Flame Sensor Interfacing with 8051 MC

A Course Project Report

Submitted to the APJ Abdul Kalam Technological University

in partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Electronics and Communication Engineering

by

Bristo C J

(TVE21EC022)

Computer Architecture and Microcontrollers

ECT 206

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING TRIVANDRUM

KERALA

June 2023

DEPT. OF ELECTRONICS & COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING TRIVANDRUM

2022 - 23

CERTIFICATE

This is to certify that the report entitled **Flame Sensor Interfacing with 8051 MC** submitted by **Bristo C J** (TVE21EC022) to the APJ Abdul Kalam Technological University in partial fulfillment of the B.Tech. degree in Electronics and Communication Engineering is a bonafide record of the course project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Joaquim Ignatious Monteiro
Assistant Professor
Dept.of ECE
College of Engineering
Trivandrum

Abstract

This document contains essential templates required to write technical reports using LaTeX. This template may be used for the preparation of B.Tech seminar reports of APJ Abdul Kalam Technological University, Kerala. Also minimum working examples to create equations, include figure, include table, table of contents symbols list and bibliographic citation in a LaTeX document are provided.

Please note that this template is provided without warranty on an AS IS basis.

JIM

Contents

List of Figures

Chapter 1

Introduction

Flame sensor interfacing with 8051 micro controller is a fascinating project that involves the integration of advanced sensor technology with a powerful micro controller. This project aims to detect flames using a flame sensor and to process the data obtained from the sensor using an 8051 micro controller.

In this project report, we will discuss the design, implementation, and testing of the flame sensor interfacing with 8051 micro controller. We will also discuss the various challenges faced during the project and how they were overcome. The report will provide a detailed explanation of the hardware and software used, the methodology employed, and the results obtained.

1.1 Problem statement

Fire is one of the most dangerous events possible. Fire is destructive, and the smoke from a fire creates a toxic, dangerous atmosphere.The problem addressed by the project is the detection of fires in various settings, such as homes, offices, and industrial plants. Traditional methods of detecting fires, such as smoke detectors, may not be effective in certain situations, particularly when the fire is not accompanied by smoke. The rapid detection of a fire and its control can save lives and property.

1.2 Project objectives

The project aims to develop a cost-effective and efficient solution for detecting fires by using a flame sensor and an 8051 micro controller. The goal is to design a prototype that can accurately detect the presence of a flame and trigger an appropriate response which is a LCD display in this case.

The completion of the project will grant immense knowledge on the working of 8051 micro controller ; interfacing it with a LCD display and sensors,also improves our usage of software like Keil [5], MCU 8051IDE,Arduino IDE [6],Proteus .

Chapter 2

Project Design and Testing

The project design and testing requires several important steps including Code development in a suitable IDE,testing,simulating using a suitable simulation software.

On to the hardware implementation part we need to burn the code as a hex file to the 8051 micro controller and connect the resources needed for working of the prototype.

2.1 Code development : Keil uVision

Embedded C was chosen for programming the micro controller, as ALP will be a herculean task. Base code for the program was available in Embetronicx [1].I initialised necessary ports to their variable names .

Port 1 bit 0 is taken as sensor output.

Port 3 is taken as data pins to the LCD.

port 2 bit 0,1,2 are register select,read/write,enable pins respectively.

we make use of if else statements to recognize when the sensor output is high and thus display a warning message on the LCD.

2.1.1 Testing : keil

We translate , build and rebuild before debugging in order to find that no errors were produced during the compilation of code.

Further we debug the code . Also choosing to make the port outputs visible help in understanding the port movement upon the running of code.Below figure shows how

the port movements can be realised .

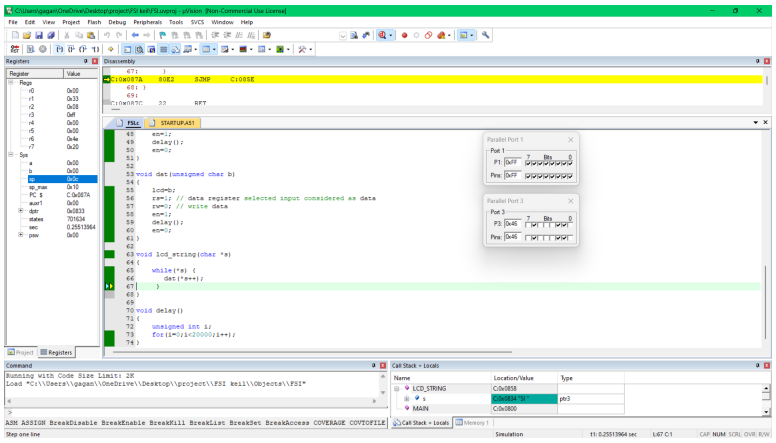


Figure 2.1: code debugging : Keil

2.1.2 Hex file : Keil

In order to use the 8051 micro controller we need to create a hex file of the code we have made and burn it into the micro controller. Keil uVision helps in creating a hex file . Figure below shows the hex file generated using keil.

```
:030000000208DA19
:0C08DA00787FE4F6D8FD75810A02080062
:1008320020465349200046 6C616D65204465746574650D
:10084200637465640020202020202020202020A6
:0608520020202020200000
:1008000012087D7BFF7A087932120858309013 7FE6
:10081000C01208BE7BFF7A0879381208581208AF58
:1008200080EA7FC01208BE7BFF7A08794712085819
:0208300080DA6C
:10087D007F381208BE7F0E1208BE7F061208BE7F9B
:09088D00011208BE7F800208BEC2
:0E08BE008FB0C2A0C2A1D2A21208AFC2A22265
:0E08CC008FB0D2A0C2A1D2A21208AFC2A22247
:100858008B088A09890AAB08AA09A90A12089660AE
:1008680013050AE50A7002050914F9120896FF1221
:0508780008CC80E22223
:0E08AF00E4FFFE0FBF00010EBE4EF8BF20F5A5
:0108BD002218
:10089600BB010689828A83E0225002E722BBFE0260
:0908A600E32289828A83E4932293
:00000001FF
```

Figure 2.2: Hex file via keil

2.2 Simulation : Proteus

Proteus is powerful software which we can simulate projects using electronic components seamlessly .Proteus contains all necessary components like the 8051 micro controller,capacitors,oscillator crystals, sensors and everything necessary.

In order to use a micro controller in proteus we need the hex file of the code .we make use of the hex file from keil for this.below figure shows the proteus model of the project .

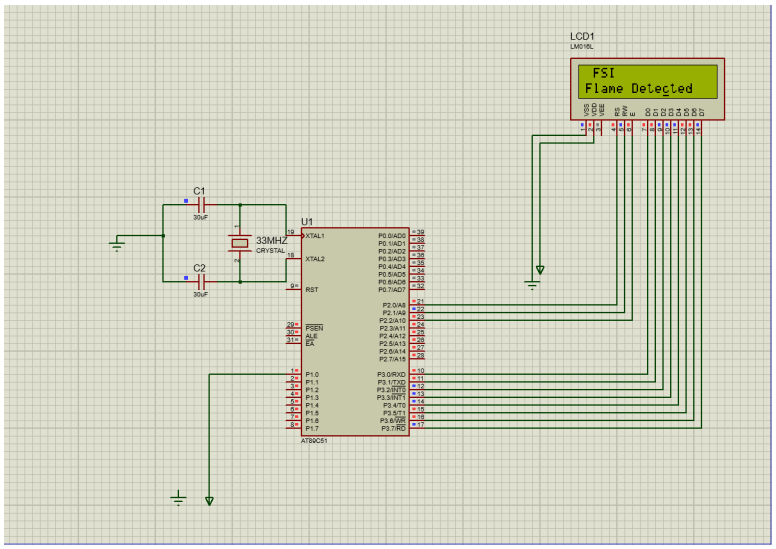


Figure 2.3: Simulation : Proteus

2.3 Burning code : 8051

2.3.1 preparation : Arduino IDE

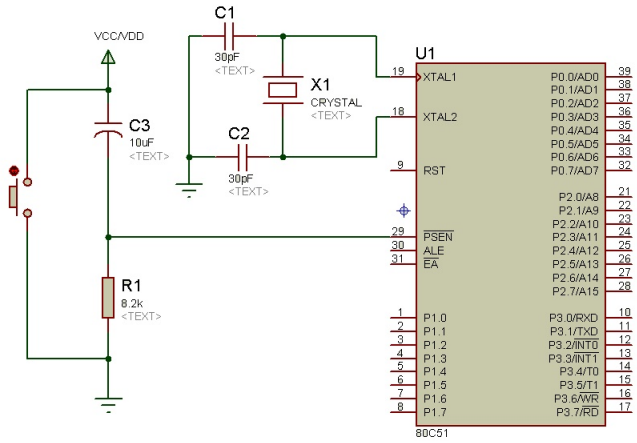
we obtained the sketch of AT89S52 [2] . Compiled it and uploaded it to the Arduino UNO .we prepare the hex file to be used using Keil.

2.3.2 Connection : Arduino and 8051

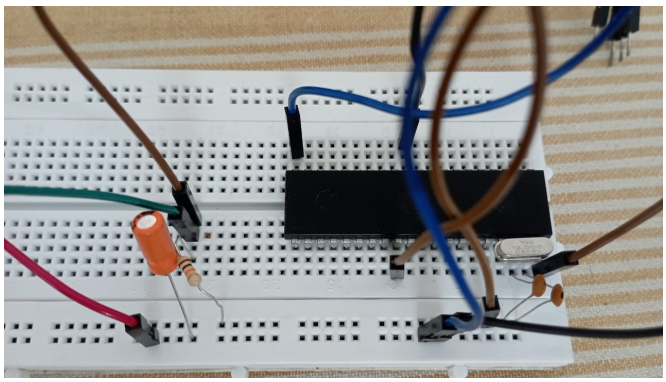
We provide circuitry for basic working of 8051 micro controller .

This include 2x33pF Capacitors,12MHz crystal oscillator,Reset circuit,Source and Ground connections.

The inter-pin connections are as follows



(a) Reset circuit



(b) Basic circuit

Figure 2.4: Basic Circuitry

Connect Pin 2 of arduino to RST pin or pin 9 of 89s52 micro controller.

Connect Pin 3 of arduino to pin no. 8 of 89s52 micro controller.

Connect Pin 4 of arduino to pin no. 7 of 89s52 micro controller.

Connect Pin 5 of arduino to pin no. 6 of 89s52 micro controller.

2.3.3 Burning : 8051 SPI Programmer

we use 8051 SPI Programmer to burn the hex file to the 8051 IC via Arduino .Now the 8051 is ready for hardware implementation.

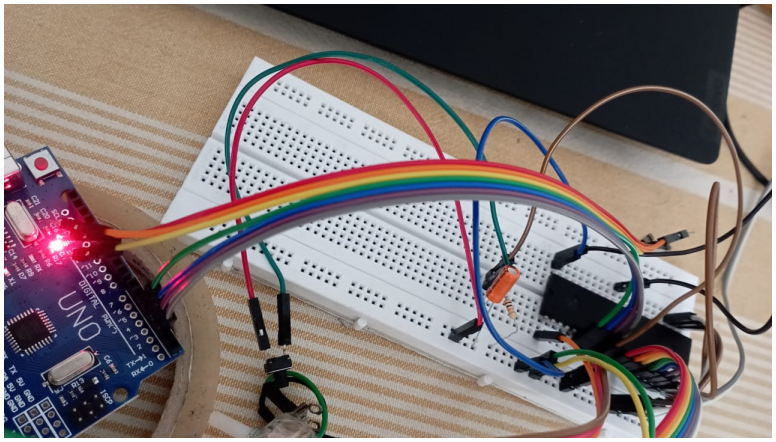


Figure 2.5: Burning via Arduino

2.4 Hardware connections

2.4.1 8051 Essentials

The circuitry for basic working of 8051 micro controller . This include 2x33pF Capacitors,12MHz crystal oscillator,Reset circuit,Source and Ground connections are all provided in the early step.

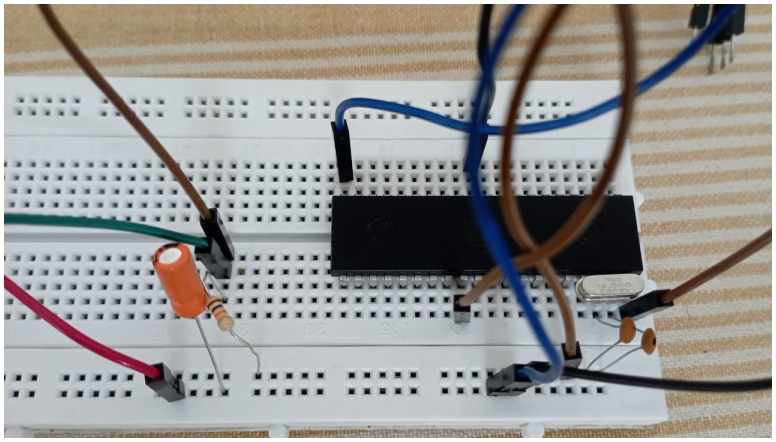


Figure 2.6: Basic connection 8051

2.4.2 Flame sensor

This sensor detects flame otherwise wavelength within the range of 760 nm – 1100 nm from the light source. This sensor can be easily damaged to high temperature. So this sensor can be placed at a certain distance from the flame.

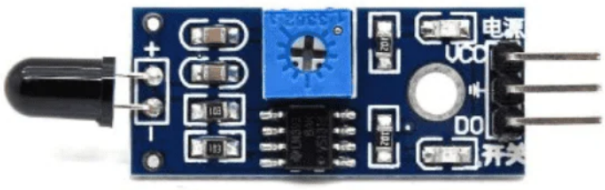


Figure 2.7: Flame sensor

The flame detection can be done from a 100cm distance and the detection angle will be 60 degree. The output of this sensor is an analog signal or digital signal.

2.4.3 LCD Interfacing

The data pins in LCD are connected accordingly to the 8051 mc .This provide for display of information upon outputs from sensor.

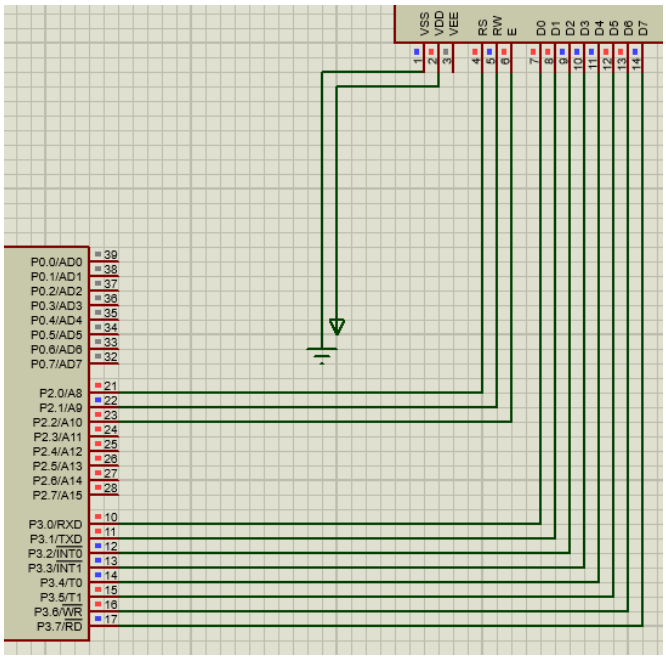


Figure 2.8: LCD interfacing

VDD : Source voltage ; VSS : Ground:D 0-7 : Data pins RS: Register select ; RW

: Read/write ; E: Enable ;

Also the VEE pin need to be connected to a 10K Ohm Potentiometer to control the Contrast of display.The corresponding pins in 8051 are connected to the pins. [3]

2.5 Testing

we provide a suitable test condition in order to observe the working of the prototype.A flame is shown at a distance of 20 cm from the sensor and we can observe the LCD displays ”Flame Detected”.Below shows the testing of project.

The Flame can be kept up to a distance of 80cm and still practically the sensor detects the flame.

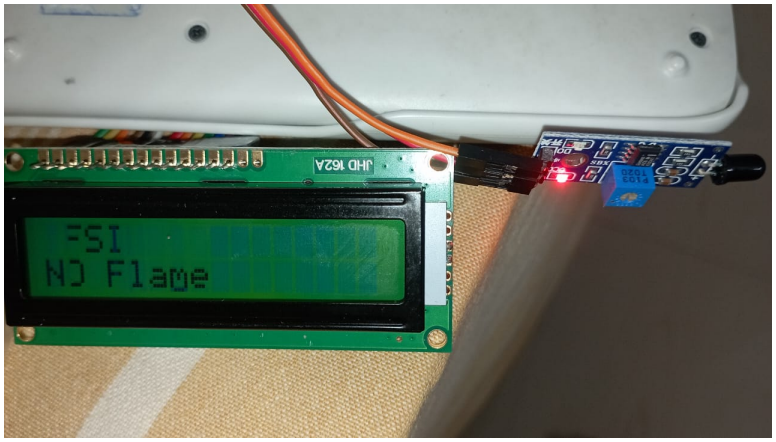


Figure 2.9: NO Flame

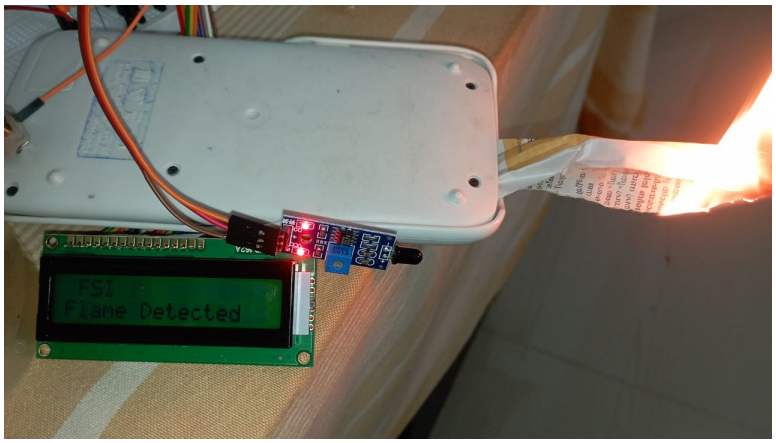


Figure 2.10: Flame Detected

Chapter 3

Conclusion

3.1 Observations

Observed fine working of the Flame sensor with 8051 .

In the absence of flame the LCD show ”NO FLAME.

In presence of flame the LCD shows ”Flame detected”.

Warnings provided in the LCD was accurate and as per the plan.Through out the project was successful .

3.2 Skills acquired

Learned to use the following softwares:

1.Keil uVision

2.Arduino IDE

3.Proteus

4.8051 SPI programmer

Familiarised with hands-on experience on 8051 micro controller,Arduino,LCD and its basic connections;which includes

- usage of Potentiometer.

- need of regulating voltage when supply is provided.

-need of components with compatible ratings.

Proper usage of Internet for gaining knowledge is improved.

3.3 Difficulties faced

Difficulties faced during implementation of project are listed below:

1. Uneasiness in using keil.

which I learned to use from Youtube. [4]

2.Clueless on burning the code to 8051.

Which was also learned from Youtube Video [2] .One of my friends helped me throughout this.

3.Selection of wrong components.

Had to rework on the clock circuit numerous times due to the wrong capacitors and crystals with the wrong frequencies. This was the most hard part.

4.Code correction.

Had to rework the code as sensor was active low.

5.Connecting was a mess.

Hardware connections was a bit hectic as I was not familiar with it.

Chapter 4

Bill of Materials

The total cost incurred for realizing the project is Rupees Three Hundred and Eighty Eight. The bill of materials is given in table 4.1. Except the Flame sensor all other components in the bill of materials were shared with one other student, hence the effective cost for realizing the project came down to Rupees Two Hundred and Thirty Nine.

Sl.No	**Item**	**Manufacturer**	**Price/Unit (Rs.)**	**Quantity**	**Cost (Rs.)**
1	AT89S52	Atmel	75	1	75
2	33 pF Cap	Keltron	1.5	2	3
3	12Mhz crystal	YXC	10	1	10
4	10K Resistor		2	1	2
5	10uF Cap	Keltron	3	1	3
6	Button Switch		1	1	1
7	Flame Sensor	KY	90	1	90
8	10K Pot	SATO	10	1	10
9	LCD1602	JHD China	110	1	110
11	Jumper wires		21	4	84
				Total	**388**

Table 4.1: Bill of Materials

Chapter 5

Code

```
/* Flame sensor interfacing with 8051*/

#include<reg51.h>
#define lcd P3

sbit FLAME=P1^0;

sbit rs=P2^0; //register select
sbit rw=P2^1; //RW
sbit en=P2^2; //enable

void lcd_init();
void cmd(unsigned char);
void dat(unsigned char);
void delay();
void lcd_string(char *s);

void main()
{
    lcd_init();
    lcd_string(" FSI ");
    while(1) {
        if(FLAME) {
            cmd(0xc0); //Force cursor to beginning of 2nd line
            lcd_string("NO Flame");
            delay();
```

```
27          } else {
28              cmd(0xc0); // Force cursor to beginning of 2nd line
29              lcd_string("Flame Detected");
30          }
31      }
32  }
33
34  void lcd_init()
35  {
36      cmd(0x38); // 2 lines and 5 7 matrix (8-bit mode)
37      cmd(0x0e); // Display on, cursor on
38      cmd(0x06); // for entry mode
39      cmd(0x01); // Clear displays
40      cmd(0x80); // Force cursor to beginning of 1st line
41  }
42
43  void cmd(unsigned char a)
44  {
45      lcd=a;
46      rs=0; // command register selected  input considered as commands
47      rw=0; // write command
48      en=1;
49      delay();
50      en=0;
51  }
52
53  void dat(unsigned char b)
54  {
55      lcd=b;
56      rs=1; // data register selected input considered as data
57      rw=0; // write data
58      en=1;
59      delay();
60      en=0;
61  }
62
63  void lcd_string(char *s)
64  {
```

```
65      while(*s) {
66          dat(*s++);
67       }
68 }
69
70 void delay()
71 {
72      unsigned int i;
73      for(i=0;i<20000;i++);
74 }
```

Listing 5.1: Source code of the project

References

[1] @onlineEmbectronicx,https://embetronicx.com/tutorials/microcontrollers/8051/flame-sensor-interfacing-with-8051/Online; accessed 15-April-2023

[2] @onlineJcbro,https://www.jcbrolabs.org/8051-arduinoOnline; accessed 4-May-2023

[3] @onlinelcd-interface,https://www.electronicshub.org/interfacing-16x2-lcd-8051/ Online; accessed 5-May-2023

[4] @onlinekeil,https://www.youtube.com/watch?v=oBvP_LxoqxoOnline; accessed 10-April-2023

[5] @onlinedownload keil,https://www.keil.com/download/

[6] @onlinedownload Arduino IDE,https://www.arduino.cc/en/software/