WEATHER MONITORING SYSTEM

# Krish Sharma (21Z315)

# Umesh Gupta (21Z365) Prasun Jha (21Z371) Nilarghya Sarkar (21Z372)

**Vasanth C (22z462)**

**19Z604 - EMBEDDED SYSTEMS**

report submitted in partial fulfillment of the requirement for the award of degree of

**BACHELOR OF ENGINEERING**

**Branch: COMPUTER SCIENCE AND ENGINEERING**

Of Anna University



April 2024

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**PSG COLLEGE OF TECHNOLOGY**

**(Autonomous Institution)**

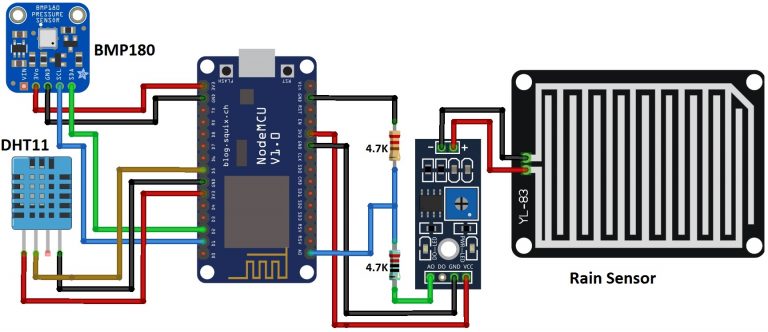
# PROBLEM STATEMENT:

In rural agricultural communities worldwide, the absence of accessible, accurate, and affordable weather monitoring systems poses significant challenges to sustainable farming practices. Due to the lack of localized weather data, farmers often rely on distant weather forecasts or expensive commercial solutions, both of which may not adequately reflect the specific microclimatic conditions of their fields. This reliance on generalized weather information leads to suboptimal decision-making regarding crop management, irrigation scheduling, pest control, and resource allocation.

Moreover, the existing gap in weather monitoring infrastructure exacerbates the vulnerability of rural farmers to weather-related risks such as droughts, floods, and extreme temperature fluctuations. Without timely and accurate weather information, farmers struggle to adapt their agricultural practices to changing environmental conditions, resulting in decreased crop yields, increased resource wastage, and heightened economic vulnerability.

The integration of sensor technology with Arduino microcontrollers offers a promising avenue for developing cost-effective and locally deployable weather monitoring solutions tailored to the needs of rural agricultural communities.

# SCHEMATIC DIAGRAM:

****

**EMBEDDED C CODE:**

#include <SoftwareSerial.h>

#include "DHT.h"

SoftwareSerial ESP8266(2, 3); // Rx, Tx

long writingTimer = 17;

long startTime = 0;

long waitTime = 0;

const int sensor=A0; // Assigning analog pin A0 to variable 'sensor'

const int rain\_sensor=4; // Assigning analog pin A1 to variable 'sensor'

const int DHT11\_Sesnor = 5;

#define DHTTYPE DHT11

DHT dht(DHT11\_Sesnor, DHTTYPE);

int humudity\_value;

float tempc; //variable to store temperature in degree Celsius

float vout; //temporary variable to hold sensor reading

bool rain\_status = 0;

unsigned char check\_connection=0;

unsigned char times\_check=0;

boolean error;

String myAPIkey = "8GX33VU8QE8WXEH9";

void setup()

{

Serial.begin(9600);

ESP8266.begin(9600);

pinMode(rain\_sensor, INPUT);

dht.begin();

startTime = millis();

delay(2000);

Serial.println("Connecting to Wifi");

while(check\_connection==0)

{

Serial.print(".");

ESP8266.print("AT+CWJAP=\"2.4g\",\"prasun@2310\"\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))

{

vout=analogRead(sensor);

vout=(vout\*500)/1023;

tempc=vout; // Storing value in Degree Celsius

rain\_status = digitalRead(rain\_sensor);

readDTH11\_Sesnor();

writeThingSpeak();

startTime = millis();

}

}

void readDTH11\_Sesnor()

{

// Reading temperature or humidity takes about 250 milliseconds!

humudity\_value = dht.readHumidity();

// Check if any reads failed and exit early (to try again).

if (isnan(humudity\_value))

{

Serial.println(("Failed to read from DHT sensor!"));

return;

}

// Serial.print((" Humidity: "));

//Serial.print(humudity\_value);

// Serial.print(("%"));

// Serial.print("\n");

// delay(100);

}

void writeThingSpeak(void)

{

startThingSpeakCmd();

// preparacao da string GET

String getStr = "GET /update?api\_key="8GX33VU8QE8WXEH9";

getStr += myAPIkey;

getStr +="&field1=";

getStr += String(tempc);

getStr +="&field2=";

getStr += String((rain\_status == 0 ? 1:0));

getStr +="&field3=";

getStr += String(humudity\_value);

getStr += "\r\n\r\n";

GetThingspeakcmd(getStr);

}

void startThingSpeakCmd(void)

{

ESP8266.flush();

String cmd = "AT+CIPSTART=\"TCP\",\"";

cmd += "184.106.153.149";

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");

}

}

**CHANGE BAUDE RATE CODE:**

#include <SoftwareSerial.h>

#include <stdlib.h>

SoftwareSerial ESP8266(2, 3); // RX, TX

void setup() {

Serial.begin(115200);

ESP8266.begin(115200);

ESP8266.print("\*\*\*VER:");

delay(2000);

ESP8266.println("AT+RST");

delay(1000);

}

void loop() {

ESP8266.println("AT+GMR");

delay(3000);

ESP8266.println("AT+CWMODE=3");

delay(3000);

ESP8266.println("AT+UART\_DEF=9600,8,1,0,0");

delay(3000);

Serial.println("Done");

while(1);

}

**CONNECTION WITH WI-FI CODE**

#include <SoftwareSerial.h>

#include <stdlib.h>

SoftwareSerial ESP8266(2, 3); // RX, TX

unsigned char check\_connection=0;

unsigned char times\_check=0;

void setup() {

Serial.begin(9600);

ESP8266.begin(9600);

ESP8266.print("\*\*\*VER:");

delay(2000);

ESP8266.println("AT+RST");

delay(1000);

ESP8266.println("AT+GMR");

delay(1000);

ESP8266.println("AT+CWMODE=3");

delay(1000);

ESP8266.println("AT+CWLAP");

delay(1000);

}

void loop() {

Serial.println("Connecting to Wifi");

while(check\_connection==0)

{

Serial.print(".");

ESP8266.print("AT+CWJAP=\"RJ\",\"1234567898\"\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..");

}

}

while(1);

}

**READING OF SENSORS:**

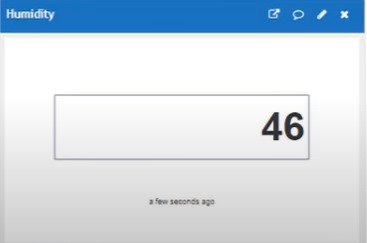
1. **TEMPERATURE**

|  |  |
| --- | --- |
| **C:\Users\student.PLAB56\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WhatsApp Image 2024-04-17 at 3.13.23 PM.JPEG** | **C:\Users\student.PLAB56\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WhatsApp Image 2024-04-17 at 3.09.53 PM.JPEG** |

**2.RAIN**

|  |  |
| --- | --- |
| **C:\Users\student.PLAB56\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WhatsApp Image 2024-04-17 at 3.50.17 PM.JPEG** | **C:\Users\student.PLAB56\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WhatsApp Image 2024-04-17 at 3.09.15 PM.JPEG** |

1. **HUMIDITY**

****