IOT BASED WEATHER STATION

[1]Harishkumar Yetam, [2]Kapil Sahu, [3]Pankaj Bhasme, [4]Shubhangi Dahake, [5]Rajesh Kothekar, [6]Prof. Deepak Bhoyar

Department of Electronics and Communication Engineering
Tulsiramji Gaikwad Patil Institute of Technology, Nagpur

***Abstract*: A Weather station can be labelled as an instrumentor device, which provides us with the information of the meteorological conditions in our neighbouring environment. For example it can provide us with details about the surrounding temperature, humidity, etc. Hence, this device basically senses the temperature, pressure, humidity, light intensity and rain value. Thereare several types of sensors available in the sample, using whichall the aforementioned strictures can be measured. It can be used to display the temperature and humidity of a specific room/place. With the help of temperature and humidity we can calculate other data strictures, such as the dew point. In addition to the above mentioned functionalities, we can display the light intensity of the place as well. We have also enabled to display the atmospheric pressure of the room. We can also display the rain value. The brain of the sample is the ESP8266 based Wi-fi module Nodemcu (12E). Raindrop module, whenever these values exceed a chosen threshold limit for each a display, and mobile device and a Tweet post is published alerting the owner of the appliance to take necessary measures.**

***Index Terms*: IoT, ESP8266, Sensors**

I. INTRODUCTION

Through the start of high speed Internet, more and more humans around the globe are interconnected. In the world Internet of Things (IoT) takes this a step more, and connects not only humans but electronic devices which can speak amongst themselves [1]. With falling costs of Wifi enabled devices thistrend will only pleat more momentum. The main concept behind the Internet of Things (IoT) is to connect several electronic devices through a network and then recover thedata from these devices (sensors) which can be spread in any fashion, upload them to any fog service where one can analyze and process the collected information. In the fag service one can apply these data to alert people by many means such as using mobile ip address show the data.

As mentioned earlier, IoT enables not only Human- interaction, but also Human-Device interaction as well as Device-Device interaction. This particular development in the shape of new streets of interactions will impact essentially every industry such as conveyance and logistics, energy, healthcare etc. For example, in the case of energy, IoT is being applied to create Smart Grids which can detect and respond to changes in local and larger level changes in energy consumption, which is going to be an integral part of any nations energy policy.
Looking beyond the aforesaid energy example, there are many areas of interests where IoT can make a meaningful impact such human life such as, Smart Homes, which involve IoT to amplify the degree of automation; Vesture technologies such as smartwatches and fitness bands; One of the biggest areas of potential in IoT is connected healthcare.
Many universal electronics behemoths have already invested deeply in the Internet of Things infrastructure. With analysts predicting there will be 26 Billion connected devices, more than 4 per human on the planet, and the industry is projected to bring in $19 Trillion, in costs savings and profits with firms like Samsung and Google leading the pack.

With this new technological platform however, comes its own set of challenges and obstacles, such as what to do with the enormous amounts of data which is collected
This project as well measures environmental strictures such as temperature, humidity, , rain etc. In the cloud the data are analyzed and if the retrieved datas are above or below a certain threshold limit, depending on the value, an e-mail, an SMS and a twitter post is published at the exact moment [3].

Earlier people staying in home and busy in their home chores or people busy in their offices work pressure had no idea about the environmental strictures outside their home or office. They have no idea about enivermental temperature in outside is quite high or quite low or normal or if it is raining outside or not or what is the value of the humidity in the outside environment. This device can come in quite a handy in these situations.It will notify us whenever the temperature is too low or too high through an mobile device and get into display. It willalso automatically notify whenever there is a rainstorm in the surrounding and remind us to carry an umbrella or a raincoat It will also greet us with good morning and good evening messages as it also has an which measures to the surrounding environment. The C program written in Arduino IDE and uploaded to the ESP8266 through a serial bus. Once the code is uploaded then the board is connected to a Wifi and the device will be start. Working. The code has to be uploaded only once.

II. IMPLEMENTATION SETUP

A. Components required: Hardware

1) ESP8266 based wifi module Nodemcu[6]
2) Temperature and Humidity Sensor(DHT11)[7]
3) Raindrop Module[10]
4) Mobile phone to recieve in Ip.address
5) LCD Display 16*2
6) Hardware 7805
7) AC Power supply

B. Components required: Software

1) Arduino IDE 1.8.6 (2021 version)
2) Proteous 8.3 (2017 version)
3) Express PCB 7.8.0 (2016 version)

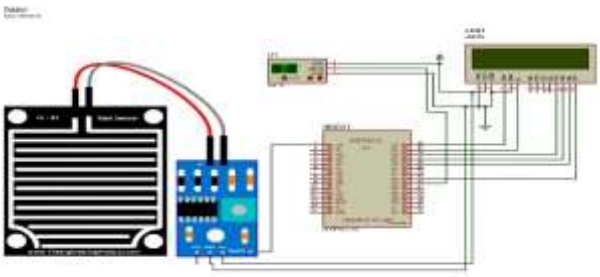
Fig. 1. The complete setup of the device

Fig. 3. DHT11.

C. DHT-11 (Temperature Sensor)

Its technology ensures the high reliability and excellent long-term stability. It has excellent quality, fast response, anti-interference ability and high performance. Each DHT11 sensors features extremely accurate calibration of humidity calibration chamber. The calibration coefficients stored in the OTP program memory,

E. Circuit Diagram:



D. Node mcu:

Fig. 2. Node mcu

It is the heart of the device. It provides the platform for IOT. It's a wifi module having esp8266 firmware within. All the other sensors are connected to this micro-controler. They send the measured values to it and it uploads all the values to the cloud where the values are analyzed. The developer of this board is ESP8266 Opensource Community. It has an operatingsystem called XTOS. The CPU is ESP8266(LX106). It has an in-built memory of 128 KBytes and a storage capacity of 4 MBytes.

F, ,Raindrop Module

It is used for the detection of rain. It can also be used for measuring the intensity of the rain. It has both digital outputas well as analog output. This module measures the moisture through analog output pin and when the threshold of moisture exceeds too much it provides a digital output. The more water or the lower resistance means lower output voltage. Where as, the less water means higher resistance,i.e, high output voltage on the analog pin. For example a completely dry board will cause the module to output five volts. The analog output ofthe module is connected to the A0 pin of the nodemcu. Fig 6.

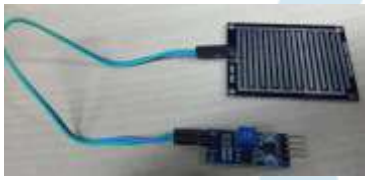


Fig. 6. Raindrop Module.

G, power Supply 12v : fig 7

Fig. 7

C. Raindrop Module

It is used for the detection of rain. It can also be used for measuring the intensity of the rain. It has both digital outputas well as analog output. This module measures the moisture through analog output pin and when the threshold of moisture exceeds too much it provides a digital output. The more water or the lower resistance means lower output voltage. Whereas, the less water means higher resistance,i.e, high output voltage on the analog pin. For example a completely dry board will cause the module to output five volts. The analog output ofthe module is connected to the A0 pin of the nodemcu.

Aurdino Software:

```
#include <ESP8266WiFi.h>
#include <LiquidCrystal.h>
#include "DHT.h"

#define LCD_RS          16
#define LCD_EN          5
#define LCD_DB4         4
#define LCD_DB5         2
```

```
#define LCD_DB6         14
#define LCD_DB7         12
#define DHTPIN          0
#define DHTTYPE           DHT11   // DHT 11

WiFiServer server(80);
LiquidCrystal lcd(LCD_RS,LCD_EN,LCD_DB4,LCD_DB5,LCD_DB6,LCD_DB7);
DHT dht(DHTPIN, DHTTYPE);

void gpio_init(void);
void lcd_init(void);
void update_lcd(void);
void update_web(void);

void read_dht11(void);

int s=0;
float h,t,f,hic,hif;
int rain;

void setup()
{

lcd_init();
gpio_init();

  lcd.setCursor(0,0);
  lcd.print("System Started");
  lcd.setCursor(0,1);
  delay(3000);
}

void loop()
{
  lcd.setCursor(0,0);
  lcd.print("Temp:-      ^C ");
  lcd.setCursor(7,0);
  lcd.print(t);

  lcd.setCursor(0,1);
  lcd.print("Hum :-      %  ");
  lcd.setCursor(7,1);
  lcd.print(h);
}

void gpio_init(void)
{   delay(50);
    Serial.begin(9600);
    delay(50);

  Serial.print("Connecting to ");
  Serial.println(ssid);
   {
    delay(500);
    Serial.print(".");
   }
  Serial.println("");
  Serial.println("WiFi connected");
  server.begin();
  Serial.println("Server started");
  Serial.print("Use this URL to connect: ");
  Serial.println("/");
```

```
}
  void lcd_init(void)
  {
  lcd.begin(16, 2);
  lcd.setCursor(4,0);
  lcd.print(" Welcome ");
  delay(2000);

  lcd.setCursor(0,0);
  lcd.print("    Weather     ");
  lcd.setCursor(0,1);
  lcd.print("Monitoring   IOT");
  delay(3000);

  lcd.setCursor(0,0);
  lcd.print("Connecting IOT  ");
  lcd.setCursor(0,1);
  lcd.print("Please Wait ....");
  }

void update_web(void)
{

  WiFiClient client = server.available();
  if (!client)
  {
    return;
  }
 // Serial.println("new client");
  while(!client.available())
  {
    delay(1);
  }
  String request = client.readStringUntil('\r');
 // Serial.println(request);
  client.flush();
  delay(5);

     client.println("<HTML>");
     client.println("<HEAD>");
     client.println("<TITLE>ELECSTARK</TITLE>");
     client.println("<BODY>");

     client.println("<H1 style=color:orange;font-size:60px;text-align:center;>");
     client.println(" Weather Monitoring Station Using IOT ");
     client.println("</H1>");

     client.println("<H1 style=color:orange;font-size:40px;text-align:center;>");
     client.println("System Status");
     client.println("</H1>");

     client.println("<H1 style=color:green;font-size:40px;text-align:center;>");
     client.println("Temprature :- ");
     client.println(t);
     client.println("^C");
     client.println("</H1>");

     client.println("<H1 style=color:red;font-size:40px;text-align:center;>");
     client.println("Temprature :- ");
     client.println(t);
     client.println("^C");
     client.println("</H1>");
```

```
    client.println("<H1 style=color:green;font-size:40px;text-align:center;>");
    client.println("Hummidity :- ");
    client.println(h);
    client.println("%");
    client.println("</H1>");

    client.println("<H1 style=color:green;font-size:40px;text-align:center;>");
    client.println("Rain      :- ");
    client.println(rain);
    client.println("%");
    client.println("</H1>");

    client.println("<H1 style=color:red;font-size:40px;text-align:center;>");
    client.println("Rain      :- ");
    client.println(rain);
    client.println("%");
    client.println("</H1>");

  }

void read_dht11(void)   {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  // Compute heat index in Fahrenheit (the default)
   hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
    hic = dht.computeHeatIndex(t, h, false);
/*
  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(F("°F  Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C "));
  Serial.print(hif);
  Serial.println(F("°F"));

*/
  int q = analogRead(A0);
  q = 1023 - q;

  if(rain <= 0)
  rain = 0;

  }
```

REFERENCES

[1] M. H. Asghar, A. Negi, and N. Mohammadzadeh, "Principle application and vision in internet of things (iot)," in *International Conference on Computing, Communication Automation*, May 2015, pp. 427–431.

[2] A. Gheith, R. Rajamony, P. Bohrer, K. Agarwal, M. Kistler, B. L. W. Eagle, C. A. Hambridge, J. B. Carter, and T. Kaplinger, "Ibm bluemix mobile cloud services," *IBM Journal of Research and Development*, vol. 60, no. 2-3, pp. 7:1–7:12, March 2016.

[3] S. Gangopadhyay and M. K. Mondal, "A wireless framework for environmental monitoring and instant response alert," in *2016 Interna- tional Conference on Microelectronics, Computing and Communications(MicroCom)*, Jan 2016, pp. 1–6.

[4] H. Saini, A. Thakur, S. Ahuja, N. Sabharwal, and N. Kumar, "Ar- duino based automatic wireless weather station with remote graphical application and alerts," in *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, Feb 2016, pp. 605–609

FUTURE SCOPE

The proposed IoT based weather station can be modifiedto incorporate many more features. We can add an OLED display to display the surrounding parameters into it. We can also add a GPS module in the design so that the locationof the surrounding will also be mailed or messaged to theuser along with the surrounding parameters, like, temperature, humidity, pressure, light intensity etc. It can also be modifeid such that whenever a message or email is sent from a particular phone number or email id to the server, all the environmental parameters of the device along with its locationwill be delivered to that phone or email id. This device can also be used to monitor a particular room or place whose environmental parameters are required to be monitored continuously.