IMPLEMENTING A WEATHER STATION USING IOT.

PROJECT REPORT

by

AMUKA ANURAG

(17BEC0853)

SAI AAKASH BHATNAGAR

(17BEC0593)

AVINASH KUMAR

(17BEC0332)

Department of Electronics and Communication
Vellore Institute of Technology
Vellore

April,2018.

Declaration by Authors

This is to declare that this report has been written by us. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be plagiarized, we are shall take full responsibility for it.

Anurag Amuka

Amuka Anurag
17BEC0853

Aakash

Sai Aakash Bhatnagar
17BEC0593

Avinash. Kr.

Avinash Kumar
17BEC0332

Vellore
2-04-2018

Abstract

We were asked to design a project to check the surrounding temperature, humidity and pressure. We have designed and implemented a weather station which can provide us with the information of the weather in our neighboring environment. It would provide us with details about the surrounding temperature, barometric pressure, humidity.

A weather station can be described as an instrument or device, which provides us with the information of the weather in our neighboring environment. There are various types of sensors present in the prototype, using which all the fore mentioned parameters can be measured.

The brain of the prototype is the ESP8266 based Wi-fi module NodeMCU(12E). Two sensors are connected to the NodeMCU namely temperature and humidity sensor(DHT11), pressure sensor(BMP180).

Basically, we did this project because we felt that we will be able to make use of the most of the useful sensors which would help us learn more about how the different sensors are used. We found this project the most interesting out of all the other projects. After all, it is the interest which matters.

Another reason of why we did this project because there is a lot of scope of IoT in the present day as well as in the future.

TABLE OF CONTENTS

INTRODUCTION

1.1 problem addressed:

As humans we want to know what’s happening around us. That’s our nature. We are enthusiastic. When it comes to atmospheric conditions ,it is must know the conditions around us , because we can be careful incase of any drastic changes that may lead to dangerous situations.so to solve this problem we came up with the idea of creating a small weather station, through which we can wirelessly access the data and get real time updates of atmospheric conditions around us .

1.2 Importance:

A weather station can be described as an instrument or device, which provides us with the information of the weather in our neighboring environment. For example it can provide us with details about the surrounding temperature, barometric pressure, humidity, etc. Hence, this device basically senses the temperature, pressure, humidity, light intensity, rain value. There are various types of sensors present in the prototype, using which all the fore mentioned parameters can be measured. It can be used to monitor the temperature or humidity of a particular room/place. With the help of temperature and humidity we can calculate other data parameters, such as the dew point

The brain of the prototype is the ESP8266 based Wi-fi module NodeMCU(12E). Two sensors are connected to the NodeMCU namely temperature and humidity sensor(DHT11), pressure sensor(BMP180). Whenever these values exceed a chosen threshold limit for each an SMS, an E-mail and a Tweet post is published alerting the owner of the appliance to take necessary measures. The temperature sensor sensed the different temperature and gave out the output. These sensors can be very useful in implementing a real-big-life model of weather system.

1.3 Related literature:

Now ,similar ideas were already there for someone ,they did similar project but they used Arduino and someone even used GSM module so that the user would get updates through sms.But here it is difficult to keep looking for messages, so we thought that if operate the sensor wirelessly and access the data ,then it would be easy for a consumer to keep updated about the conditions around him.

1.4 Scope of the project:

Now, scope of the project is we wirelessly operate the sensors and see the live working of sensor using blynk app .In upcoming chapters we are going to brief about approach used, results, discussion and conclusion, also we will give references.

1.5 Briefing:

COMPONENTS USED:

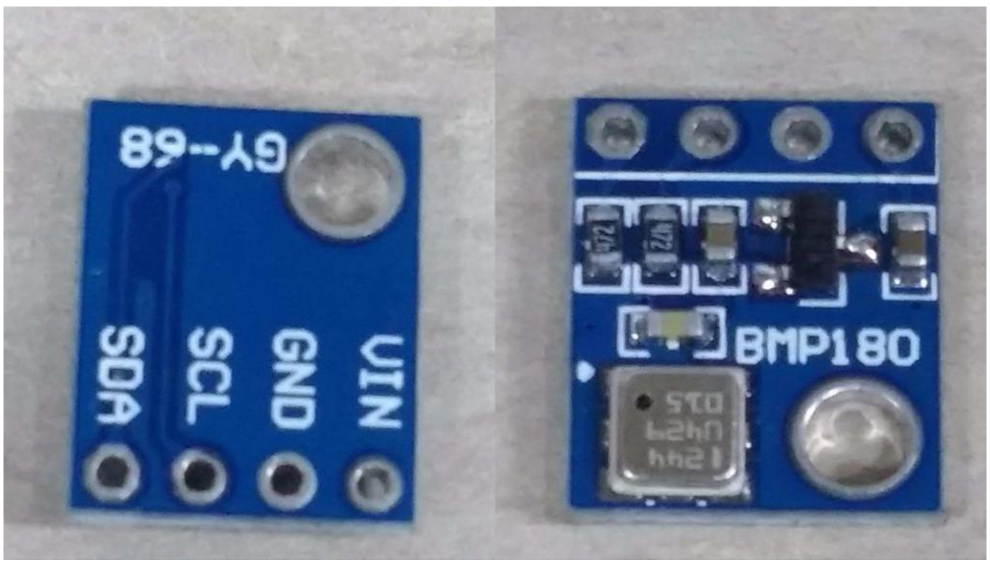
1. ESP8266 based wifi module Nodemcu
2. Temperature and Humidity Sensor(DHT11)
3. Barometric Pressure Sensor(BMP180)
4. JUMPER WIRES(M-M, M-F,F-F)
5. MOBILE/LAPTOP

ABOUT COMPONENTS:

NODEMCU: It provides the platform for IOT. It's a wifi module having esp8266 firmware within. All the other sensors are connected to this chip. 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 Open source Community. It has an operating system called XTOS. The CPU is ESP8266(LX106). It has an in- built memory of 128 KBytes and a storage capacity of 4Mbytes

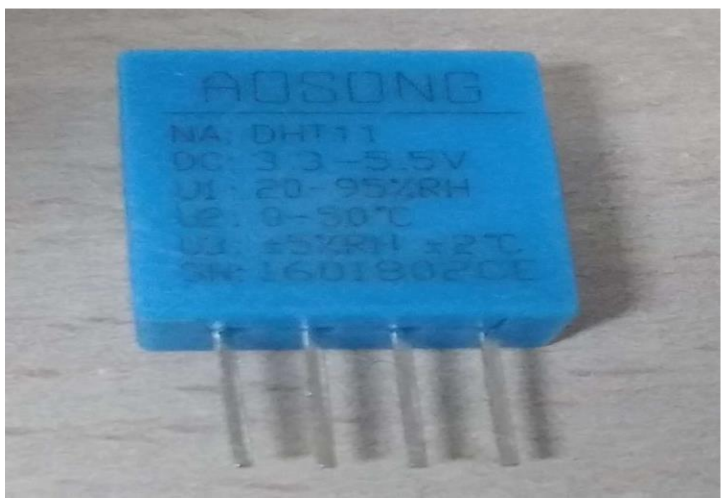
DHT-11 (Temperature Sensor): It senses the temperature of the surrounding. Its a 4-pindevice. Humidity sensors detect the relative humidity of the immediate environments in which they are placed. They measure both the moisture and temperature in the air and express relative humidity as a percentage of the ratio of moisture in the air to the maximum amount that can be held in the air at the current temperature. As air becomes hotter, it holds more moisture, so the relative humidity changes with the temperature.

Most humidity sensors use capacitive measurement to determine the amount of moisture in the air. This type of measurement relies on two electrical conductors with a non-conductive polymer film laying between them to create an electrical field between them. Moisture from the air collects on the film and causes changes in the voltage levels between the two plates. This change is then converted into a digital measurement of the air's relative humidity after taking the air temperature into account



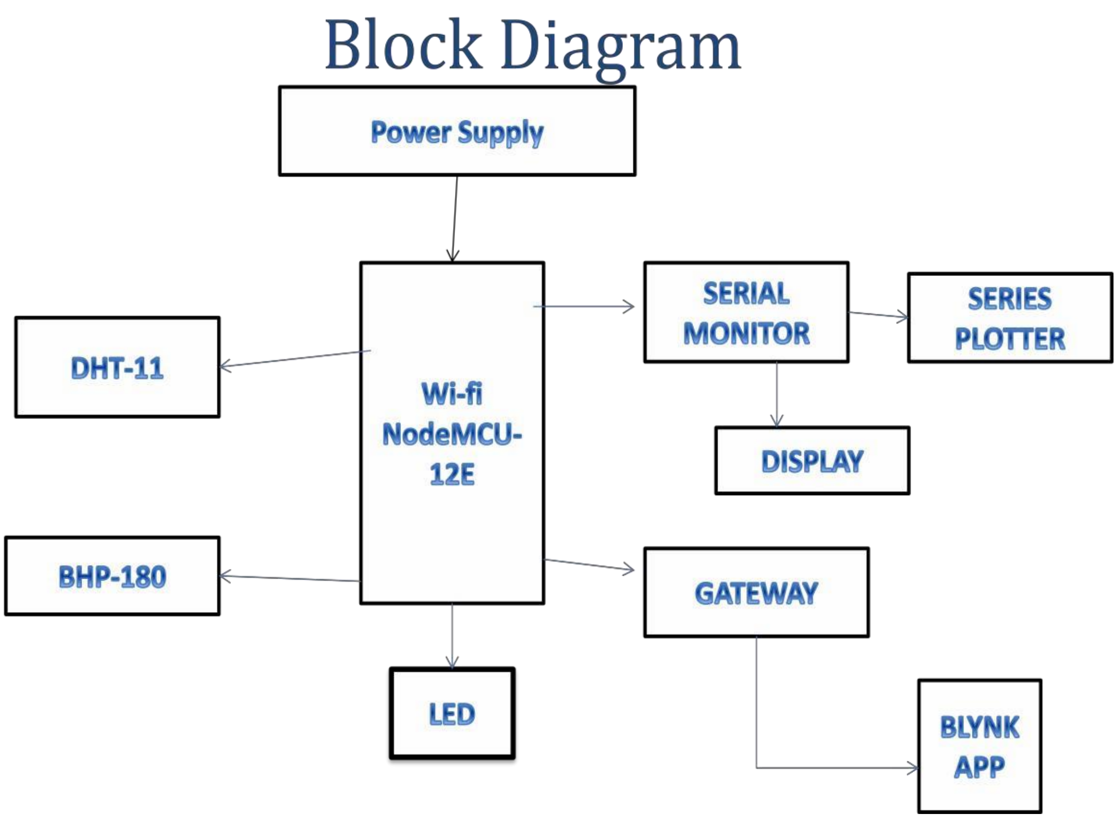
BMP 180(Pressure Sensor): It senses the barometric pressure from the surrounding. BMP180 is an I2C standard device. Its a 4-pin device, viz, SDA, SCL, VIN, GND. VIN and GND are connected to 3.3V and GND respectively. SDA is connected to D2 pin of nodemcu and SCL is connected to D3 pin of nodemcu.

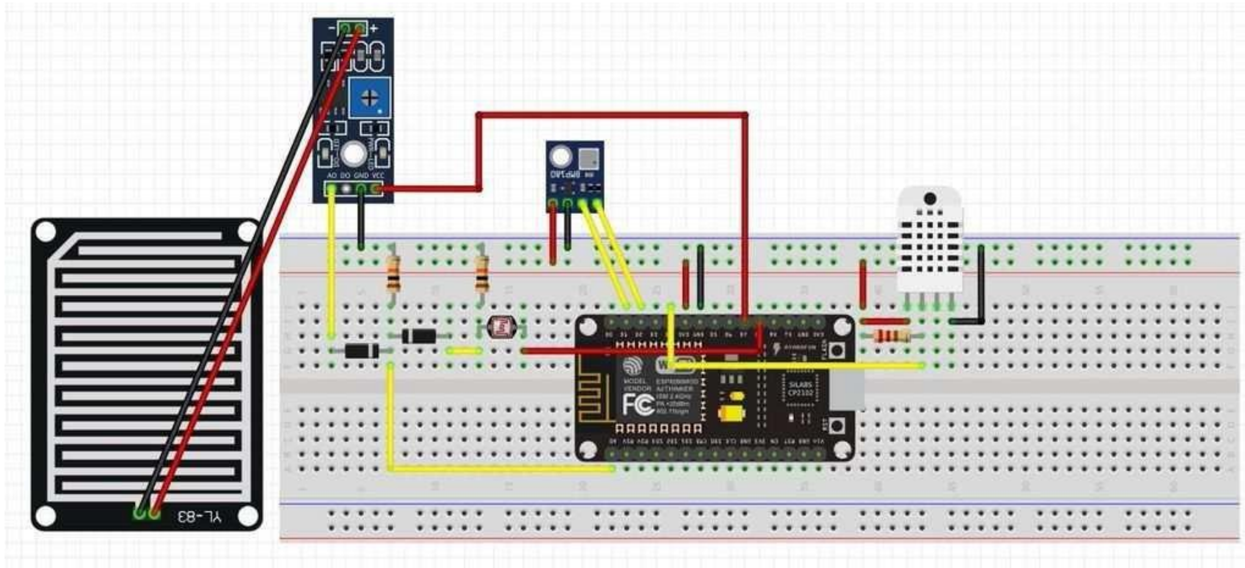
The BMP180 consists of a piezo-resistive sensor, an analog to digital converter and a control unit with E2PROM and a serial I2C interface. The BMP180 delivers the uncompensated value of pressure and temperature.The microcontroller sends a start sequence to start a pressure or temperature measurement. After converting time, the result value (pressure or temperature respectively) can be read via the I2C interface. For calculating temperature in °C and pressure in hPa, the calibration data has to be used. These constants can be read out from the BMP180 E2PROM via the I2C interface at software initialization.The sampling rate can be increased up to 128 samples per second (standard mode) for dynamic measurement. In this case, it is sufficient to measure the temperature only once per second and to use this value for all pressure measurements during the same period.



2.Approach

At first we thought of using Arduino as our base. But then we came up with the plan of using a wireless approach. In order to use a wireless approach (which would save our time and cost) we decided to use Nodemcu. We tried of using Nodemcu at first without the help of an app named ‘blynk’ and we got the output on the laptop itself using ‘serial monitor’ which is included in the Arduino IDE itself. After that we decided to try using mobile app. We connected the mobile’s hotspot to the nodemcu using the code and then we compiled the code into the
nodemcu. That’s how our approach went correct.





2.1 PROCEDURE:

1. Build a circuit as shown in the above image.
2. Connect the NodeMCU to the PC
3. Code using the ARDUINO IDE software.
4. With the proper library files installed, get the output of various sensors.
5. Coding is the most essential step in the whole project. We used the Arduino IDE Software to input our code to the nodemcu microcontroller.
6. We connected the NODEMCU with the laptop and the code was transferred to the board.

2.2 CODE:

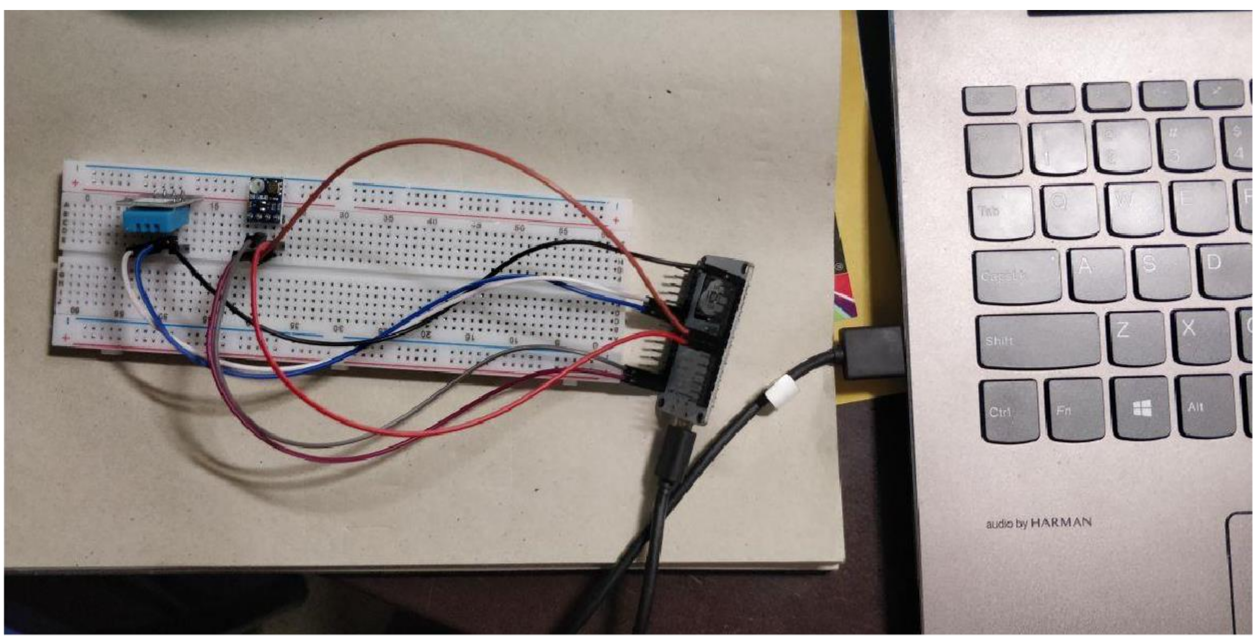
```
#define BLYNK_PRINT Serial

#include <SPI.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <DHT.h>

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "1c06a2f58aa3458da54b42cbecb8d61e";

// Your WiFi credentials.

// Set password to "" for open networks. char

ssid[] = "Warlock";

char pass[] = "warlock123";

#define DHTPIN D3

#define led       D2

int sensorThres = 75;
```

```
#define DHTTYPE DHT22

#define DHTTYPE DHT22 DHT
dht(DHTPIN, DHTTYPE);
SimpleTimer timer;

// This function sends Arduino's up time every second to Virtual Pin (5).
// In the app, Widget's reading frequency should be set to PUSH. This means
// that you define how often to send data to Blynk App. void
sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {
    Serial.println("sucessfully reading the data from DHT sensor!"); return;
  }
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
```

```
  Blynk.virtualWrite(V5, h);
  Blynk.virtualWrite(V6, t);
}

void setup()
{
  Serial.begin(9600); // See the connection status in Serial Monitor
  Blynk.begin(auth, ssid, pass);
  pinMode(led, OUTPUT);
  dht.begin();

  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}

void loop()
{
  Blynk.run(); // Initiates Blynk timer.run();
  // Initiates SimpleTimer
  int analogSensor = analogRead(DHTPIN);
 Serial.print("Pin D4: ");
 Serial.println(analogSensor);
```

```
if (analogSensor > sensorThres)
{
  tone(led, 100, 10);
}
else
{
  noTone(led);
}
delay(3500);


}
```

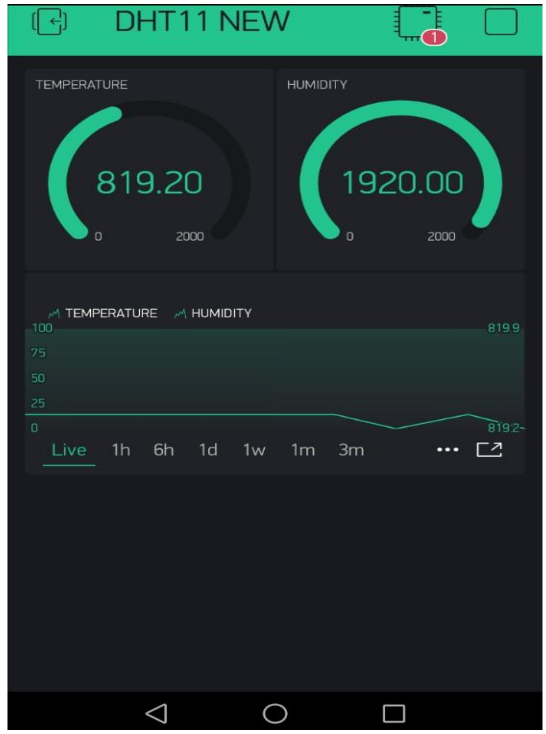
2.3 CIRCUIT:



3.RESULTS: We obtained the following results on the blynk app which is an app used in mobile. It basically acts a monitor.

We could lively track the temperature, pressure and humidity changes around us.

Once connected to the phone it shows the project is connected.



We can also plot graphs spontaneously and check statistics from the graphs. Everything will be shown clearly and spontaneously.

4.CONCLUSION:

The temperature sensor sensed the different temperature and gave out the output. And also Bmp180 did its job by showing the appropriate output.

These sensors can be very useful in implementing a real-big-life model of weather system

5.APPENDICES:

The proposed IoT based weather station can be modified to 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 location of the surrounding will also be mailed or messaged to the user along with the surrounding parameters,like, temperature, humidity, pressure, light intensity etc. It can also be modified 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 location will 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

OLED DISPLAY

6.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 International Conference on Microelectronics, Computing and Communications (MicroCom), Jan 2016, pp. 1–6.

4. H. Saini, A. Thakur, S. Ahuja, N. Sabharwal, and N. Kumar, “Arduino 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.
5. A. Lage and J. C. Correa, “Weather station with cellular communication network,” in 2015 XVI Workshop on Information Processing and Control (RPIC), Oct 2015, pp. 1–5.
6. T. Thaker, “Esp8266 based implementation of wireless sensor network with linux based web-server,” March 2016.

[7.] Y. Zhou, Q. Zhou, Q. Kong, and W. Cai, “Wireless temperature

amp; humidity monitor and control system,” in 2012 2nd International Conference on Consumer Electronics, Communications and Networks