ДОДАТОК А
ЛІСТИНГ ПРОГРАМИ КЕРУВАННЯ

```
// DHTLibrary
#include "DHT.h"

//FirebaseLibrary
#include <FirebaseESP8266.h>
#include <ESP8266WiFi.h>

//CO2 Library
#include "MHZ19.h"
#include <SoftwareSerial.h>

// Defines the pin and type of DHT sensor
#define DHTPIN D1
#define DHTTYPE DHT22
DHT dht22(DHTPIN, DHTTYPE);

//Defines the pin CO2 and other settings
#define RX_PIN D4
#define TX_PIN D3
#define BAUDRATE 9600
MHZ19 myMHZ19;
SoftwareSerial mySerial(RX_PIN, TX_PIN);

// Data for Firebase
#define FIREBASE_HOST "fir-meteo-*****-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "jgit5zgoshBS*******etbyEp4FwAOrkPBh3XvII"

//Name Wifi
#define WIFI_SSID "TP-Link_6**E"
#define WIFI_PASSWORD "733***"

// Declare data objects from FirebaseESP8266
FirebaseData firebaseData;

void setup() {
  Serial.begin(115200);

  dht22.begin();

  mySerial.begin(BAUDRATE);
  myMHZ19.begin(mySerial);                    // *Serial(Stream) refence must be passed
to library begin().
  myMHZ19.autoCalibration(false);                    // Turn auto calibration ON (OFF
autoCalibration(false))

  // Connecting to Wifi
```

```
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {

  // The DHT11 sensor reads temperature and humidity
  float t = dht22.readTemperature();
  float h = dht22.readHumidity();

  int CO2;
  CO2 = myMHZ19.getCO2();

  // Check whether the sensor is successful in reading temperature and humidity
  if (isnan(t) || isnan(h)) {
    Serial.println("DHT11 doesn't read");
    delay(1000);
    return;
  }

  // Displays temperature and humidity on the serial monitor
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.println(" *C");
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.println(" %");
  Serial.print("CO2 (ppm): ");
  Serial.println(CO2);

  // Provides temperature and humidity status to firebase
  if (Firebase.setFloat(firebaseData, "/MeteoAssistant/temperature", t)){
    } else{
      Serial.println("Temperature doesn't upload");
      Serial.println("Because: " + firebaseData.errorReason());
    }
```

```
  if (Firebase.setFloat(firebaseData, "/MeteoAssistant/humidity", h)){
    } else{
      Serial.println("Humidity doesn't upload");
      Serial.println("Because: " + firebaseData.errorReason());
    }

  if (Firebase.setFloat(firebaseData, "/MeteoAssistant/CO2", CO2)){
    } else{
      Serial.println("Level of CO2 doesn't upload");
      Serial.println("Because: " + firebaseData.errorReason());
    }

  delay(1000);
}
```

ДОДАТОК Б

ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ

Формат	Позиція	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кільк.	Примітки
			Текстові документи		
А4	1	ГЮІК. 442351.014 ПЗ	Пояснювальна записка	74	с.
			Інші документи		
	2		Демонстраційний матеріал у	15	с.
			вигляді презентації		

Змін.	Арк.	Номер докум.	Підпис	Дата	ГЮІК. 442351.014 ПЗ			
Розроб.		Халімонов			Розробка інтелектуального модулю для автоматизованого визначення умов життєдіяльності у житлових та робочих приміщеннях *Відомість кваліфікаційної роботи*	Літера	Аркуш	Аркушів
Перевір.		Сотник				Н	1	1
						Кафедра КІТАР ХНУРЕ		
Н.контр.		Бронніков						
Затв.		Невлюдов						