GOVERNMENT COLLEGE OF ENGINEERING, ERODE

INTERNET OF THINGS

ENVIRONMENTAL MONITORING

PHASE 4

TEAM MEMBERS

BRINTHA SHREE S S [22CSE56L]

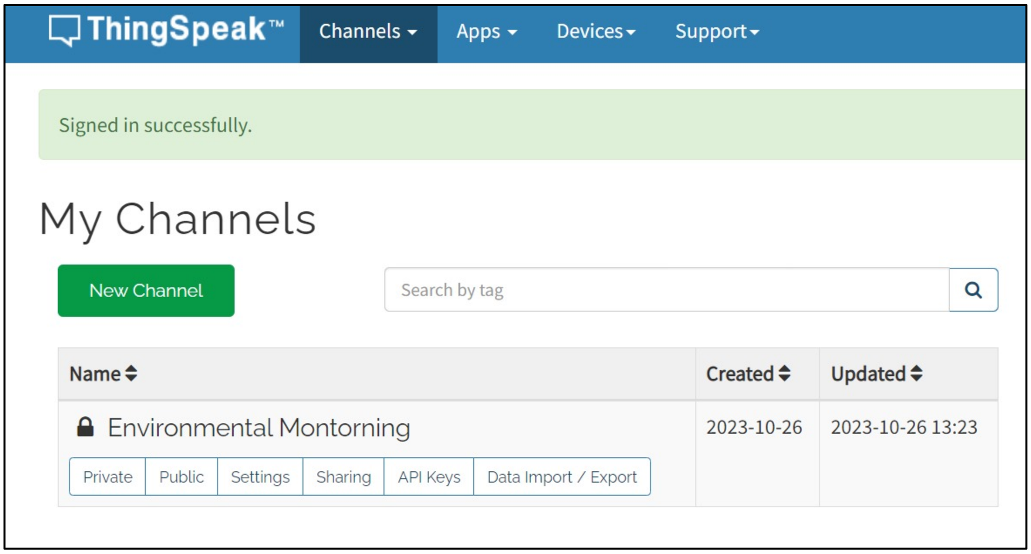
KALPANA CHAWLA M [21CSE17]

KAVIPRIYA P [21CSE19]

YOGALAKSHMI S [21CSE54]

Design the platform to receive and display real-time temperature and humidity data from IoT devices.

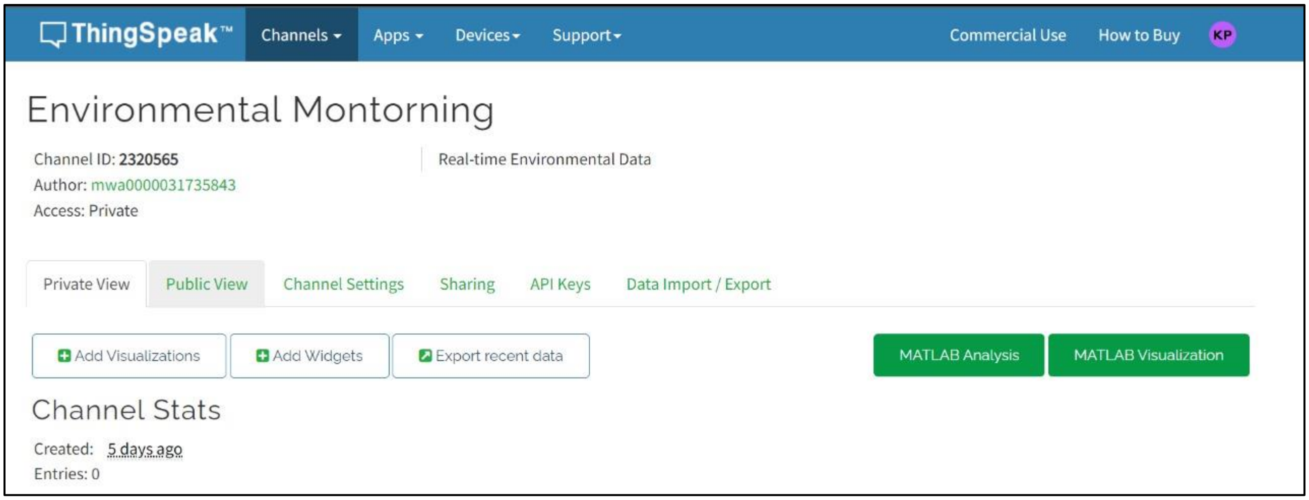
TO RECEIVE DATA FROM ARDUINO UNO



Signed in successfully.

My Channels

Name	Created	Updated
Environmental Montorning	2023-10-26	2023-10-26 13:23



Environmental Montorning

Channel ID: **2320565**
Author: mwa0000031735843
Access: Private

Real-time Environmental Data

Channel Stats

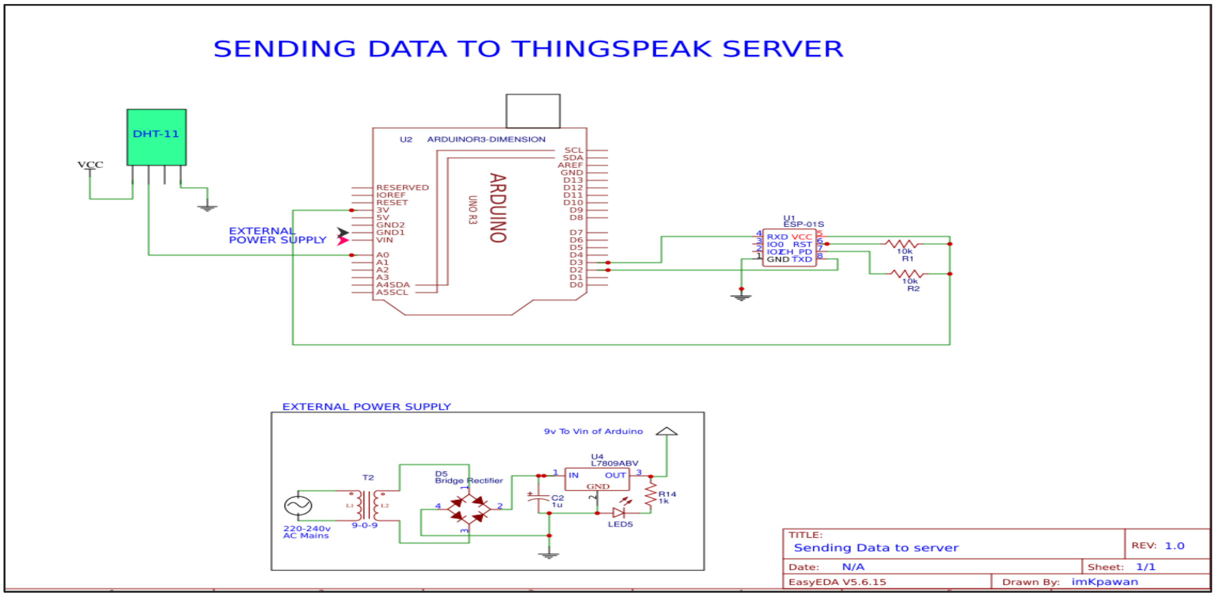
Created: 5 days ago
Entries: 0

CODE:

```
// Thingspeak
String myAPIkey = "OII2KGS7KGFTVLW5";
#include <SoftwareSerial.h>
#include <DHT.h>;
SoftwareSerial ESP8266(2, 3); // Rx,  Tx
/* DHT SENSOR SETUP */
#define DHTTYPE DHT11
#define DHTPIN  A0
DHT dht(DHTPIN, DHTTYPE,11);
float humidity, temp_f;
long writingTimer = 17;
long startTime = 0;
long waitTime = 0;
boolean relay1_st = false;
boolean relay2_st = false;
unsigned char check_connection=0;
unsigned char times_check=0;
boolean error;
void setup()
```

```
{
  Serial.begin(9600);
  ESP8266.begin(9600);
   dht.begin();
  startTime = millis();
  ESP8266.println("AT+RST");
  delay(2000);
  Serial.println("Connecting to Wifi");
   while(check_connection==0)
  {
    Serial.print(".");
  ESP8266.print("AT+CWJAP=\"TP-LINK_FDBA\",\"jaishrikrishna~12\"\r\n");
  ESP8266.setTimeout(5000);
 if(ESP8266.find("WIFI CONNECTED\r\n")==1)
 {
 Serial.println("WIFI CONNECTED");
 break;
 }
 times_check++;
 if(times_check>3)
 {
  times_check=0;
   Serial.println("Trying to Reconnect..");
  }
  }
}
void loop()
{
  waitTime = millis()-startTime;
  if (waitTime > (writingTimer*1000))
  {
    readSensors();
    writeThingSpeak();
    startTime = millis();
  }
}
void readSensors(void)
{
  temp_f = dht.readTemperature();
  humidity = dht.readHumidity();
}
void writeThingSpeak(void)
{
  startThingSpeakCmd();
  // preparacao da string GET
  String getStr = "GET /update?api_key=";
  getStr += myAPIkey;
  getStr +="&field1=";
  getStr += String(temp_f);
  getStr +="&field2=";
  getStr += String(humidity);
  getStr += "\r\n\r\n";
  GetThingspeakcmd(getStr);
}
void startThingSpeakCmd(void)
{
  ESP8266.flush();
  String cmd = "AT+CIPSTART=\"TCP\",\"";
  cmd += "184.106.153.149"; // api.thingspeak.com IP address
  cmd += "\",80";
  ESP8266.println(cmd);
  Serial.print("Start Commands: ");
  Serial.println(cmd);
  if(ESP8266.find("Error"))
  {
    Serial.println("AT+CIPSTART error");
    return;
  }
}
String GetThingspeakcmd(String getStr)
{
  String cmd = "AT+CIPSEND=";
  cmd += String(getStr.length());
  ESP8266.println(cmd);
  Serial.println(cmd);
  if(ESP8266.find(">"))
  {
    ESP8266.print(getStr);
    Serial.println(getStr);
    delay(500);
    String messageBody = "";
    while (ESP8266.available())
    {
      String line = ESP8266.readStringUntil('\n');
      if (line.length() == 1)
      {
        messageBody = ESP8266.readStringUntil('\n');
      }
    }
    Serial.print("MessageBody received: ");
    Serial.println(messageBody);
    return messageBody;
  }
  else
  {
    ESP8266.println("AT+CIPCLOSE");
    Serial.println("AT+CIPCLOSE");
  }
}
```

SENDING DATA TO THINGSPEAK



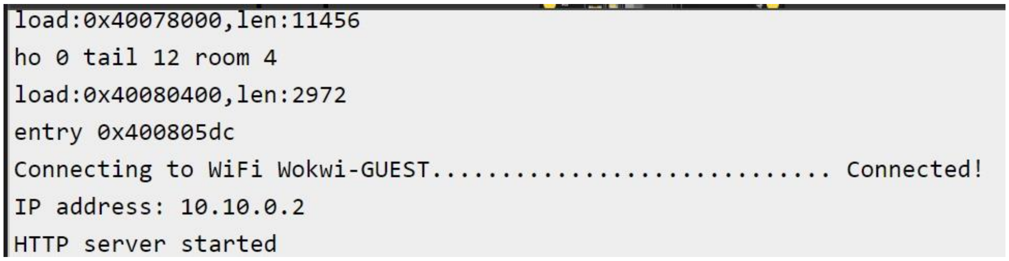
Use web development technologies (e.g., HTML, CSS, JavaScript) to create a platform that displays real-time environmental data.

ARDUINO TO WEBSITE USING WIFI

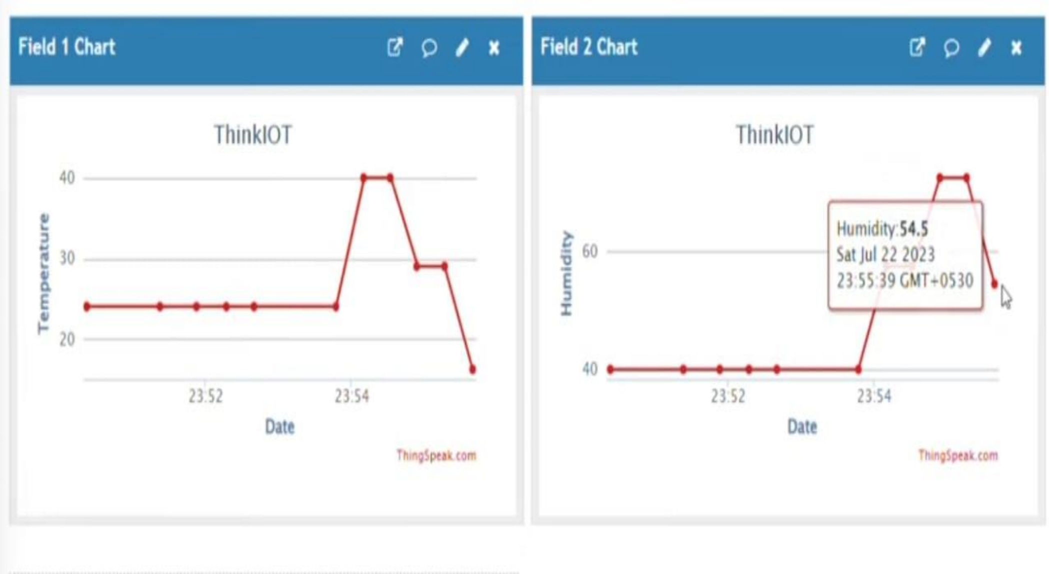
```
#include <WiFi.h>

#include <PubSubClient.h>

#include <DHTesp.h>

const int DHT_PIN = 15;

DHTesp dht;

const char* ssid = "Workwi-"; ///  wifi ssid

const char* password = "87654321";

const char* mqtt_server = "test.mosquitto.org";// mosquitto server url

WiFiClient espClient;

PubSubClient client(espClient);

unsigned long lastMsg = 0;

float temp = 0;

float hum = 0;

void setup_wifi() {

  delay(10);

  Serial.println();

  Serial.print("Connecting to ");

  Serial.println(ssid);

  WiFi.mode(WIFI_STA);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }}
void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      Serial.println("Connected");
      client.publish("/ThinkIOT/Publish", "Welcome");
      client.subscribe("/ThinkIOT/Subscribe");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }}
```

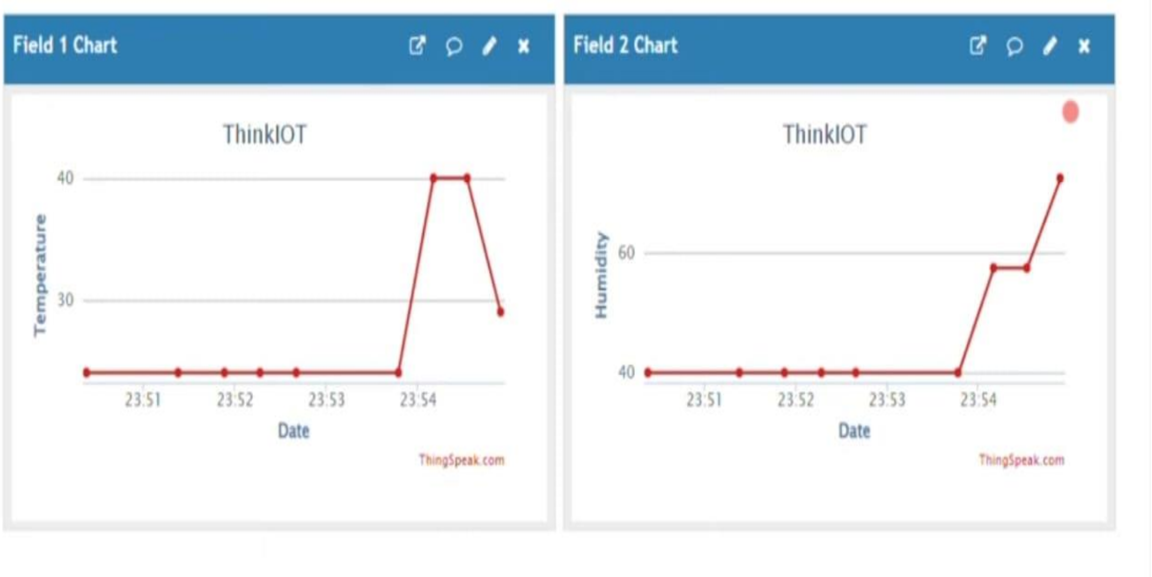
```
}
void setup() {
  pinMode(2, OUTPUT);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  dht.setup(DHT_PIN, DHTesp::DHT22);
}
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  unsigned long now = millis();
  if (now - lastMsg > 2000) { //perintah publish data
    lastMsg = now;
    TempAndHumidity  data = dht.getTempAndHumidity();
    String temp = String(data.temperature, 2);
    client.publish("/Thinkitive/temp", temp.c_str()); // publish temp topic /ThinkIOT/temp
    String hum = String(data.humidity, 1);
    client.publish("/Thinkitive/hum", hum.c_str());   // publish hum topic /ThinkIOT/hum
    Serial.print("Temperature: ");
    Serial.println(temp);
    Serial.print("Humidity: ");
    Serial.println(hum);
  }
}
```

OUTPUT



```
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
Connecting to WiFi Wokwi-GUEST........................... Connected!
IP address: 10.10.0.2
HTTP server started
```


Field 1 Chart
ThinkIOT
Temperature
23:52
23:54
Date
ThingSpeak.com
Field 2 Chart
ThinkIOT
Humidity
Humidity: 54.5
Sat Jul 22 2023
23:55:39 GMT+0530
23:52
23:54
Date
ThingSpeak.com


Field 1 Chart
ThinkIOT
Temperature
23:51
23:52
23:53
23:54
Date
ThingSpeak.com
Field 2 Chart
ThinkIOT
Humidity
23:51
23:52
23:53
23:54
Date
ThingSpeak.com