FINAL DELIVERABLES

TEAM ID : PNT2022TMID41407

SUBMITTED BY:

MEENA. S
GAYATHRI.S
MOULISHA .R
RITHIKA .B

```
#include <dht.h>


#define dht_apin A0                                    //
Analog Pin 0 is connected to DHT sensor
#define mqt_apin A1                                     //
Analog Pin 1 is connected to MQT 135
sensor dht DHT; int sensorValue;


void setup(){


  Serial.begin(9600);
//Serial port to communicate with Python
code
  Serial1.begin(9600);
//Serial port to communicate with Wearable
device through Bluetooth (HC-05)
```

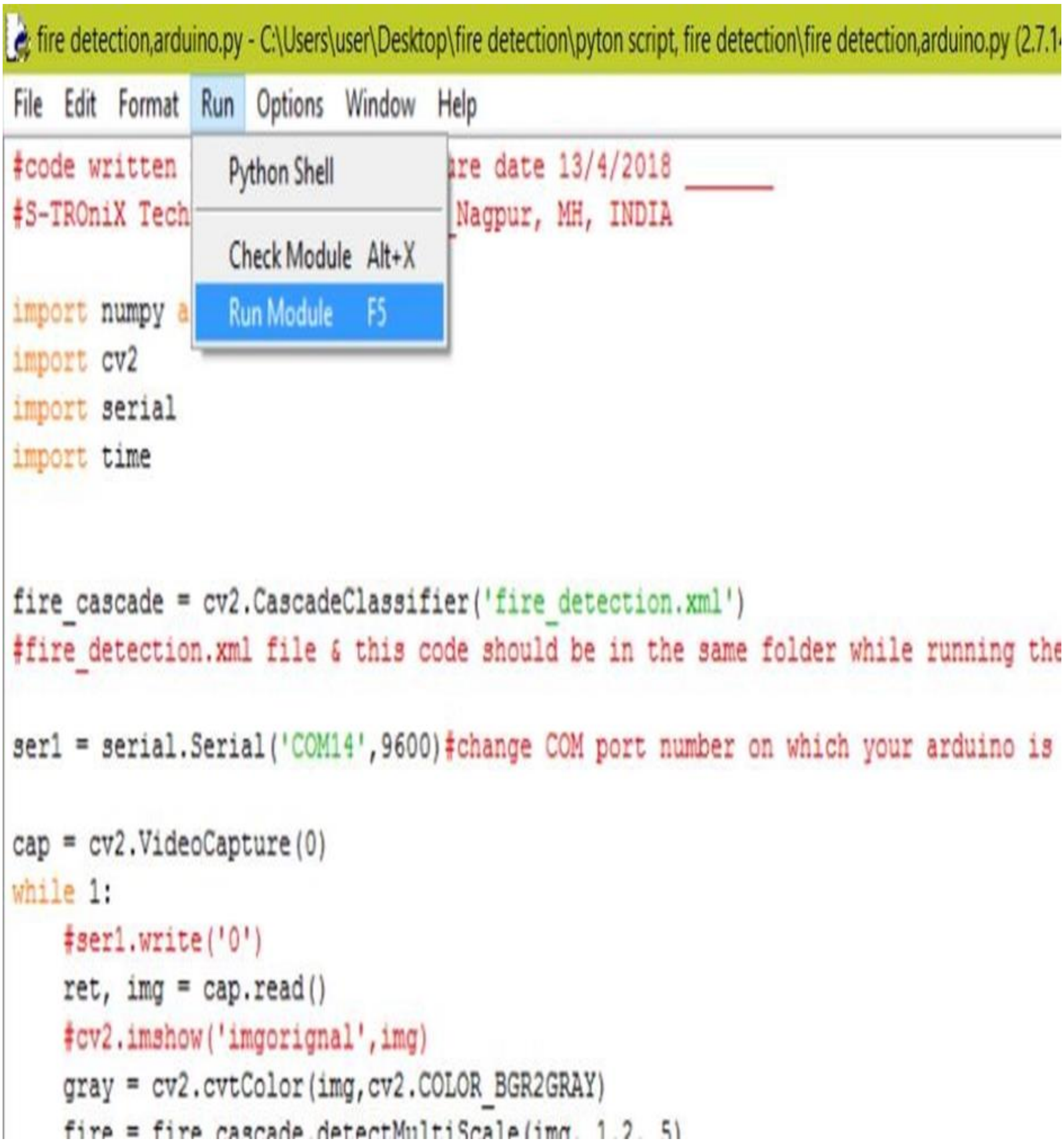
```
  delay(500);                              //Delay to
let system boot  }
void loop(){


   DHT.read11(dht_apin);                              //
read analog input pin 0(DHT11)    sensorValue
= analogRead(mqt_apin);                   // read
analog input pin 1(MQ135)


   //Send Humidity status to Python Code
   Serial.print("Current humidity = ");
   Serial.print(DHT.humidity);
   Serial.print("%  ");
   //Send Temperature status to Python Code
   Serial.print("temperature = ");
   Serial.print(DHT.temperature);
   Serial.println("C  ");


   //Send AirQuality sensor value to Python
code
   Serial.print("AirQua=");
   Serial.print(sensorValue, DEC);
   Serial.println(" PPM");
   //Send signals to the Wearable
   Serial1.println("H T A");
   Serial1.println(DHT.humidity);
   Serial1.println(DHT.temperature);
   Serial1.println(sensorValue, DEC);
    delay(100);                                   // wait
100 milliseconds for next reading  }
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
WiFiClient wifiClient;
String data3;
#define DHTTYPE DHT11
#define DHTPIN 9
DHT dht(DHTPIN, DHTTYPE);

#define ORG "v6wg8x"
#define DEVICE_TYPE "nodeMcu"
#define DEVICE_ID "NodeMCU"
#define TOKEN "123456789"
#define speed 0.034

void callback(char* topic, byte* playload,
unsigned int payloadLength);
```

```
char server[] = ORG
".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-
2/evt/Data/fmt/json"; char topic[] = "iot-
2/cmd/test/fmt/String"; char authMethod[] =
"use-tokenauth"; char token[] = TOKEN; char
clientId[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID; PubSubClient client(server, 1883,
callback , wifiClient); void publishData();

String command;



String data = "";


long duration; float
dist;




void setup()
{
  Serial.begin(115200);   dht.begin();
wifiConnect();   mqttConnect();
}
```

```
void loop() {   publishData();   delay(500);
  if (!client.loop()) {
mqttConnect();
   }
 }

void wifiConnect() {
  Serial.print("Connecting to ");
Serial.print("Wifi");
WiFi.begin("SSID","Passord");   while
(WiFi.status()
!= WL_CONNECTED) {
    delay(500);
    Serial.print(".");
   }
  Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
 }
void mqttConnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting MQTT client to ");
Serial.println(server);     while
(!client.connect(clientId, authMethod, token))
 {
```

```
Serial.print(".");      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void initManagedDevice() {
  if (client.subscribe(topic)) {
    Serial.println("IBM subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void publishData()
{
  int sensorValue = analogRead(34); //MQT 135
connected to GPIO 34 (Analog ADC1_CH6)
  Serial.print("AirQua=");
  Serial.print(sensorValue, DEC);
Serial.println(" PPM");   float humid =
dht.readHumidity();   float temp =
dht.readTemperature(true);   float airQty =
sensorValue/4095;   String payload =
"{\"Temperature\":";   payload += temp;
payload += "}";    if
(client.publish(publishTopic, (char*)
payload.c_str())) {
    Serial.println("Publish OK");
  }
  payload = "{\"Air Quality\":";   payload +=
airQty;   payload += "%}";   if
(client.publish(publishTopic, (char*)
payload.c_str())) {
    Serial.println("Publish OK");
  }
}
void callback(char* subscribeTopic, byte*
payload, unsigned int payloadLength) {
  Serial.print("callback invoked for topic:");
Serial.println(subscribeTopic);   for (int i =
0; i < payloadLength; i++) {     dist +=
(char)payload[i];
  }
  Serial.println("data:" + data3);   if (data3 ==
"lighton") {
    Serial.println(data3);
  }
```

```
 data3 = "";
}
```


fire detection,arduino.py - C:\Users\user\Desktop\fire detection\pyton script, fire detection\fire detection,arduino.py (2.7.1
File Edit Format Run Options Window Help
Python Shell
Check Module Alt+X
Run Module F5

```
#code written                    re date 13/4/2018 ______
#S-TROniX Tech                   Nagpur, MH, INDIA

import numpy a
import cv2
import serial
import time


fire_cascade = cv2.CascadeClassifier('fire_detection.xml')
#fire_detection.xml file & this code should be in the same folder while running the

ser1 = serial.Serial('COM14',9600)#change COM port number on which your arduino is

cap = cv2.VideoCapture(0)
while 1:
    #ser1.write('0')
    ret, img = cap.read()
    #cv2.imshow('imgorignal',img)
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    fire = fire_cascade.detectMultiScale(img, 1.2, 5)
```

COM3 (Arduino/Genuino Mega or Mega 2560)
Send
Autoscroll