**PROJECT DEVELOPMENT PHASE**

**DELIVERY OF SPRINT-3**

|  |  |
| --- | --- |
| Date | 15 November 2022 |
| Team ID | PNT2022TMID40719 |
| Project Name | Industry Specific Intelligence Fire Management System |

#include <WiFi.h>

#include <Wire.h>

#include <SPI.h>

#include "ThingSpeak.h"

#include

<WiFiClient.h>

unsigned long myChannelNumber = 2; const char \* myWriteAPIKey = "25V40ZAPI6KIZFGY"; int LED\_PIN = 32;

// the current reading from the input pin int

BUZZER\_PIN= 12;

const int mq2 = 4; int

value = 0;

//Flame int flame\_sensor\_pin = 10 ;

// initializing pin 10 as the sensor digital output pin int flame\_pin = HIGH ; // current state of sensor

char ssid[] = "VIKRAM"; char pass[] = "VIKRAM"; WiFiClient client;

#define PIN\_LM35 39

#define ADC\_VREF\_mV 3300.0

#define ADC\_RESOLUTION 4096.0

#define RELAY\_PIN 17

#define RELAY\_PIN1 27

void setup()

{

Serial.begin(115200);

pinMode(RELAY\_PIN, OUTPUT);

pinMode(RELAY\_PIN1, OUTPUT);

Serial.print("Connecting to "); Serial.println(ssid); WiFi.begin(ssid, pass);

int wifi\_ctr = 0;

while (WiFi.status() != WL\_CONNECTED)

{

delay(1000); Serial.print(".");

}

Serial.println("WiFi connected");

ThingSpeak.begin(client); pinMode(LED\_PIN, OUTPUT); pinMode(mq2, INPUT); pinMode ( flame\_sensor\_pin , INPUT );

// declaring sensor pin as input pin for Arduino pinMode(BUZZER\_PIN, OUTPUT);

}

void temperature()

{

int adcVal = analogRead(PIN\_LM35); float milliVolt = adcVal \*

(ADC\_VREF\_mV / ADC\_RESOLUTION);

float tempC = milliVolt / 10;

Serial.print("Temperature: ");

Serial.print(tempC); Serial.print("°C"); if(tempC > 60)

{

Serial.println("Alert");

digitalWrite(BUZZER\_PIN, HIGH);

// turn on

} else

{

digitalWrite(BUZZER\_PIN, LOW);

// turn on

}

int x = ThingSpeak.writeField(myChannelNumber,1, tempC, myWriteAPIKey);

}

void GasSensors()

{

//mq2

int gassensorAnalogmq2 = analogRead(mq2);

Serial.print("mq2 Gas Sensor: ");

Serial.print(gassensorAnalogmq2);

Serial.print("\t");

Serial.print("\t");

Serial.print("\t");

if (gassensorAnalogmq2 > 1500)

{

Serial.println("mq2Gas"); Serial.println("Alert"); digitalWrite(RELAY\_PIN1, HIGH);

// turn on fan 10 seconds delay(100);

} else

{

Serial.println("No mq2Gas"); digitalWrite(RELAY\_PIN1,

LOW);

// turn off fan 10 seconds delay(100);

}

Int a = ThingSpeak.writeField(myChannelNumber,4,gassensorAnalogmq2, myWriteAPIKey);

}

void flamesensor()

{ flame\_pin = digitalRead ( flame\_sensor\_pin ) ;

// reading from the sensor if

(flame\_pin == LOW )

// applying condition

{

Serial.println ( " ALERT: FLAME IS DETECTED" ) ; digitalWrite (BUZZER\_PIN, HIGH ) ;

// if state is high, then turn high the BUZZER

} else

{

Serial.println ( " NO FLAME DETECTED " ) ;

digitalWrite (BUZZER\_PIN , LOW ) ;

// otherwise turn it low

} int value = digitalRead(flame\_sensor\_pin);

// read the analog value from sensor if

(value ==LOW)

{

Serial.print("FLAME");

digitalWrite(RELAY\_PIN, HIGH);

} else

{

Serial.print("NO FLAME");

digitalWrite(RELAY\_PIN, LOW);

}

} void loop()

{

temperature(); GasSensors(); flamesensor();

}