Namal Institute Mianwali

EE 300L: Embedded System

Open Ended Lab

Smoke Level Detector using MQ-135 with LCD Interfacing

Student's Name	Muhammad Faizan Ikram
Roll No.	2018-UET-NML-Elect-27
Date Submission	August 27, 2021
Marks Obtained	
Instructor's signature	

Instructors: Dr. Hamza Zad

Lab Engineer: Engr. Awais Yaqoob

Namal Institute Mianwali

Table of Contents

Namal Institute Mianwali

Introduction:

This report is about a Smoke Level Detector based Alarm System project which has been done with the help of MQ-135 Sensor and Arduino. We have used this smoke sensor module to measure the level of smoke in the environment. This module helps us to detect the concentration of smoke in environment and send data to Arduino using analog read function. These modules are widely used in various type of applications such as gas leakage detection, harmful gases concentration detection etc. In this project we have used it to measure the level of the smoke in environment. We set a threshold for the measured smoke level and based on this threshold value, we activate or deactivate the alarm system automatically. Thus, we achieved the alarm system objective using this module with Arduino.

Problem Statement:

"To design a Smoke Level Detector with Alarm system using MQ-135 and display data on LCD."

Experimental Tools:

1. Arduino Uno Board
2. LCD Display
3. Potentiometer
4. MQ-135 Sensor
5. Buzzer
6. LEDs

Environment (IDE):

1. Arduino IDE

Methodology:

LCD Interfacing and Working:

To achieve our objective, first of all, we build our circuit by interfacing LCD with Arduino. With have interfaced LCD with Arduino as following:

LCD Pins 1, 3, 5, 16 ——GND
LCD Pins 2, 16 ———— VCC (+5V)
LCD Pin 4 ———————— Arduino pin D7
LCD Pin 6 ———————— Arduino pin D6
LCD Pin 11 ————————- Arduino pin D5
LCD Pin 12 ————————- Arduino pin D4
LCD Pin 13 ————————- Arduino pin D3
LCD Pin 14 ————————- Arduino pin D2

These are the basic connection for interfacing the LCD with Arduino and can be changed depending on the type of circuit that we are going to design. The LCD is powered up by Arduino with operating voltage of 5V and start displaying data sent on given pins

MQ-135 Sensor Connection and Working:

After interfacing LCD, we connected the MQ-135 sensor with Arduino. The positive side is connected with Arduino +5V and ground is connected with Arduino ground. There are two other data pins for digital and analog data D0 and A0 respectively. Since we are measuring the analog value so we have used analog data pin and connected this with Arduino analog pin A0.

MQ-135 Module Sensor has lower conductivity **[1]** in clean air. When the target gas or smoke is exists, the sensor's conductivity is increased. This change in conductivity is converted to the corresponding output signal of smoke or gas concentration. The value read by using Arduino function of **analogread()** that reads analog value and execute analog to digital conversion (ADC) directly using the following formula **[2]**.

$$\frac{Resolution\ of\ the\ ADC}{System\ Voltage} = \frac{ADC\ Reading}{Analog\ Voltage\ Measured}$$

$$\frac{1023}{5} = \frac{x}{Analog\ Voltage\ Measured}$$

This formula relate to the Arduino Uno which has **10-bit** resolution (1023 bits) and system voltage is 5V **[3]**. So the ADC value **x** can be found based on analog voltage measured by sensor. It is also noticed that by changing measuring voltage the ADC values are also changed.

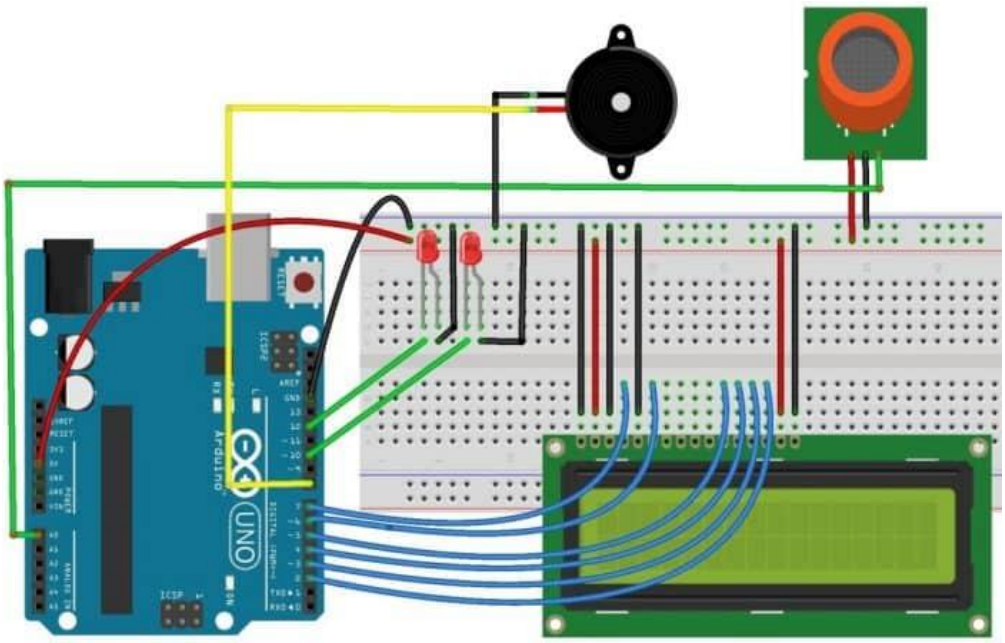
Bluetooth Module Connection and Working:

At the end, we connected the Buzzer and LEDs for Alarm purposes. The pin configuration is given as:

LED1 Pin +ve end ————Arduino PinD10
LED1 Pin -ve end ————GND
LED2 Pin +ve end ————Arduino PinD12
LED2 Pin -ve end ————GND
Buzzer Pin +ve end ————Arduino PinD8
Buzzer Pin -ve end ————GND

Based on ADC values and specified threshold, the alarm system works. When the ADC values are greater than the threshold, the Red LED turned on and Buzzer starts beeping and it prints “Alert!” on LCD. Otherwise in normal condition, the Green LED remains on and it prints “Normal” on LCD.

Circuit Diagram:



Smoke Level Detector Arduino Circuit Reference [1]

Code:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

int redLed = 10;
int greenLed = 12;
int buzzer = 8;
int smokeA0 = A0;
int feedback = 15
// Your threshold value
int sensorThres = 30;

void setup() {
  pinMode(redLed, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(smokeA0, INPUT);
  Serial.begin(9600);
  lcd.begin(16,2);
}
```

```
void loop() {
  int analogSensor = analogRead(smokeA0);
  Serial.print("Pin A0: ");
  Serial.println(analogSensor-feedback);
  lcd.print("Smoke Level:");
  lcd.print(analogSensor);
  // Checks if it has reached the threshold value
  if (analogSensor-feedback > sensorThres)
  {
    digitalWrite(redLed, HIGH);
    lcd.setCursor(0, 2);
    lcd.print("Alert....!!!");
    digitalWrite(greenLed, LOW);
    tone(buzzer, 1000, 200);
  }
  else
  {
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, HIGH);
    lcd.setCursor(0, 2);
    lcd.print(".....Normal.....");
    noTone(buzzer);

  delay(500);
  lcd.clear();
}
```

Code Working:

This code first sets the environment by defining the LCD pins and the pins for LEDs, Buzzer and Sensor. We have also specified the feedback for calibration and threshold value for comparison.

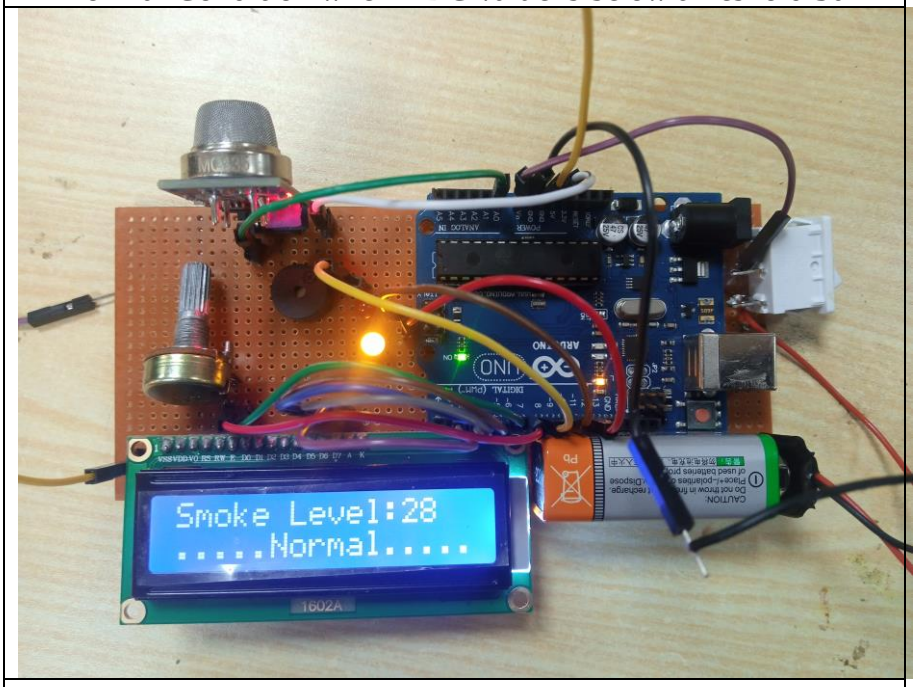
In **void setup()** part, we have defined the pinMode of the used pins. The pinMode is set OUTPUT for LEDs, and Buzzer, while the pinMode for smoke is INPUT as we are taking input from this pin A0.

In **void loop()** part, we have used analogRead() function which takes smoke input from pin A0 and then prints its calibrated values both on serial monitor and LCD. Then this value is also compared for alarm system with the help of **IF-ELSE** loop. If the calibrated value is greater than specified threshold value then Red LED is turned on and Buzzer starts beeping along with it also prints "Alert..!!" on LCD. Otherwise if the calibrated value is below the threshold value, it turns Green LED is turned on and Red LED is turned off. The buzzer is also stopped and it prints "Normal" on LCD.

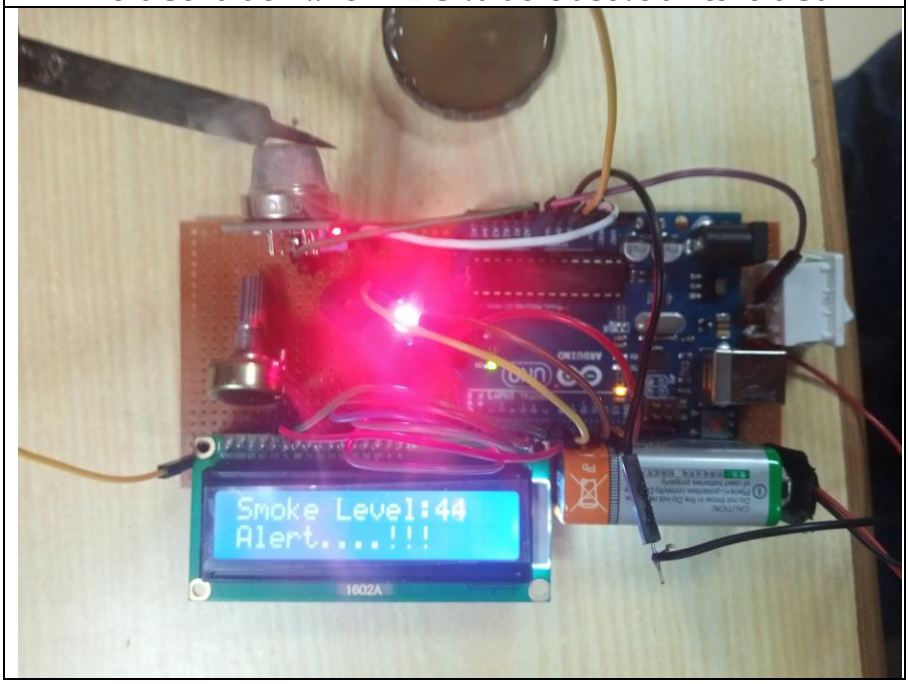
Results and Discussion:

Hardware circuit has shown exactly same results as we discussed earlier. The snapshots of the hardware implementation are given below:

Normal Condition when ADC value is below threshold 30



Alert Condition when ADC value is above threshold 30



The above pictures are practical demonstration of our **Smoke Level Detector using MQ-135 Sensor Module**. The Green LED is turned on in normal condition and prints Normal on LCD screen when the concentration of smoke is normal below the threshold in environment.

When the concentration of smoke is increased, the conductivity of sensor is increased and above the threshold the Red LED is turned on along with Buzzer beep sound and it prints Alert on LCD Screen.

This remains continue and upon each change of condition, it repeats same instructions. Thus, we achieve our desired results and Alarm system works perfectly as assumed. The MQ-135 gas sensor has high sensitivity to Benzene steam, Ammonia and Sulphide also sensitive to smoke and other harmful gases. It is with low cost and suitable for different applications such as harmful gases or smoke detection. So using this sensor we have achieve a low cost and economical alarm system with LCD display.

Conclusion:

At the end of this project, we conclude that using sensor like MQ-135, an alarm system can be made that is low cost and efficient with high sensitivity. We have achieved the system with the help of arduino and sensor with threshold set by the user. We applied the concepts and knowledge of previous labs of Arduino interfacing and programming required for this microcontroller. This systme is also useful for detection of other harmful gases or gas leakage detection.

References:

[1] *Arduino Smoke Level Detector using MQ-135 Sensor with Alarm.* [Online]. Available at: Arduino Smoke Level Detector using MQ-135 Sensor with Alarm (how2electronics.com)

[2] *Analog to Digital Conversion.* [Online]. Available at: Analog to Digital Conversion - learn.sparkfun.com

[3] *analogRead().* [Online]. Available at: analogRead() - Arduino Reference

Appendix-A:

Connections:

LCD Pins 1, 3, 5, 16 ——— GND
LCD Pins 2, 16 ————— VCC (+5V)
LCD Pin 4 ————————— – Arduino pin D7
LCD Pin 6 ————————— – Arduino pin D6
LCD Pin 11 ————————- Arduino pin D5
LCD Pin 12 ————————- Arduino pin D4
LCD Pin 13 ————————- Arduino pin D3
LCD Pin 14 ————————- Arduino pin D2
MQ-135 Module Pin -GND —— GND
MQ-135 Module Pin +VCC —— VCC
MQ-135 Module Pin A0 — Arduino Pin A0
LED1 Pin +ve end ————- Arduino PinD10
LED1 Pin -ve end ————-GND

LED2 Pin +ve end ————- Arduino PinD12
LED2 Pin -ve end ————-GND

Buzzer Pin +ve end ————- Arduino PinD8
Buzzer Pin -ve end ————-GND

Namal Institute Mianwali

Assessment Rubric for Lab

Method of Evaluation: Lab report and instructor observation during Lab Session.

Outcomes Assessed:

CLO1: Ability to analyze and extract meaningful information from observed data (P).

CLO2: Ability to function as an effective team member (A).

CLO3: Ability to follow instructions and convey experiment results in an effective manner (A).

CLO5: Ability to use the techniques, skills, and modern engineering tools necessary to practice control engineering (P).

Performance	Good (4-5)	Satisfactory (2-3)	Unsatisfactory (1)	Marks
Task Completion (CLO5)	All tasks completed correctly	Most tasks completed correctly with some incomplete or incorrect	Most tasks either incomplete or incorrect	
Teamwork (CLO2)	Actively engages and cooperates with other group members in an effective manner	Cooperates with other group members in a reasonable manner	Distracts or discourages other group members from conducting the experiment	
Lab Report (CLO3)	Lab report has been filled in neatly with proper grammar and scientific terminology	Parts of the report not filled in neatly or with improper grammar	Report filled in illegible writing with improper pucntuations or a casual, non-scientific tone	
Discussion (CLO1)	Provides meaningful interpretation of results. Provides scientific reasoning and draws appropriate conclusions based on data.	Provides some interpretation of results. Includes some sort of conclusions.	Interpretation of results is not clearly explained. No conclusions are made based on the results.	
Validating results from instructor (CLO3)	Shows the obtained graphs/results at all points indicated in the manual.	Shows the graphs/results at some of the points indicated in the manual	Does not show the graphs/results at all.	
Difference of report from Lab Partner (CLO2)	The report is clearly filled in independently of the lab partner with same findings but with distinct text.	Most of the report is filled in independently. There is overlap with the lab partner in text of some answers	Significant overlap of report with lab partner. Clearly not filled in independently.	
			Total	