Fire Alarm

Idea:

"My idea for the application is a fire alarm."

Description:

Many people lose their lives in fire related accidents daily. Many of these accidents happen due to the not-intime detection of the fire. This device will solve this problem by detecting fire using a smoke sensor and infrared sensor. T

Components:

This fire alarm system is designed to keep you and your surroundings safe in the event of a fire. It incorporates advanced technology, including the "p18f4520" programmable logic controller, as well as an IR sensor and a smoke sensor. These components work together to detect the presence of smoke or flames accurately.

When a fire is detected, the system springs into action. It activates a combination of a buzzer and lights to alert you to the danger. Additionally, it has a special pin that sends a signal to the emergency response team, ensuring that help is on the way as soon as possible.

To make all of this happen, we start by setting up the pins of the system, designating which ones will be used for input and output. This allows us to connect the sensors and control the alarm components effectively.

We also declare two important variables *smokeDetected* and *fireDetected.* These variables keep track of whether smoke or flames have been detected, respectively.

Once everything is set up, we enter the main loop of the program. Here, we establish the initial conditions for the buzzer, lights, and emergency response signal. By continuously monitoring the readings from both the smoke sensor and the IR sensor, we create a logical condition using an "OR" statement. If either sensor detects smoke or visible flames, a signal is sent to the programmable logic controller.

Upon receiving this signal, the PLC triggers the activation of the alarm, buzzer, and lights. Simultaneously, it sends a signal through the dedicated pin to the emergency response system, notifying them of the fire. This way, the alarm rings loudly, alerting everyone in the vicinity, while the emergency response team is promptly informed to take necessary action.

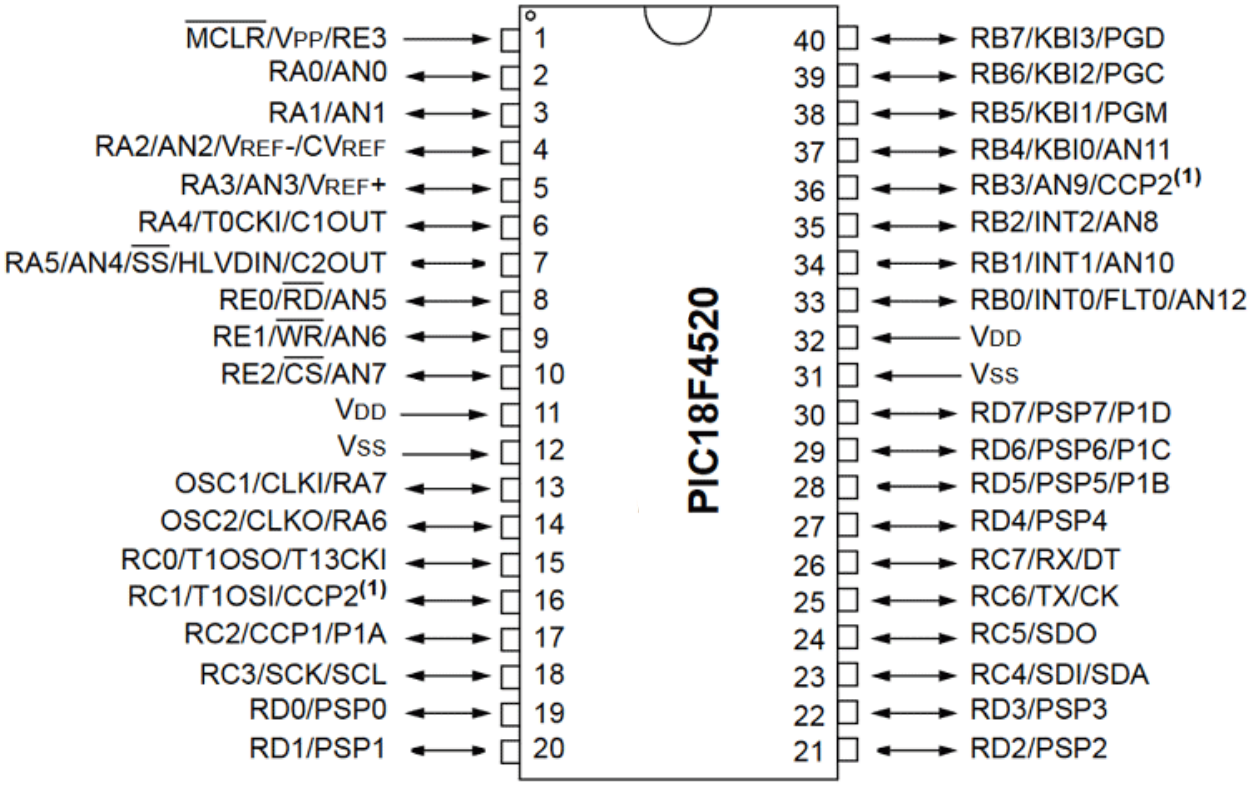
With its advanced features and careful integration of various components, this fire alarm system offers reliable and efficient fire detection capabilities. It ensures your safety and provides peace of mind, knowing that you are protected by a sophisticated and responsive fire safety solution.

The flame sensor works by detecting infrared radiation which is emitted by any visible flame and smoke sensor works by detecting concentrations of different gases in the atmosphere.

The combination of both will make systems much more effective and fire will be detected as early as it is ignited.

Programable Logic Controller 18F4520

The PLC looks like



MCLR (Master Clear): This pin is responsible for external reset and is activated when pulled low. It allows you to reset the microcontroller and restore it to its default state.

RA0 (AN0): This pin serves as an input for reading analog voltages. It can also be used as a digital input or output for connecting various devices.

RA1 (AN1): Like RA0, this pin is an input for reading analog voltages or a digital input/output pin for general-purpose use.

RA2 (AN2/VREF-): This pin can be used for analog input, allowing you to measure voltages accurately. It can also serve as the negative voltage reference input for the analog-to-digital converter (ADC).

RA3 (AN3/VREF+): Like RA2, this pin is an analog input and can be used for precise voltage measurements. It can also function as the positive voltage reference input for the ADC.

RA4/T0CKI: This versatile pin can be used for multiple purposes, such as an external clock input or a digital input/output pin.

RA5/SS/HLVDIN: This multipurpose pin can be used as the Slave Select input for SPI communication or as a digital input/output pin. It can also serve as the input for the High/Low Voltage Detect module.

RE0/RD/AN5, RE1/WR/AN6, RE2/CS/AN7: These pins are multipurpose and can be configured as digital input/output pins or analog inputs. They also have additional functions related to memory interfaces.

VDD and VSS: These pins are the power supply pins. VDD is connected to the positive power source, while VSS is connected to ground.

RC0/T1OSO/T1CKI, RC1/T1OSI, RC2/CCP1: These pins have various functions, including digital input/output and association with specific modules like Timer1 oscillator and Capture/Compare/PWM.

RC3/SCK/SCL, RC4/SDI/SDA, RC5/SDO: These pins are involved in communication protocols like SPI and I2C. They can serve as digital input/output pins or as inputs for specific communication signals.

RC6/TX/CK, RC7/RX/DT: These pins are associated with USART communication and can function as digital input/output pins or transmit/receive lines.

VREF- and VREF+: These pins are used as reference voltage inputs for the ADC, ensuring accurate analog voltage measurements.

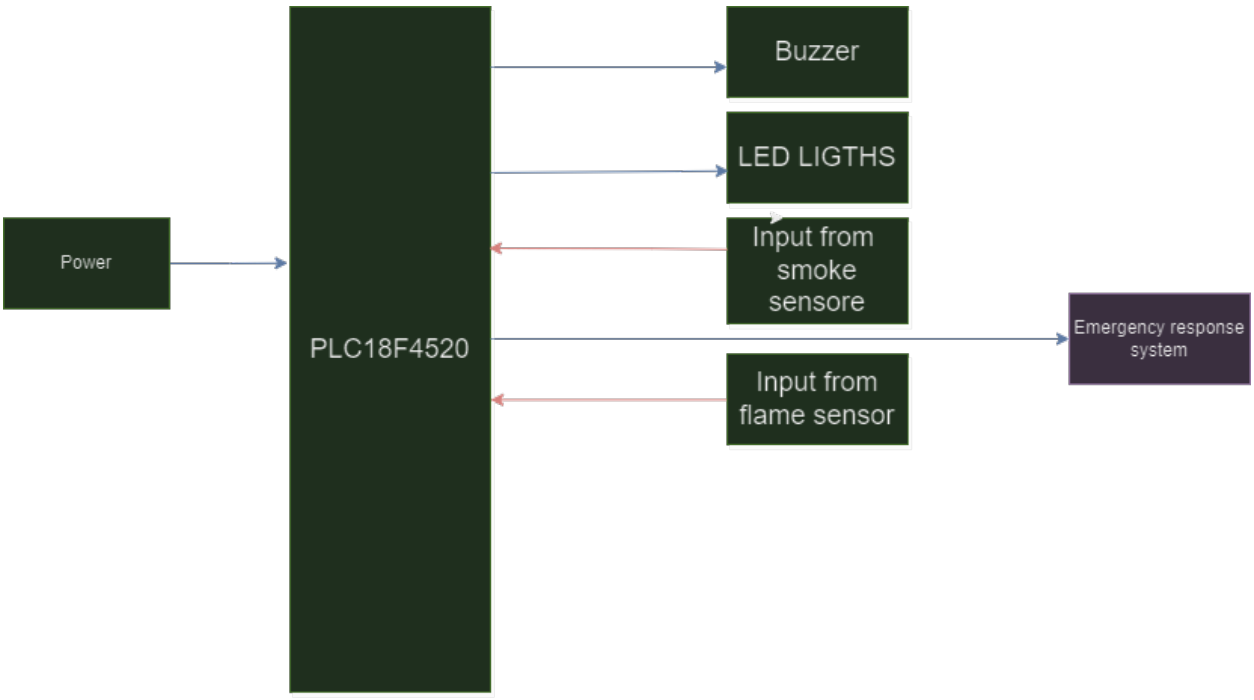
Here is our Schematic for the Fire Alarm

Pin 33 (RB0/INT): This pin can be set up as a general-purpose input or an external interrupt input. It can be connected to smoke sensor in the fire alarm system,

Pin 34 (RB1) is a general-purpose input pin that can be used to connect to additional input devices, such as IR sensor in Fire Alarm.

Pin 35 (RB2): This pin is a general-purpose output pin and can be connected to control an output device, such as a buzzer. When activated, it produces an audible alarm to alert occupants of the fire.

Pin 33 (RB0/INT): This pin can be set up as a general-purpose input or an external interrupt input. It can be connected to smoke sensor in the fire alarm system,



Pin 34 (RD1) is a general-purpose input pin that can be used to connect to additional input devices, such as IR sensor in Fire Alarm.

Pin 35 (RD2): This pin is a general-purpose output pin and can be connected to control an output device, such as a buzzer. When activated, it produces an audible alarm to alert occupants of the fire.

Pin 36 (RB3/PGM): It is used for in-circuit debugging and programming. In the fire alarm system, it may not be directly utilized for output purposes.

Pin 37 (RB4): Another general-purpose output pin that can be connected to control a visual indicator, such as an LED or a set of lights. It illuminates to visually indicate the presence of a fire.

Pin 6 (RB5): It can be used as a general-purpose output pin for controlling emergency response.

The controller will manipulate the data given by the sensor in the way it is programmed.

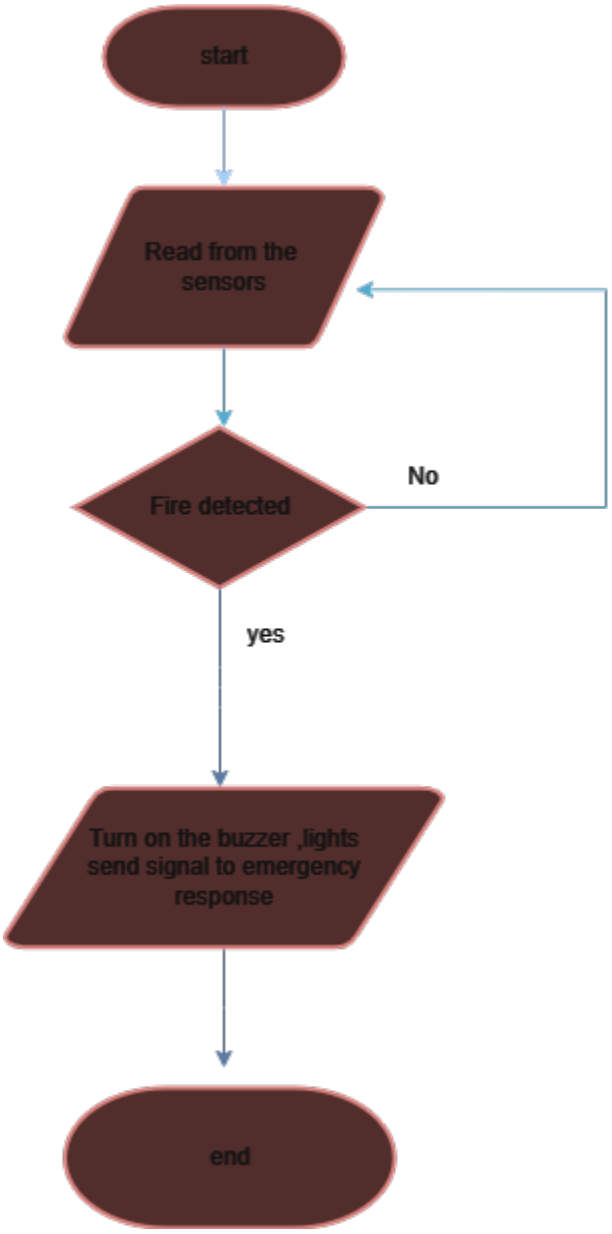
The input pins will take data from the sensors they are inputting because the controller will read from this pin and other pin which are termed output are because the controller will write on these pins.

Now the thing which will put soul in our controller is the program the logic of which is being explained in the flowchart.

Flowchart:

The flowchart of the program is given by:

This represents how the program instructions will be execcuted.



Explanation

Start: The program begins its execution

Read Sensor Inputs: The program continuously reads the status of the smoke sensor and IR sensor.

Check for Fire: An OR gate condition is implemented to check if either the smoke sensor or the IR sensor detects any signs of fire.

Fire Detected: If fire is detected, the program proceeds to activate the alarm system.

Activate Buzzer and Lights: The output pins controlling the buzzer and lights are turned on to indicate the presence of fire and send signal to the emergency response.

Not detected: Return to the read part.

End: The program execution ends.

Description:

The program begins by setting up the microcontroller and configuring it accordingly. It prepares the necessary connections for the buzzer, lights, smoke sensor, and IR sensor.

Once everything is set up, the program enters a loop that constantly checks the status of the smoke sensor and IR sensor. It keeps an eye out for any indications of a fire by looking for signals from either sensor.

If the program detects a fire, it activates the alarm system. This means that the buzzer and lights are turned on to alert everyone nearby about the fire. It also sends a signal to the emergency response system so that appropriate actions can be taken.

The program continues to monitor the sensor inputs, repeating the process of checking for fire signs. As long as no fire is detected, it remains in the loop, ensuring continuous surveillance.

Now moving towards the real code we are going to fetch:

Code

```
#include <xc.h>

// Pin Definitions

#define SMOKE_SENSOR_PIN RD0

#define IR_SENSOR_PIN RD1

#define BUZZER_PIN RB2

#define LIGHTS_PIN RB3

#define EMERGENCY_PIN RB4

// Function Declarations

void initialize();
```

```
void activateAlarm();
void deactivateAlarm();

// Main Function
int main()
{
   initialize();

   while (1)
   {
      // Check for fire conditions
      if (PORTBbits.RB0 || PORTBbits.RB1) //  “OR” gate
      {
         activateAlarm();
      }
      else
      {
         deactivateAlarm();
      }
   }

   return 0;
}

// Function to initialize the microcontroller and pins
void initialize()
{
   TRISBbits.TRISD1 = 1; // Set RB0 as input pin for smoke sensor
   TRISBbits.TRISD2 = 1; // Set RB1 as input pin for IR sensor
```

```
    TRISBbits.TRISB2 = 0; // Set RB2 as output pin for buzzer
    TRISBbits.TRISB3 = 0; // Set RB3 as output pin for lights
    TRISBbits.TRISB4 = 0; // Set RB4 as output pin for emergency response

    PORTB = 0x00; // Set all output pins initial state as low
}

// Function to activate the alarm system
void activateAlarm()
{
    LATBbits.LATB2 = 1; // Turn on the buzzer
    LATBbits.LATB3 = 1; // Turn on the lights
    LATBbits.LATB4 = 1; // Set emergency response pin to high
}

// Function to deactivate the alarm system
void deactivateAlarm()
{
    LATBbits.LATB2 = 0; // Turn off the buzzer
    LATBbits.LATB3 = 0; // Turn off the lights
    LATBbits.LATB4 = 0; // Set emergency response pin to low
}
```

This is the code for the application.

The code has following functions,

Initializer: In this function we are configuring the pins of the controller.

Main function.: This is the function which is calling all the function we have featured an infinite loop in it.

Activate Alarm: The function which is activating the alarm.

Deactivate Alarm: Function keeping the alarm deactivated.

The xc.h is essential header for PLC coding.

Conclusion:

Our fire alarm will be ready by compiling the code through compiler MPLAB IDE can be used for it and fetching the code to PLC through PIC genios board.