PROJECT-II REPORT
FIRE ALARM SYSTEM

Fundamental of Automation & Engineering

Institute of Engineering and Technology

JK Lakshmipat University, Jaipur

SUBMITTED TO

Dr. Hanuman Prasad Agarwal

Dr. Devika Kataria

SUBMITTED BY

Bhavishi Bansal(2020BTechCSE019)

Garv Baheti(2020BTechCSE031)

14 July 2021

CERTIFICATE

This is to certify that the project work entitled "**Fire Alarm System**" submitted by **Bhavishi Bansal(2020BTechCSE019), Garv Baheti (2020BTechCSE031)** towards the partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science Engineering of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted.

-------------------------- -------------------------

Dr.H.P. Agrawal
Assistant Professor,
Electrical & Electronics Engineering
Institute of Engineering and technology,
JK Lakshmipat University

Dr. Devika Kataria
Assistant Professor
Electrical & Electronics Engineering
Institute of Engineering and technology,
J.K. Lakshmipat University, Jaipur

ACKNOWLEDGEMENTS

We would like to express my special thanks of gratitude to our Fundamental of Automation course instructor **Dr.H.P.Agrawal** and **Dr.Devika Kataria** , JK Lakshmipat University as well as our Vice chancellor-**Dr.Roshan Lal Raina** and Director IET- **Dr.Sanjay Goel** who gave us the golden opportunity to do this wonderful project which also helped us in doing a lot of Research and we came to know about so many new things We are really thankful to them.

Secondly, we would like to thank our parents who helped me a lot in gathering different information, collecting data and guiding me from time to time in making this project , despite of their busy schedules ,they gave me different ideas in making this project unique.

We would also like to expand our deepest gratitude all those who have directly and indirectly guided us in writing this report. We thank all the people for their help directly and indirectly to complete our report.

Sincerely yours,

Bhavishi Bansal (2020BtechCSE019)

Garv Baheti (2020BtechCSE031)

PROBLEM STATEMENT

To Interface MSP430 along with Flame Sensor and display output with the help of LCD Display, LED, and Buzzer to make Fire Alarm System and to control the 220V supply with the help of Relay.

LIST OF COMPONENTS

- MSP430 Lunchbox with USB Cable
- Code Composer Studio on PC
- Jumper Wires
- LED
- Resistance 1kΩ
- LCD Display
- Fire Sensor Module
- Potentiometer 10kΩ
- Buzzer
- Project Board
- ML1547B Transistor
- Relay
- 220V Bulb

PRODUCT DESCRIPTION

We had designed an automated fire alarm system with the help of MSP430 Lunchbox which can be used to detect fire at any area. This is intended to caution us with the assistance of Buzzer sound, LED and LCD display all together that we will make a move to alert ourselves as well as other people. Also, with the help of Relay we have controlled a 220V Power Supply and made a 220V Bulb Glow, in place of 220V Blub a Fire Extinguisher System can be controlled to Supress Fire.

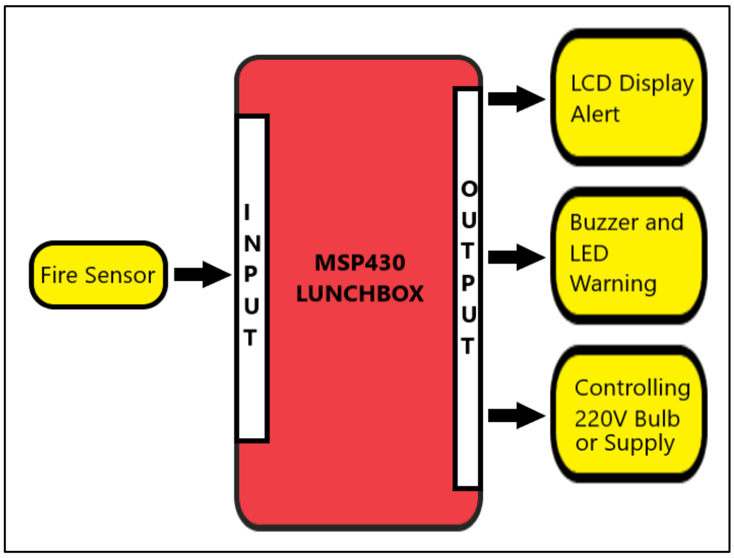
APPLICATIONS

Fire Alarm System can be used in:

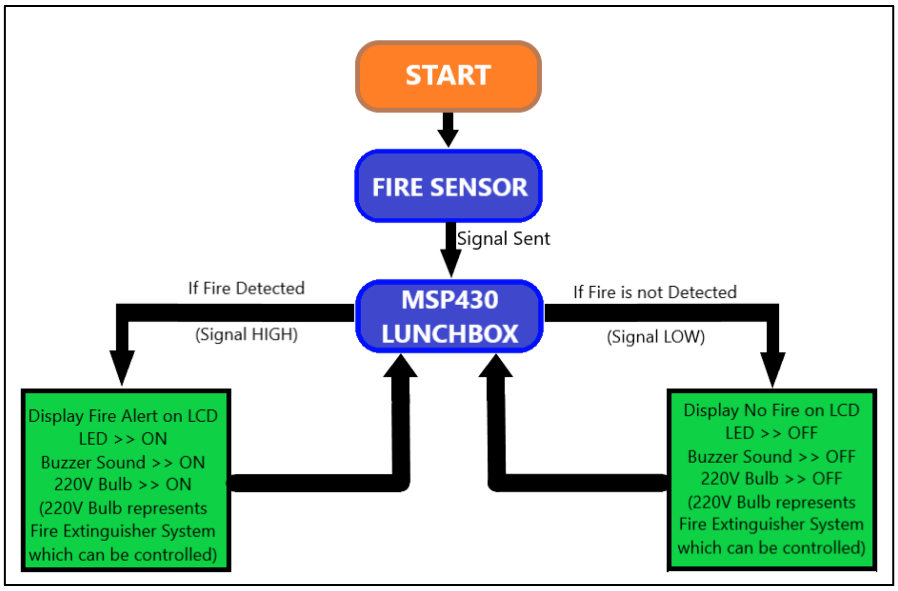
- Home Security System
- School/College Labs
- Industries
- To protect valuable in warehouses
- Museums/Theatres and other public places
- Electrical Departments

METHODOLOGY

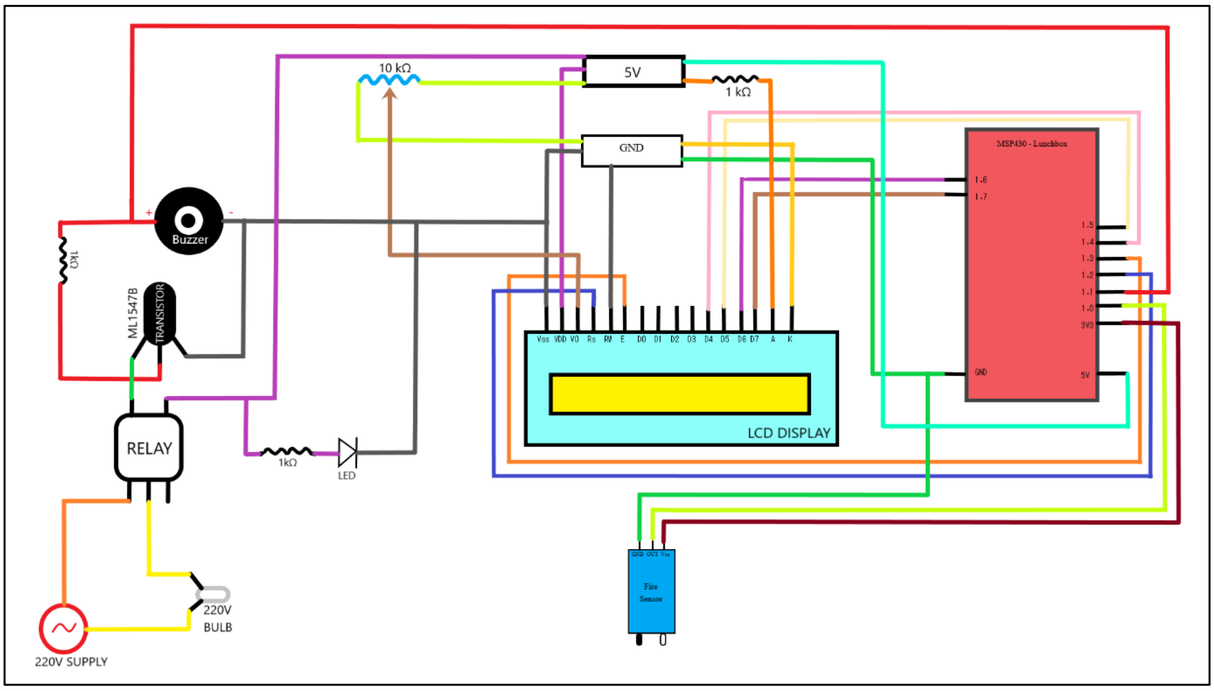
BLOCK DIAGRAM



FLOW CHART



SCHEMATIC DIAGRAM



PROGRAM CODE

Code Composer Studio (C/C++)

```
#include <msp430.h>
#include <inttypes.h>
#include<stdio.h>

#define CMD         0
#define DATA        1

#define LCD_OUT     P1OUT
#define LCD_DIR     P1DIR
#define D4          BIT4
#define D5          BIT5
#define D6          BIT6
#define D7          BIT7
#define RS          BIT2
#define EN          BIT3
#define SW          BIT0
#define LED         BIT1
/**
 *@brief Delay function for producing delay in 0.1 ms increments
 *@param t milliseconds to be delayed
 *@return void
 **/
void delay(uint16_t t)
{
    uint16_t i;
    for(i=t; i > 0; i--)
        __delay_cycles(100);
}

/**
 *@brief Function to pulse EN pin after data is written
 *@return void
 **/
void pulseEN(void)
{
    LCD_OUT |= EN;                              // Giving a falling edge at EN
pin
    delay(1);
    LCD_OUT &= ~EN;
    delay(1);
}

/**
 *@brief Function to write data/command to LCD
 *@param value Value to be written to LED
 *@param mode Mode -> Command or Data
 *@return void
 **/
void lcd_write(uint8_t value, uint8_t mode)
{
    if(mode == CMD)
        LCD_OUT &= ~RS;             // Set RS -> LOW for Command mode
```

```c
    else
        LCD_OUT |= RS;              // Set RS -> HIGH for Data mode

    LCD_OUT = ((LCD_OUT & 0x0F) | (value & 0xF0));              // Write high nibble first
    pulseEN();
    delay(1);

    LCD_OUT = ((LCD_OUT & 0x0F) | ((value << 4) & 0xF0));       // Write low nibble next
    pulseEN();
    delay(1);
}

/**
 *@brief Function to print a string on LCD
 *@param *s pointer to the character to be written.
 *@return void
 **/
void lcd_print(char *s)
{
    while(*s)
    {
        lcd_write(*s, DATA);
        s++;
    }
}

/**
 *@brief Function to move cursor to desired position on LCD
 *@param row Row Cursor of the LCD
 *@param col Column Cursor of the LCD
 *@return void
 **/
void lcd_setCursor(uint8_t row, uint8_t col)
{
    const uint8_t row_offsets[] = { 0x00, 0x40};
    lcd_write(0x80 | (col + row_offsets[row]), CMD);
    delay(1);
}

/***@brief Initialize LCD**/
void lcd_init()
{
    LCD_DIR |= (D4+D5+D6+D7+RS+EN);
    LCD_OUT &= ~(D4+D5+D6+D7+RS+EN);

    delay(150);                     // Wait for power up ( 15ms )
    lcd_write(0x33, CMD);           // Initialization Sequence 1
    delay(50);                      // Wait ( 4.1 ms )
    lcd_write(0x32, CMD);           // Initialization Sequence 2
    delay(1);                       // Wait ( 100 us )

    // All subsequent commands take 40 us to execute, except clear & cursor return (1.64 ms)
    lcd_write(0x28, CMD);           // 4 bit mode, 2 line
    delay(1);

    lcd_write(0x0C, CMD);           // Display ON, Cursor OFF, Blink OFF
    delay(1);

    lcd_write(0x01, CMD);           // Clear screen
    delay(20);

    lcd_write(0x06, CMD);           // Auto Increment Cursor
    delay(1);

    lcd_setCursor(0,0);             // Goto Row 1 Column 1
}

/**
 * @brief
 * These settings are wrt enabling ADC10 on Lunchbox
 **/

void main(void) {
    WDTCTL = WDTPW | WDTHOLD;                   //! Stop Watchdog (Not recommended for code in production and devices working in field)
    lcd_init();
    P1DIR |= LED;                              // Set LED pin -> Output
    P1DIR &= ~SW;                               // Set SW pin -> Input

    while(1)
    {
        if((P1IN & SW))                         // If Output of IR Sensor and P1IN is low
        {
            lcd_setCursor(0,0);                 // Set the cursor in the LCD
            lcd_print("Current Status");        // Printing this command
            lcd_setCursor(1,0);                 // Set the cursor in the LCD
            lcd_print("----NO FIRE---");     // Printing this command
        }
        else{                                   // If Output is high

            P1DIR |= LED;                    // Set as input
            delay(5000);

            lcd_setCursor(0,0);                 // Set the cursor in the LCD
            lcd_print("Current Status");        // Printing this command
            lcd_setCursor(1,0);                 // Set the cursor in the LCD
            lcd_print("!!FIRE ALERT!!");      // Printing this command

        }
        P1DIR &= ~ LED;                         // Set as Input
    }
}
```

RESULTS

In this project with the help of MSP430 Lunchbox we were able to build Fire Alarm System which had several components like LED, Resistance, LCD Display, Fire Sensor Module, Potentiometer, Buzzer, ML1547B Transistor, Relay, 220V Bulb etc.

The Fire Sensor used here is Digital Sensor, when the sensor detects fire it sends HIGH signal to MSP430 Lunchbox and LCD starts displaying !!FIRE ALERT!! on it, buzzer starts producing sound, LED glows up, also with the help of transistor and relay we were able to control the 220V supply that can help us in controlling the Fire Extinguisher System which can be turned on in case of fire to suppress it. Since our MSP430 Lunchbox's PIN 1.1 can only handle 3V so to overcome it transistor was used here so that Relay can work properly. By using this in Fire Alarm System, while alerting us it can also extinguish fire.