Sprint-4

Team ID	PNT2022TMID50107
Project Name	Industry-specific intelligent fire management system

DELIVERY OF SPRINT-4

```
#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[] = "RIYAZ";

char pass[] = "RIYAZ";

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();
}
```