# 🔥 Fire Alarm Detection using Logistic Regression 🚒

**1. Project Overview :**

This project involves the development of a fire alarm system using a logistic regression model. The model calculates the probability of a fire based on temperature sensor data and an alarm is triggered if the probability exceeds a certain threshold.

**2. Hardware Components**

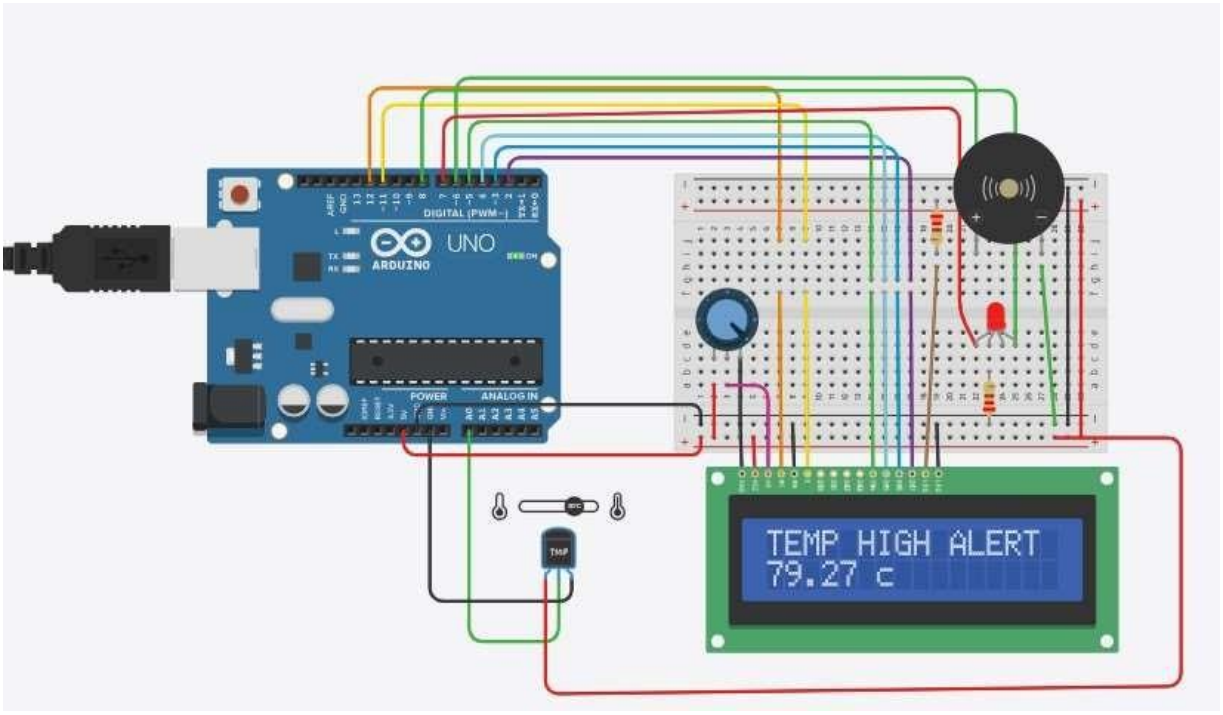
| Name | Quantity | Component |
|---|---|---|
| U3 | 1 | Temperature Sensor [TMP36] |
| U7 | 1 | Arduino Uno R3 |
| Rpot2 | 1 | 300 kΩ Potentiometer |
| R2 | 1 | 220 Ω Resistor |
| R1 | 1 | 220 Ω Resistor |
| PIEZO3 | 1 | Piezo |
| D2 | 1 | LED RGB |
| U8 | 1 | LCD 16 x 2 |

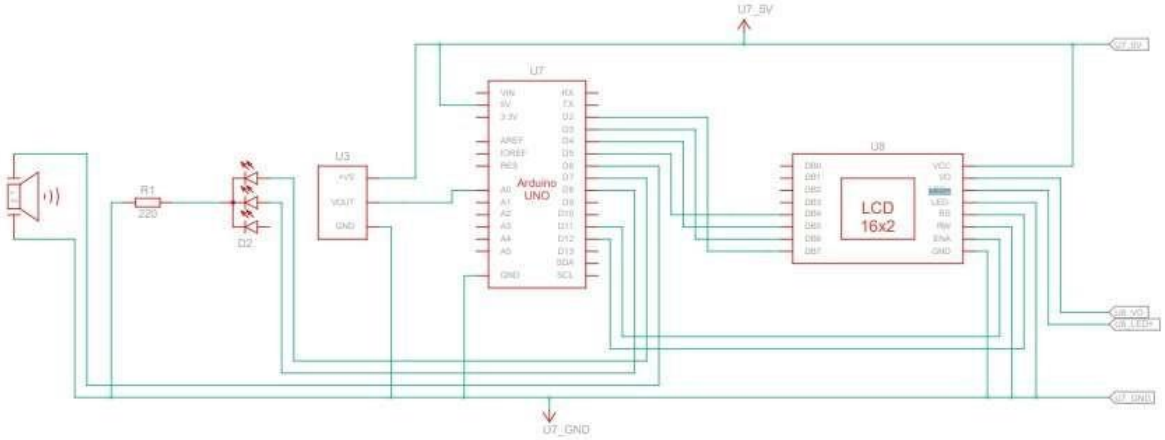
**3. Software Components**

(i) Arduino IDE

(ii) Python (for logistic regression model training)

## 4. System Architecture




**5. Data Flow and Processing**

This Arduino code implements a fire alarm system that predicts the likelihood of a fire based on temperature readings using a logistic regression model.

1. Libraries and Variables:
   - The code includes the LiquidCrystal library for LCD control.
   - Variables are declared for temperature, LED pins (red and green), buzzer pin, and an instance of the LiquidCrystal class for the LCD.

2. Machine Learning Model Class (`ML_Model`):
   - The `ML_Model` class includes a placeholder `load` function (not implemented) for loading a model file and a `predict` function for making predictions using logistic regression.

3. Setup Function:
   - Initializes the LCD display and sets pin modes for LEDs and the buzzer.

4. Loop Function:
   - Reads a simulated temperature value from an analog pin.
   - Make a prediction using the logistic regression model.
   - Converts the analog reading to temperature in Celsius and displays it on the LCD.
   - Checks the prediction and triggers actions:
   - If the prediction is 1 (indicating high temperature), activates the red LED, displays an alert on the LCD, and sounds the buzzer with alternating tones.
   - If the prediction is 0 (indicating low temperature), activate the green LED and display a message on the LCD.

## 6. Code Snippet (Arduino)

Text

```
#include <LiquidCrystal.h>
float temperature;
int led_red=7;
int led_green=8;
int buzzer=6;
int sensorval;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
class ML_Model {
public:
  float predict(float* features) {
    float temp = features[0];
    // Logistic regression model parameters (you need to adjust these based on your trained model)
    float intercept =  0.9701847660091361;
    float coef_temperature = 0.05901801;

    // Calculate the linear combination
    float linear_combination = intercept + coef_temperature * temp ;

    // Apply the logistic function to get the probability
    float probability = 1.0 / (1.0 + exp(-linear_combination));

    // Return the probability
    return probability;
   }
};

ML_Model model;
void setup() {
  lcd.begin(16, 2);
  pinMode(led_red,OUTPUT);
  pinMode(led_green,OUTPUT);
  pinMode(buzzer,OUTPUT);

  Serial.begin(9600);
}

void loop() {
  temperature=analogRead(A0);
  temperature=temperature*1.34;
  lcd.setCursor(0, 1);
  lcd.setCursor(6, 1);
  float features[] = {temperature};
  int alarmPrediction = model.predict(features);
  if(alarmPrediction==1){
    digitalWrite(led_red,HIGH);
    digitalWrite(buzzer,HIGH);
     lcd.setCursor(0,0);
    lcd.print("TEMP HIGH ALERT");
    tone(buzzer,800,200);
    delay(200);
    tone(buzzer,600,200);
  }
 else{
    digitalWrite(led_green,HIGH);
    digitalWrite(led_red,LOW);
    lcd.setCursor(0,0);
    lcd.print("TEMP LOW ALert");
  }
  Serial.print(" Fire Alarm Prediction: ");
  Serial.println(alarmPrediction);
}
```

## 7. Machine Learning Model ( Link : Jupyter notebook)

In a machine learning model with temperature as an independent feature and fire_alarm as the dependent feature, a logistic regression model is used. The model calculates the probability of a fire_alarm being triggered based on temperature data. Training the model involves adjusting parameters, such as the intercept and temperature coe cient, to fit historical data. During prediction, the model evaluates the temperature input and outputs a probability, which can be used to determine if the fire_alarm should be activated.

## 8. Conclusion

This project demonstrates the integration of a logistic regression model into a fire alarm system using Arduino. The model calculates the probability of fire based on sensor data, providing a more nuanced approach to fire detection. Adjust the threshold and model parameters as needed for optimal performance. Continuous monitoring and testing are recommended to ensure the reliability of the system.