# PROJECT REPORT

**Topic:** Fire Alarm Using PIC Microcontroller and Gas Sensor

**Course:** CSE 331

**Section:** 4

**Faculty Initial:** RSF

**Semester:** Fall 2019 (193)

**Submission Date:** 31st December, 2019

## Group Members:

| Name | ID |
|---|---|
| Sheikh Tahmim Mehdi | 161 2139 042 |
| Mustafizur Rahaman | 171 2847 642 |
| S. M. Al Faruqui | 172 1395 042 |
| Ankit Dev | 172 1512 042 |

# Acknowledgement

We would like to express our special thanks and gratitude to our teacher, Rishad Rfin who gave us the opportunity to make a fire alarm system using a microcontroller. We would also like to thank our friends who helped us greatly with their valuable comments, constructive criticism and suggestions. The community of stackoverflow, geekforgeeks have greatly helped us in learning everything we needed to know about PIC micro-controller and mikro c programming and progress with our work.

Working on this project has helped us in gaining in dept knowledge and experience regarding embedded systems. We are very thankful for this wonderful opportunity.

# Table of Contents

# Abstract

This report is about creating a fire alarm system using the popular PIC16F877A microcontroller. Our report primarily focuses on the process of creation of this system and less so on its application. We also briefly discussed about the major components used in our project. The code written to program our microcontroller is included in the procedure section.

We used PIC16F877A as the microcontroller, MQ-2 gas sensor to detect flammable gas, 16X2 LCD to show the status and a buzzer. When there is flammable gas nearby the buzzer will ring.

# Introduction

PIC microcontrollers (Programmable Interface Controllers), are semiconductor devices that can be programmed to carry out a vast range of tasks. We are making a fire alarm system using out PIC16F877A microcontroller. We are using a MQ-2 gas sensor to detect the presence of flammable gas. Our fire alarm system is cheap and easy to make. It also requires very small space. Which makes it feasible to be implemented in home or office space.

## Objectives

The objectives of our project are as follows:

- Learn about the PIC16F877A microcontroller
- Learn the similarities and difference between PIC microcontroller and MPU8051
- Learn Mikro C programming and proteus simulation
- Implement PIC microcontroller to solve a real-world problem
- Create a simple fire alarm system with PIC microcontroller

# Application

- Can be used as a fire alarm in home and offices
- Can be used to detect combustible gas like methane and butane
- Detect industrial gas leakage

We can extend our project to detect the amount of gas present in air. We can also include humidity sensor, temperature sensor and flame sensor to make our fire alarm more accurate and reliable.

# Block Diagram

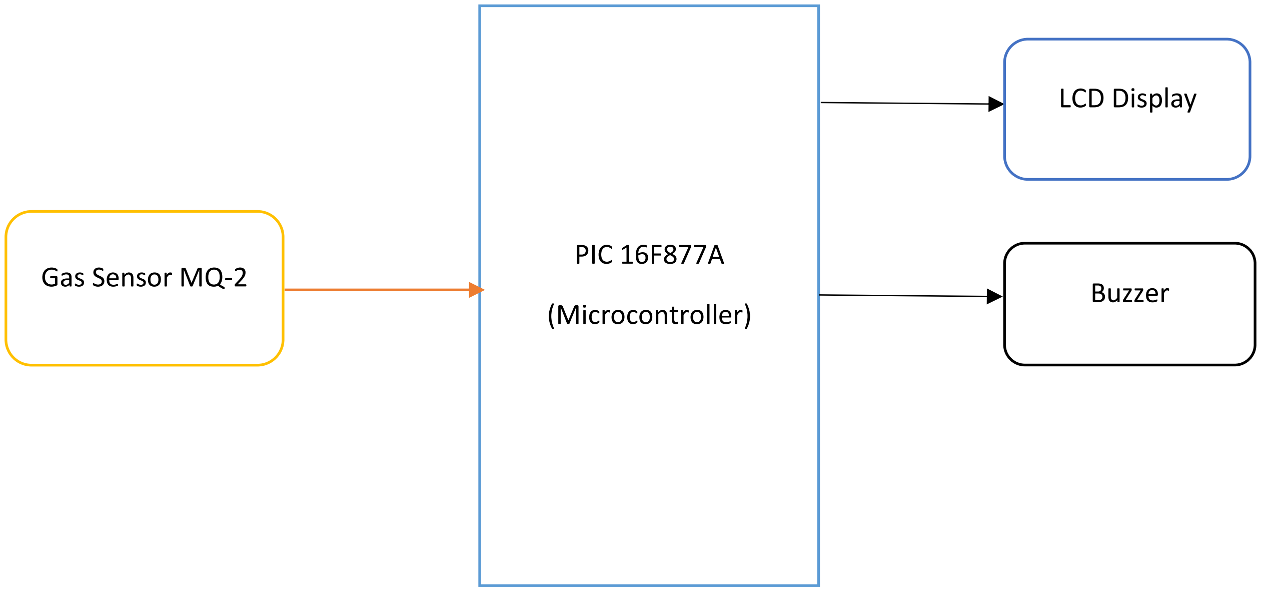


Fig. Block diagram of fire alarm

# Requirements

## Software Requirements

| | |
|---|---|
| ➢ MikroC PRO for PIC (Free Version) | [For programming] |
| ➢ Proteus 8 Professional | [For simulation] |
| ➢ PICkit 3 v3.10 | [For burning the code into MCU] |

## Hardware Requirements

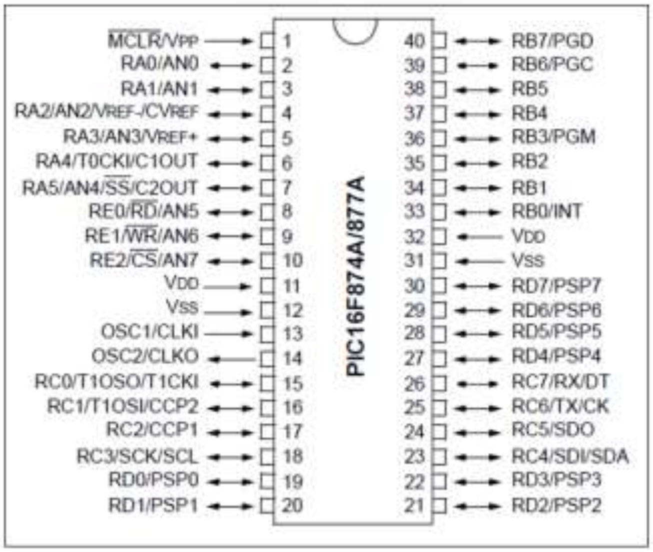
- PIC16F877A
- PICKit 3
- Gas Sensor MQ-2
- Buzzer
- Resistors, Potentiometer, Capacitors, Voltage Regulator
- LCD display

## Cost of components

| # | Items | Quantity | Rate (BDT) | Amount (BDT) |
|---|---|---|---|---|
| 1 | PIC16F877A | 1 | 182.8 | 182.8 |
| 2 | Gas Sensor MQ-2 | 1 | 139.9 | 139.9 |
| 3 | LM7805 Voltage Regulator (5V) | 1 | 7.9 | 7.9 |
| 4 | 8MHz Crystal Oscillator | 1 | 6.9 | 6.9 |
| 5 | 10k ohm Potentiometer | 1 | 13.8 | 13.8 |
| 6 | 9V Battery (PAKKO) | 2 | 40 | 80 |
| 7 | Buzzer (5V) | 1 | 16.1 | 16.1 |
| 8 | I2C LCD Display (16x2) | 1 | 272.9 | 272.9 |
| 9 | Breadboard (Big) | 1 | 84.9 | 84.9 |
| 10 | Jumper Wire Set (20) | 1 | 29 | 29 |
| 11 | PIC Adapter for PICkit 3 | 1 | 369 | 369 |
| 12 | PICkit 3 Burner | 1 | 1589 | 1589 |
| | **Total Cost** | | | |

## PIC16F877A:

This PIC16F877A is an 8bit microcontroller. That means it can deal with 8bits of data at a time. There are 40 pins. Number of I/O's are 33. Its minimum supply voltage is 4V and maximum supply voltage is 5.5V. Number of pins are 40. CPU speed can be up to 20 MHz. No. of ADC inputs are 8. RAM size is 368Byte.



## MQ-2 Gas Sensor:

MQ2 gas sensor is an electronic sensor used for sensing the concentration of gases in the air such as LPG, propane, methane, hydrogen, alcohol, smoke and carbon monoxide.MQ2 is a metal oxide semiconductor type gas sensor. Concentrations of gas in the gas is measured using a voltage divider network present in the sensor. This sensor works on 5V DC voltage. It can detect gases in the concentration of range 200 to 10000ppm.

# Procedure

Firstly, we had to learn about Proteus and Mikro C programming. We started with simple circuits that had a single battery, resistor and LED and moved on to building more complex electric circuits.

Proteus was used to make a simulation of our project. We added a PIC16F877A microcontroller, a 16x2 LCD, a MQ-2 gas sensor and a buzzer. (The connections are show in the next section). We needed 6 port pins of the PIC MCU for the LCD. 4 pins for data transmission and 2 pins for enable and register select. In our project the 16x2 LCD works in 4bit mode. It can also work in 8-bit mode, but we chose to go with 4bit mode as it requires less wires and we can save some pins in our MCU. However, 8bit mode is twice as fast. The gas sensor was connected to an input port and the buzzer was connected to an output port. So, in total we used 8 port pins. We also connected a 2-pin crystal oscillator for clock pulse. However, the simulation ran fine without the oscillator.

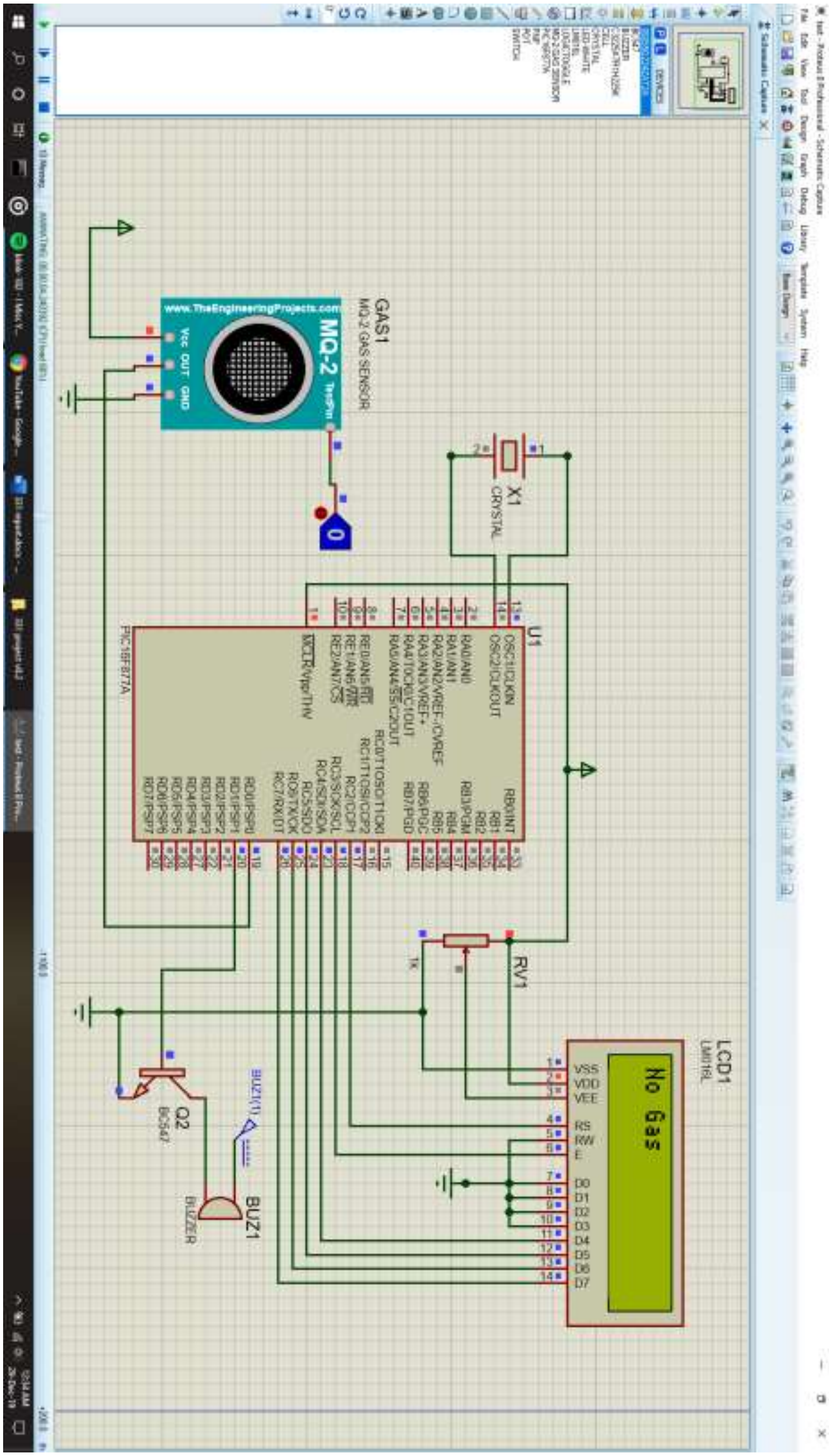
We wrote our fire alarm program in MikroC IDE. The syntax was very similar to C language. For 16x2 LCD interface code we sought the help of MikroC documentation. Then we compiled the doe. Compiling the code gave us a HEX file. We then uploaded the HEX file onto the PIC16 microcontroller in Proteus. We ran the simulation, and everything worked as expected.

Then, we used our adapter and PICkit3 Microcontroller Programmer to burn the code into our PIC microcontroller. We mounted the PIC16F877A and 16x2 LCD onto the breadboard and connected everything according to the simulation. We had a 9V battery, but all our components work at 5v. So, we had to use a voltage regulator to convert the 9V DC to 5V DC.

We used an 8MHZ crystal oscillator for clock pulse and a 10K ohm potentiometer to control the brightness of the LCD.

The MQ-2 gas sensor has 2 output pins. Ao(Analogue Out) and Do(Digital Out). For our project we used the Do pin, it gives LH (Logic High) in presence of certain combustible gases. So, when the gas sensor detected methane or butane gas it gave LH to the input pin and we got out output through the buzzer and LCD.

## Proteus Simulation



## MikroC Code

```
// LCD module connections
sbit LCD_RS at RC2_bit;
sbit LCD_EN at RC3_bit;
sbit LCD_D4 at RC4_bit;
sbit LCD_D5 at RC5_bit;
sbit LCD_D6 at RC6_bit;
sbit LCD_D7 at RC7_bit;

sbit LCD_RS_Direction at TRISC2_bit;
sbit LCD_EN_Direction at TRISC3_bit;
sbit LCD_D4_Direction at TRISC4_bit;
sbit LCD_D5_Direction at TRISC5_bit;
sbit LCD_D6_Direction at TRISC6_bit;
sbit LCD_D7_Direction at TRISC7_bit;
// End LCD module connections


sbit GAS at RD0_bit;                          // gas sensor is connected to RD0 pin

void main() {


  Lcd_Init();                          // Initialize LCD

  Lcd_Cmd(_LCD_CLEAR);                        // Clear display
  Lcd_Cmd(_LCD_CURSOR_OFF);                   // Cursor off

  TRISD0_bit=1;                               //Set RD0 as input
  TRISD1_bit=0;                               //Set RD1 as ouput

  while(1){
          if(GAS == 1){

              Lcd_out(1,1, "Gas Detected" );
              PORTD.F1 = 1;                   //Turn on buzzer
          }
          else {

             Lcd_out(1,1, "No Gas       " );
             PORTD.F1 = 0;                    //Keep buzzer off
          }
    }
}
```

## Hardware Implementation

# Discussion

The first problem we faced was with the Proetus simulation. Everything was working as expected except the buzzer. The buzzer didn’t produce any sound in our simulation. Initially we tested our simulation using an LED in place of the buzzer, so we were sure everything was done correctly. We fixed the buzzer problem by using an NPN transistor. However, during hardware implementation we didn’t have to use an NPN transistor.

We only had 9V battery but PIC16F977A operates at 5V input voltage. So, we used a voltage regulator to convert our 9v battery into 5v output.

The proteus simulation didn’t have the VCC and GND pins for the PIC16 microcontroller. So, we were a bit confused to see 2 VDD(+5v) and 2 VSS(GND) in our hardware. According to the documentation, it is recommended to connect 2 VCCs and 2GNDs with the microcontroller to better stability and performance. However, as we used low clock speed (8MHZ oscillator) we could have used only one pair of VCC and GND pin.

The MCU doesn’t work properly unless the Pin-01 (MCLR`/VPP) is connected to a 5V terminal. Which is another thing that didn’t occur in our simulation. We then burned a simple LED blinking program onto our MCU for testing, and it worked properly.

Our MQ-2 gas sensor wasn’t producing any digital output. Our sensor’s Dout pin didn’t give any Logic High output in presence of butane (form a lighter) and methane (from kitchen stove). Our entire project is based on the Do of the gas sensor.

We also faced some problem with the 16x2 LCD display. It wasn’t showing our desired output. We tried everting and double check the connections but still could not find the source of our problem. Fortunately, the 16x2 LCD isn’t crucial to our project. As long as the buzzer was working properly, our project should be able to function as a fire alarm system.

We can extend our project to detect the amount of gas present in air using the analogue output pin of our MQ-2 gas sensor. We can also include humidity sensor, temperature sensor and flame sensor to make our fire alarm more accurate and reliable.