EE322: Embedded Systems Design - Project

GAS LEAKAGE / SMOKE DETECTOR

Project Progress Report

GROUP G26

Date of Submission : 20/08/2021
Reg No E/17/043 : CHULAWANSA S.M.D.S.
Reg No E/17/183 : LAKSHAN G.A.D.
Reg No E/17/235 : PARANAWITHANA N.N.K.

Page Index **Page number**

List of Figures

List of Tables

Gas Leakage / Smoke Detector
Progress from 30/07/2021 to 20/08/2021

Overall percentage progress

0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩				

Introduction

We were asked to do an embedded system project in EE 322 as a group using PIC Microcontroller. So, we decided make an embedded system for a day today application.

In this project we decided to make a system to detect domestic, industrial smoke or gas leakage. It can be also used to detect combustible gas leakage. To make our project a success we decided to use PIC16F877A microcontroller to fulfill our tasks. Decision was made to use PIC16F877A microcontroller, because it has quite a bit more I/O than PIC16F84A and has about more RAM & FLASH.

Mainly this project can also introduce as a fire alarm system. In here what we do is, we detect a smoke or gas leakage using sensor and simply display it on a LED screen. (Can use an additional buzzer to make a noise).

Brief of past progress (up to from date of this progress report)

After the submission of the project proposal, the group had worked according to a plan. Mainly we had worked on the assembly programming because the components we were planning to use had to be coded specifically. First few weeks we designed the Proteus simulation circuit & worked on programming using it. Main issue we had to face during this period is to buy the components. Because of that we kept our hardware implementation part for later. But the sensor was not available & even the online shopping was not efficient these days.

Since, other components were bought, the group plan for the upcoming period was to work on the output part programming until the sensor is purchasable. We decided to complete the whole programming part & compute the implementation using the software if sensor is bought.

Progress for the period from 30/07/2021 to 20/08/2021

The group was working mainly on the assembly program in last few weeks. Therefore, the code for the outputs that were planned to get from the PIC was done. For the coding of the inputs, a calibration procedure was required. Since, ordered MQ-2 sensor was not purchasable in these days because of the delivering issues caused by the situation in the country, we had to look for the local market again in order to do the calibration process. Shops we closed due to the increasing pandemic situation & we had to wait. But, up to date the sensor was not available.

Our main objective was only to complete the software implementation of the whole program during this period because gatherings were limited & we had few issues on the PIC kit & the hardware implementation process. But unfortunately the calibration process was required to continue the progress. We looked for some specific gas values obtained previously using the internet but that also was unsuccessful.

Therefore, for this progress period we had to settle for what we have programmed without correct input values. The main program with the outputs were programmed using MPLABX IDE. The completed work is shown below with the Proteus simulation.

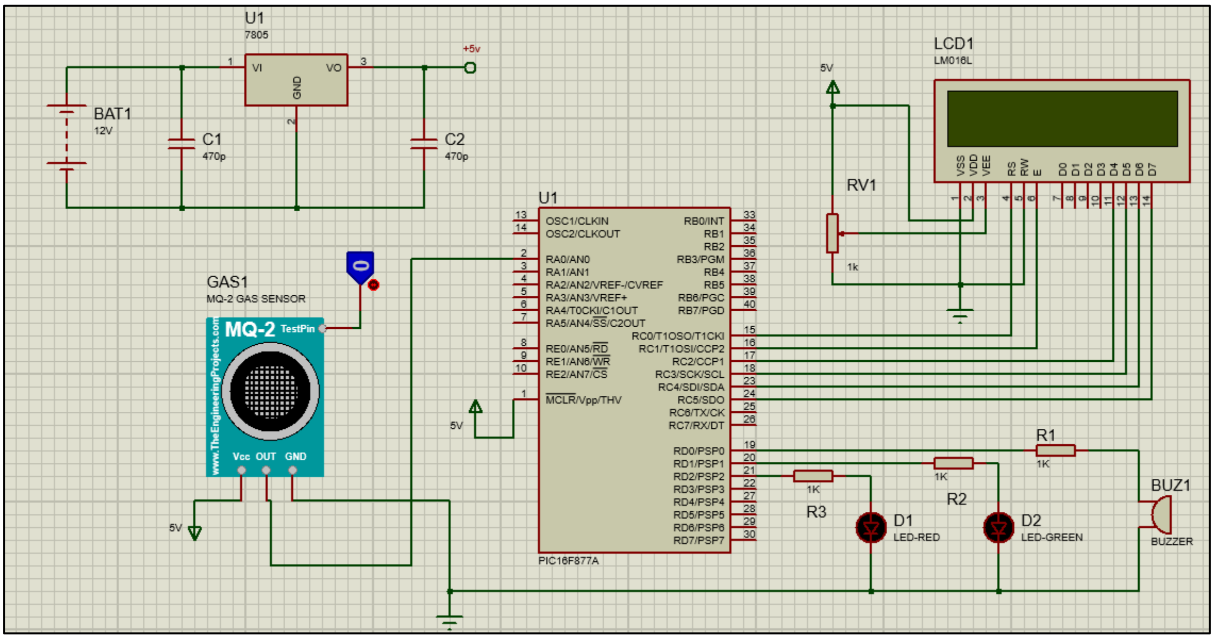

Figure 1 : Circuit implementation using Proteus

Since we could not calibrate the MQ2 sensor, We used the circuit as below

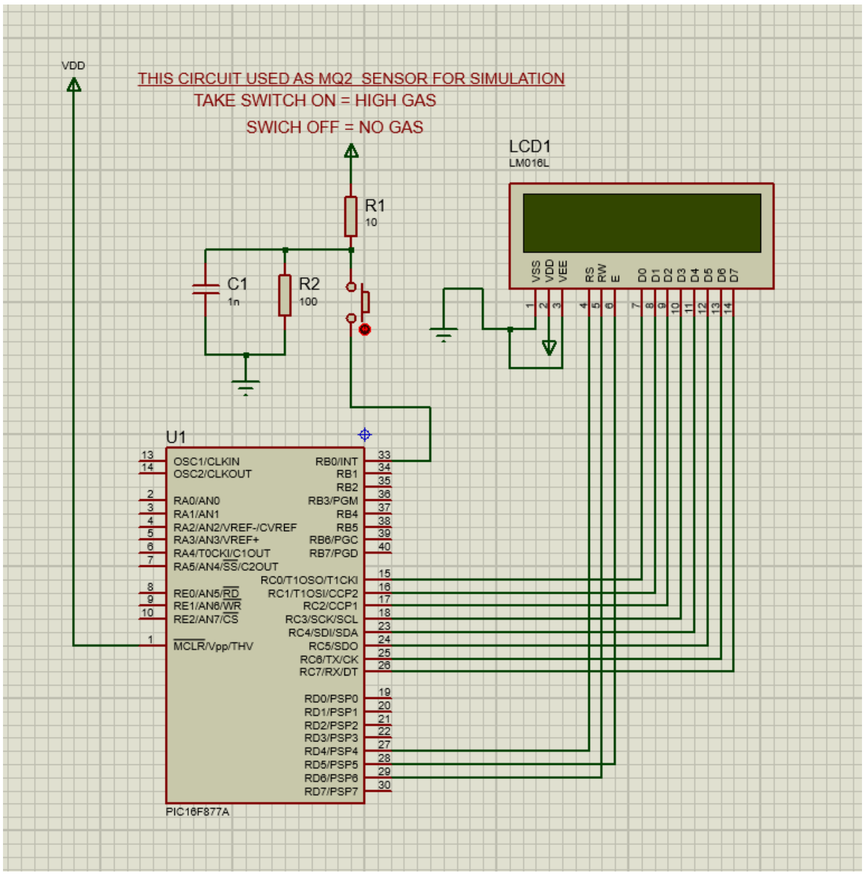

Figure 3 : Circuit implementation using Proteus for Simualation; Test 1

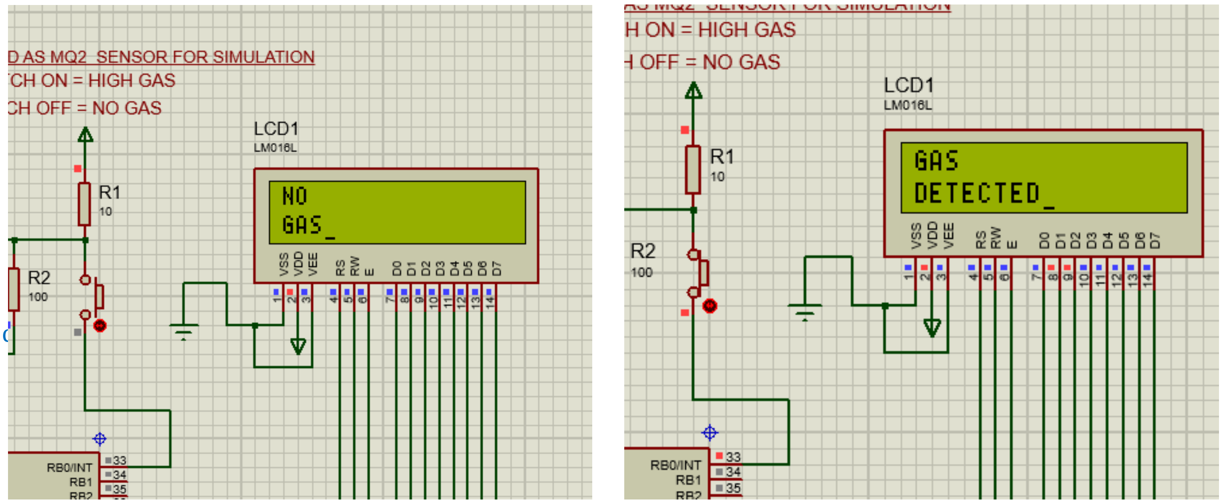

Figure 3 : Simualation at Gas Leak and No Gas Leak ; Test 1

For simulation below code was used. Mainly we used LCD to display the result. Further, we hope to improve this simulation using an alarm(buzzers). And hope to do simualation with MQ2 GAS sensor.

```
List p=16f877A                          ;List directive to define processor
Include "P16F877A.INC"                  ;Processor specific variable definitions
 __CONFIG  _CP_OFF&  _DEBUG_OFF&  _WRT_OFF&  _CPD_OFF&  _LVP_OFF&  _BODEN_OFF&
_PWRTE_ON& _WDT_OFF& _XT_OSC

    RS        EQU 0x04 ;RD4
    E         EQU 0x05 ;RD5
    RW        EQU 0x06 ;RW
    ;MQ2        EQU 0x01 ;


    CBLOCK 0x20                              ;Counter GFRs register adresses
        Count
        Countx
    ENDC

    org     0x00                             ;Reset Vector
    goto     Main
    org     0x05

Main:
        ; bsf       TRISA,MQ2
          bsf       STATUS,5                 ;switch to bank1
          clrf      TRISD                    ;RD0 to RD7 all are outputs
          clrf      TRISC                    ;RB0 to RB7 all are outputs
          bcf       STATUS,5                 ;switch back to bank0

Start:
    call      LCDinitailize
    btfss     PORTB,0
    goto       Nogas
    goto       Gasdetected


LCDinitailize:

;PINS in LCD to PIC16F877A PORTB PINS
    ;DB7 (14) -----RC7(40)
    ;DB6 (13) ----RC6(39)
    ;DB5 (12) ----RC5(38)
    ;DB4 (11) ----RC4(37)
    ;DB3 (10) ----RC3(36)
    ;DB2 (9)---- RC2(35)
    ;DB1 (8) ----RC1(34)
    ;DB0 (7) ----RC0(33)
    ;E (6) ------RC5(28)
    ;RW (5) -----RC6(29)
    ;RS (4) -----RC4(27)
    ;Vo (3) -----+5V
    ;Vdd (2) ----+5V
    ;Vss (1) -----GND

    clrf  PORTC
    clrf  PORTD

    ;LCD routine starts
    call  Delay
    call  Delay
    ;give LCD module to reset automatically
    ;Fundtion for 8-bit, 2-line display, and 5x8 dot matrix
```

```
        movlw     0x38
        call      Instwrite

        ;Display On, Cursor On, No blinking
        movlw     0x0E
        call      Instwrite

        ;DDRAM address increment by one & cursor shift to right
        movlw     0x06
        call      Instwrite

        ;DISPLAY CLEAR
        movlw     0x01
        call      Instwrite

        ;Set DDRAM ADDRES
        movlw     0x80 ;00
        call      Instwrite
        Return


Gasdetected:
        ;WRITE DATA in the 1st position of line 1
        ;Characters (G, A and S)
        movlw     0x47 ;G
        call      Datawrite
        movlw     0x41 ;A
        call      Datawrite
        movlw     0x53 ;S
        call      Datawrite

        ;Set DDRAM address for the next (D, E, T, E, C, T, E and D) in line 2
        ;Set DDRAM address for the 1st position of line 2 (40h)
        movlw     0xC0
        call      Instwrite ;RS=0

        movlw     0x44 ;D
        call      Datawrite
        movlw     0x45 ;E
        call      Datawrite
        movlw     0x54 ;T
        call      Datawrite
        movlw     0x45 ;E
        call      Datawrite
        movlw     0x43 ;C
        call      Datawrite
        movlw     0x54 ;T
        call      Datawrite
        movlw     0x45 ;E
        call      Datawrite
        movlw     0x44 ;D
        call      Datawrite


        GOTO      Start



Nogas:
        ;WRITE DATA in the 1st position of line 1
        ;Characters (N and O)
        movlw     0x4E ;N
        call      Datawrite
        movlw     0x4F ;O
        call      Datawrite

        ;Set DDRAM address for the next characters (G, A and S) in line 2
        ;Set DDRAM address for the 1st position of line 2 (40h)
        movlw     0xC0
        call      Instwrite ;RS=0

        movlw     0x47 ;G
        call      Datawrite
        movlw     0x41 ;A
        call      Datawrite
        movlw     0x53 ;S
        call      Datawrite

        GOTO       Start

;subroutine to write instructions (Instwrite), Instruction to be written is stored in W before the call
Instwrite: movwf     PORTC
           call      Delay                  ;delay may not be needed
           bcf        PORTD,RS
           call      Delay
           bsf        PORTD,E
           call      Delay
           bcf        PORTD,E
           call      Delay
           return

;Subroutine to Write Data
Datawrite:      movwf     PORTC
                call      Delay             ;delay may not be needed
                bsf       PORTD,RS
                call      Delay
                bsf       PORTD,E
                call      Delay
                bcf       PORTD,E        ;Transitional E signal
                call      Delay
                return


;Delay Soubroutine
Delay:
           movlw     D'10'
           movwf     Countx
Delayloop:
           decfsz    Count,1
           goto      Delayloop
           decfsz    Countx,1
           goto      Delayloop
           return

END
```

Cost Analysis

Task	Budgeted cost (Rs.)	Expenses up to 30/07/2021 (Rs.)	Expenses from 09/07/2021 to 30/07/2021 (Rs.)	comments
Buying PIC16F877A microcontroller	520.00	600.00		Ordered PIC was more expensive than the budgeted cost.
Buying the Gas Sensor	300.00	-	415.00	Gas sensor was not purchased yet
Buying other circuit components 1. Resistors -10.00 2. LEDs & Buzzer - 50.00 3. LCD Display -600.00 4. Breadboard – 250.00 5. Wires / Cables -160.00	1070.00	950.00	-	Less than the budgeted cost. LCD display – 480.00
Buying components for the PIC programmer & other components required for circuit	1110.00	130.00		PIC programmer was ordered but not available online up to this date.
Total cost	3000.00	2095.00	415.00	-

Table 1: Cost Analysis

Time line (Gantt chart)

🟦 Planned execution time of the task as of the initial proposal
🟥 Actual execution time of the task due to delays etc.

Task		Week										
		1	2	3	4	5	6	7	8	9	10	11
1ST Discussion among group members	Identify a problem	🟦 🟥	🟦 🟥									
	Find a solution & Decide a Project title		🟦 🟥									
Project proposal				🟦 🟥								
Buy & Order materials online			🟦	🟦	🟦 🟥	🟥			🟥	🟥	🟥	
Basic project design					🟦 🟥	🟦 🟥						
Working on assembly programming part					🟥	🟦 🟥	🟦 🟥	🟦 🟥				
Modify the basic schematic design according to the design						🟥	🟦 🟥	🟦 🟥	🟥	🟥		
Circuit Implementation & Troubleshooting									🟦	🟥	🟥	
Hardware design & testing										🟦	🟦	🟦
Finalization & report												🟦

Table 2: Gantt chart