College name: JP COLLEGE OF ENGINEERING
College code: 9512

Project ID : Proj_211934_Team_1

TEAM MEMBERS:

1. HEMAPRIYA.C (au951221106013)
2. MUTHURAMA.K (au951221106027)
3. MURUGESHWARI.N (au951221106026)
4. ANISHA PUSPHAM.A (au951221106004)
5. RAMYA.N (au951221106032)

Phase 5 : Documentation

Topic: Environmental Monitoring system

OBJECTIVE:

The monitoring system is used for know or get an information on a situation or condition in a certain place or area, with various methods used, so that many monitoring systems are developed and applied to assist human tasks in carrying out .Using Internet of Things (IOT), we can control any electronic equipment in homes and industries. Moreover, you can read a data from any sensor and analyses it graphically from anywhere in the world.Here we deploy the . Here, we can read temperature and humidity data in park using IOT techniques.

IOT DEVICE DEPLOYMENT :

1.Hardware Selection: Choose suitable IoT devices (e.g., temperature and humidity sensors) and microcontrollers (e.g., Raspberry Pi, Arduino) for your project.

2. *Sensor Placement*: Identify strategic locations within the parks for sensor deployment to collect relevant environmental data.

3. *Connectivity*: Ensure that the devices have access to the internet, either through Wi-Fi or cellular networks.

4. *Programming*: Develop a Python script for the IoT devices. You'll need to program the devices to read data from sensors and transmit it to a central monitoring platform.

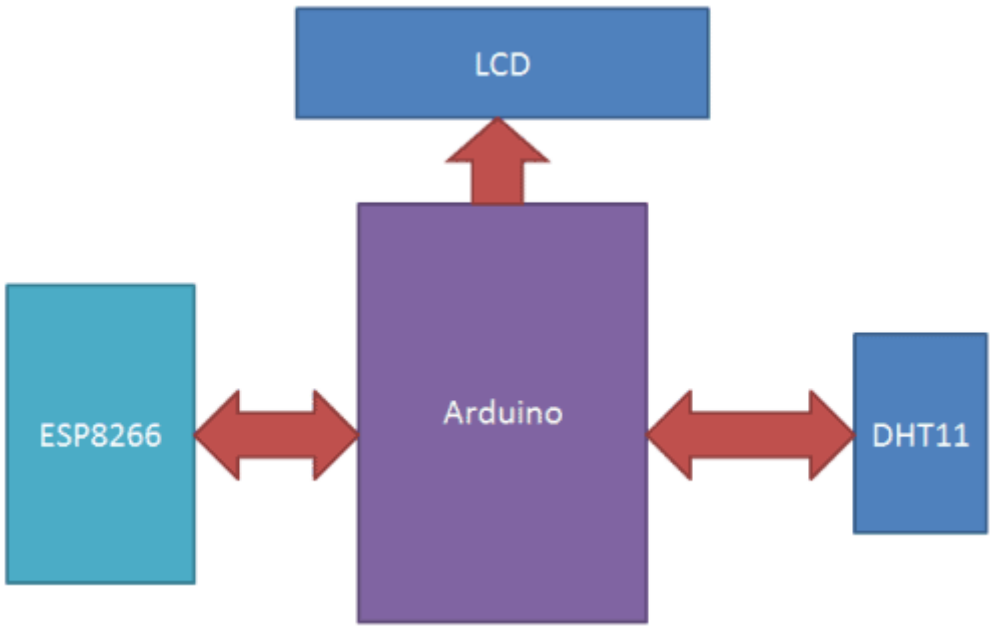
5. *Monitoring Platform*: Set up a monitoring platform that can receive and

store the incoming data. This platform can be a cloud-based server or a local server, depending on your project's scale.

6. *Data Storage*: Decide how you want to store the data. You can use databases (e.g., MySQL, MongoDB) or cloud storage solutions (e.g., AWS S3, Google Cloud Storage).

WORKING PRINCIPLE:

This IoT based project having four sections, firstly Humidity and Temperature Sensor DHT11 senses the Humidity and Tempe**rature** Data**.** Secondly Arduino Uno extracts the DHT11 sensor's data as suitable number in percentage and Celsius scale, and sends it to Wi-Fi Module. Thirdly Wi-Fi Module ESP8266 sends the data to Tinkercat. **And finally** Tinker Cat analyses the data and shows **it in a** Graph form. Optional LCD **is also used to display** the Temperature and Humidity.



PYTHON SUBSCRIPT:

These code are used to implelement the project. these code are dumbed on IOT devices.

```
import network
import time
From machine import Pin
Import dht
Import ujson
```

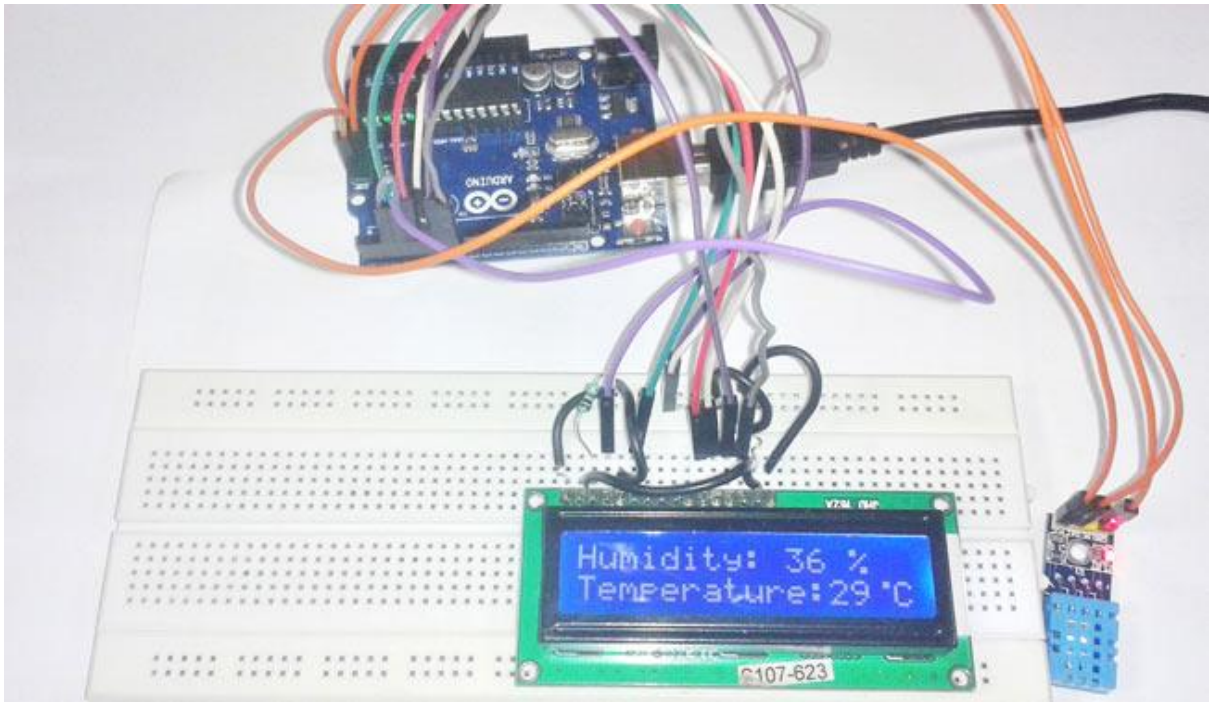
```
From umqtt.simple import MQTTClient

# MQTT Server Parameters
MQTT_CLIENT_ID = "micropython-weather-demo"
MQTT_BROKER    = "broker.mqttdashboard.com"
MQTT_USER      = ""
MQTT_PASSWORD  = ""
MQTT_TOPIC     = "wokwi-weather"

sensor = dht.DHT22(Pin(15))

print ("Connecting to Wi-Fi", end="")
sta_if = network.WLAN(network.STA_IF)
sta_if.active(True)
sta_if.connect('Wokwi-GUEST', '')
while not sta_if.isconnected():
print(".", end="")
time.sleep(0.1)
print(" Connected!")

print("Connecting to MQTT server... ", end="")
client = MQTTClient (MQTT_CLIENT_ID, MQTT_BROKER, user=MQTT_USER,
password=MQTT_PASSWORD)
client.connect()

print ("Connected!")

prev_weather = ""
while True:
Print ("Measuring weather conditions... ", end="")
sensor.measure()
message = ujson.dumps({
"temp": sensor. temperature (),
"Humidity": sensor. Humidity (),
})
if message != prev_weather:
print("Updated!")
print("Reporting to MQTT topic {}: {}".format(MQTT_TOPIC, message))
client.publish(MQTT_TOPIC, message)
prev_weather = message
else:
print("No change")
time.sleep(1)
```

simulation output:

```
 (POWERON_RESET), boot: 0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:4728
load:0x40078000,len:14876
ho 0 tail 12 room 4
load:0x40080400,len:3368
```

```
entry 0x400805cc
Connecting to
WiFi.....................................................................................................................
.....................................................................................................................
.....................................................................................................................
..... Connected!
Connecting to MQTT server... Connected!
Measuring weather conditions... Updated!
Reporting to MQTT topic wokwi-weather: {"humidity": 40.0, "temp": 24.0}
Measuring weather conditions... No change
Measuring weather conditions... No change
Measuring weather conditions... No change
Measuring weather conditions... No change
Measuring weather conditions... No change
Measuring weather conditions... Updated!
Reporting to MQTT topic wokwi-weather: {"humidity": 80.5, "temp": 48.8}
Measuring weather conditions... Updated!
Reporting to MQTT topic wokwi-weather: {"humidity": 80.5, "temp": -13.8}
Traceback (most recent call last):
  File "main.py", line 62, in <module>
  File "umqtt/simple.py", line 134, in publish
OSError: [Errno 104] ECONNRESET
MicroPython v1.21.0 on 2023-10-05; Generic ESP32 module with ESP32
Type "help ()" for more information.
```

DEVELOPER MODEL:



WEBDEVELOPMENT CODE:

C ++ program:

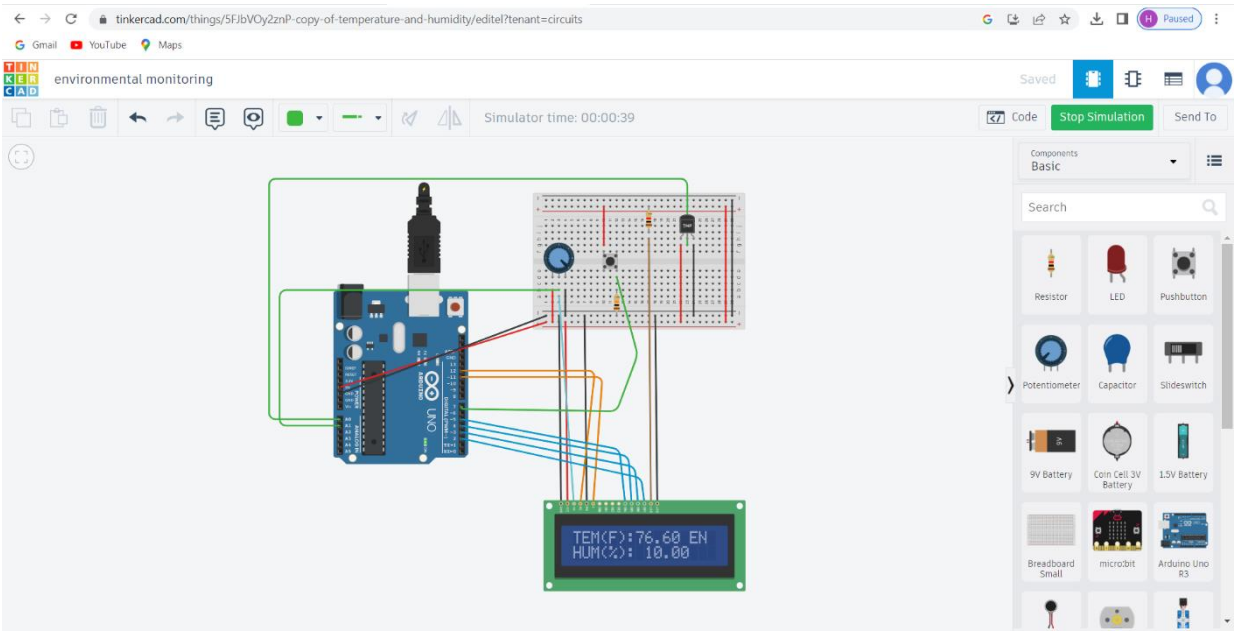
```
// include the library code:
#include <LiquidCrystal


// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);


// Defining Variables
const int analogIn = A0;
int humiditysensorOutput = 0;
int RawValue= 0;
double Voltage = 0;
double tempC = 0;
double tempF = 0;
int buttonState = 0;
```

```
char LCD_LANGUAGE = 'E';

void setup()
{
        Serial.begin(9600);
        pinMode(A1, INPUT);
        pinMode(7, INPUT);
        lcd.begin(16, 2);
}

void loop()
{
 RawValue = analogRead(analogIn);
 Voltage = (RawValue / 1023.0) * 5000; // 5000 to get millivots.
 tempC = (Voltage-500) * 0.1; // 500 is the offset
 tempF = (tempC * 1.8) + 32; // convert to F
 double humidity = map(humiditysensorOutput, 0, 1023, 10, 70);
 buttonState = digitalRead(7);

 if (buttonState == HIGH) {
  LCD_LANGUAGE = 'T';
 } else {
        LCD_LANGUAGE = 'E';
 }

 if (LCD_LANGUAGE == 'T') {
  lcd.setCursor(0,0);
  lcd.print("SIC(C):" + String(tempC,2) + " TR");
  humiditysensorOutput = analogRead(A1);
  lcd.setCursor(0,1);
  lcd.print("NEM(%): " + String(humidity,2));
 }
 else {
  lcd.setCursor(0,0);
```

```
  lcd.print("TEM(F):" + String(tempF,2) + " EN");
  humiditysensorOutput = analogRead(A1);
  lcd.setCursor(0,1);
  lcd.print("HUM(%): " + String(humidity,2));
}
}
```

ENHANCEMENT OF THE PUBLIC SAFTEY:

It give the safety to the public by monitoring the temperature and humidity in envirironmental.Due to the high temperature the people will suffer a lot .the humidity is monitor and control by the developed model so the people are protect against high tempearature and humidity. The ouput of the developed model verified by the real time example.the output is shown in below



CONCLUSION :

In conclusion, an Environmental Monitoring System using the Internet of Things (IoT) represents a transformative and highly valuable technology for addressing a wide range of environmental
challenges. This system harnesses the power of interconnected sensors, devices, and data analytics to collect, manage, and analyse environmental data in real-time.