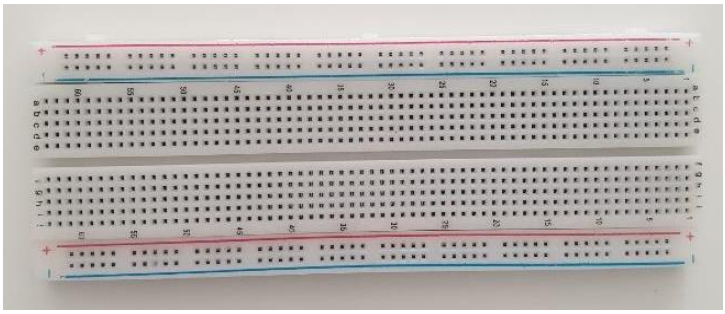
Google Drive Link:

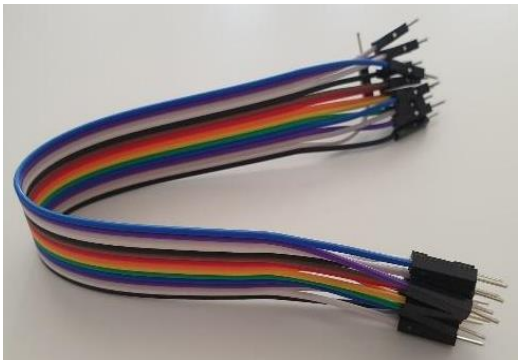
https://drive.google.com/file/d/1iRchZUtgxk5DeY9VDo9zbBNcYXftz2TZ/view

Attachment

Hardware:

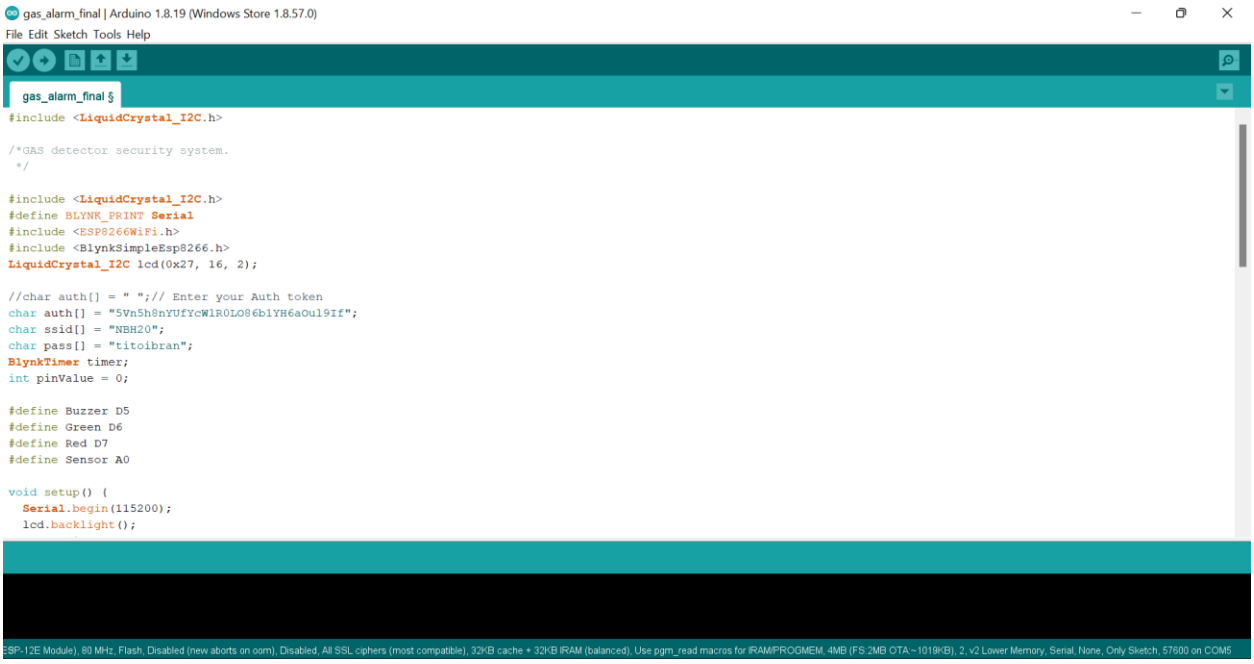
Name	Picture
NodeMCU board ESP8266	
MQ-2 gas sensor	
LCD display 16x2	
I2C module	

Buzzer	
Red LED	
Green LED	
Resistor	
Breadboard	

Jumper wires (Male to Male)	
Jumper wires (Male to Female)	
Micro USB Cable	

Software:

Arduino IDE



Blynk app