DEVELOPMENT OF AN IOT-BASED WEATHER STATION

BY EEE310 GROUP 5 (FIVE)

GROUP MEMBERS

S/N	NAME	MATRIC. NUMBER
1.	OKUNOWO SIMILOLUWA ADETOYOSI	EEG/2016/095
2.	OGBODO ENE VICTORIA	EEG/2016/045
3.	AYOTUNDE IDOWU OLUWASEGUN	EEG/2016/115
4.	OLORUNTOBA VICTOR OLUWASEGUN	EEG/2016/055
5.	AKINLADE BABATUNDE KAYODE	EEG/2016/085
6.	KAYODE OLANREWAJU TIMOTHY	EEG/2016/035
7.	EZINMA CHINWEDUM PRECIOUS	EEG/2016/025
8.	OLAONIPEKUN LUKMON OLAKUNLE	EEG/2016/125

OCTOBER, 2019.

ABSTRACT

A weather station can be defined an instrument consisting of different sensors which is used to measure and monitor the weather and atmospheric conditions in a certain area. A typical weather station consists of temperature, humidity, barometric pressure sensors, rain gauges, anenometers, soil moisture sensors, light intensity sensors, solar radiation sensors e.t.c. In this report, we discuss the development of an IOT-Based Weather Station used to measure the temperature, humidity, and pressure of a certain environment where the weather conditions can be monitored from anywhere on the internet. The "heart" of this design is the low-cost NodeMCU ESP8266 WiFi-based microcontroller which is powered by a 3.3V power supply unit. The sensor nodes consist of a temperature and humidity sensor module (DHT11) and a barometric pressure sensor (BMP180). The readings obtained by the sensor are uploaded to the internet in real-time using the MQTT (Message Queue Telemetry Transport) application layer protocol. The MQTT protocol is used here because it is light-weight, faster and more reliable for resource constrained IOT devices. A smartphone or PC running an MQTT client is used to publish and subscribe to topics sent to the MQTT broker via the internet for effective monitoring from anywhere in the world.

Furthermore, from the temperature, humidity and pressure readings, We can also compute other weather parameters such as the sea-level pressure, altitude, dew point, fog, Heat index e.t.c. All the codes for the project were written in the Arduino IDE and uploaded to the NodeMCU microcontroller. A weather station can be very helpful in obtaining localized weather forecasts and data, which can help in planning our daily activities. It can also help farmers and agricultural producers to properly plan for planting and harvesting periods. The design can be further developed to incorporate more sensors and also use an OLED display to display real-time readings. A control section can also be incorporated in this weather station to perform control and automation purposes.

KEYWORDS :- Weather station, DHT11 (Temperature and humidity sensor), BMP180 (Pressure sensor), MQTT, IOT, NodeMCU, Arduino IDE.

CHAPTER ONE

INTRODUCTION

The principles of measurement and instrumentation play a very vital role in engineering. Measurement systems, instrumentation systems and the sensors and actuators used within them have made immense contributions to a variety of day-to-day domestic and industrial activities. The advent of IoT (**I**nternet **o**f **T**hings) in measurement and instrumentation systems has enabled an increasing number of devices to get connected to the internet. IoT is a giant network of connected things and people all of which share data uploaded to the cloud. Applications of IoT include Healthcare, wearable devices, Smart Grids, instrumentation, agriculture (smart farms) e.t.c. IoT helps us to monitor and control parameters and measured values from the sensors or actuators on the Internet from anywhere around the Globe via the very large internet.

A sensor is electronic transducer and it detects or responds to some kind of physical quantity from its surroundings and converts it to an electrical signal i.e voltage. These physical quantities could be temperature, humidity, colour, light intensity, pressure, rain, moisture, radiation gradient e.t.c. Actuators are also electronic transducers and they convert electrical energy to mechanical energy or a physical quantity. Some of the common actuators include Relays, solenoids, DC motors, thermistor e.t.c. Microcontrollers such as NodeMCU ESP8266, Raspberry Pi, Arduino are used to connect the sensors or actuators. The sensors and actuators are connected via the General Purpose Input/Output (GPIO) pins on the microcontroller or through a hardware data bus. Communication protocols such as the I2C and the SPI are used to connect these components to the bus. The NodeMCU ESP8266 is an open source IoT platform and is chosen for this project because of its low-cost and WiFi capability.

MQTT (**M**essage **Q**ueue **T**elemetry **T**ransport) is a simple Client server, publish/subscribe based messaging protocol used to create a communication layer between multiple devices via a Client and a Broker connection. The MQTT protocol is chosen over the conventional HTTP (**H**ypertext **T**ransfer **P**rotocol) due to its light-weight, speed, and suitability for resource constrained devices, and devices with very low bandwidth which makes it the best application layer protocol for IoT applications. The Broker is like the server which receives messages from the client, filters the messages and publishes the messages to other subscribed clients. The ‘Client’ refers to the sensors, smartphone e.t.c. The client subscribes to a ‘topic’ on the MQTT broker and the broker can be connected to multiple clients. When a client (i.e sensor) publishes a message (i.e temperature reading) to the MQTT broker, all other clients (i.e the smartphone) subscribed to that broker receive the message. The MQTT broker can also have multiple topics and also the clients can connect to multiple topics from a single MQTT broker. This report illustrates how a IoT based weather station is developed using the NodeMCU ESP8266 Microcontroller, MQTT protocol and a temperature, pressure and humidity sensor.

CHAPTER TWO

METHODOLOGY

BRIEF OVERVIEW OF THE COMPONENTS USED

(i) NodeMCU ESP8266 Microcontroller:-

The ESP8266 NodeMCU serves as the ‘heart’ of this project.The DHT11 and BMP180 sensors used are connected to the NodeMCU via the GPIO (General purpose Input/ Output) pins and the NodeMCU also provides the Vcc (3.3v) and GND (Ground) for the sensors and other components connected to it. The NodeMCU development board also serves as the IoT platform as the sensors send the measured values to it and it uploads this values to the cloud based on the WiFi connectivity.

The ESP8266 NodeMCU has a total of 17 GPIO pins all serving different peripheral and serial communication purposes. These pins include :-

1 ADC (Analog to Digital Converter) channel – 10 bits

- UART (Universal Asynchronous Receiver/Transmitter) – Used to load the code serially
- PWM pins – For dimming LEDs or for controlling motors
- SPI, I2C and I2S interfaces – Used to connect sensors and all other peripherals

It also has an on-board voltage regulator of 3.3 V and reliably supplies 600 Ma. It has a flash memory of 4 MB and an SRAM of 128 Kilobytes. Other features include on-board switches, LEDs, FLASH and RESET buttons.

(ii) DHT11 :-

The DHT11 is a 4-pin dual sensor module which can be used to measure the absolute temperature and humidity of a surrounding and gives a calibrated digital output. The DHT11 sensor has a resistive-type humidity measurement component and an NTC type temperature measurement component with an inbuilt 8-bit microcontroller. The 4 pins of the DHT11 are the Vcc (pin 1), Data Pin (pin 2), and the GND (pin 4). The Vcc and GND pins are connected to the 3.3V and GND pins of the NodeMCU collectively. The pin 2 on the other hand is connected to the D4 of the NodeMCU.

(iii) **BMP180** :-

The BMP180 sensor is a barometric pressure sensor used to measure the absolute pressure of an area based on the weather and altitude of the area.The BMP180 sensor works using an I2C interface and also has 4 pins. The 4 PINS are the Vcc, GND, SCL, SDA. The SCL of the BMP180 is connected to the SCL pin of the NodeMCU which is pin D1 and the SDA pin of the BMP180 is connected to the SDA pin of the NodeMCU which is pin D2. The Vcc and GND are connected to the Vcc and GND of the NodeMCU respectively. The BMP180 sensor can also be used to measure temperature.

STATIC INSTRUMENT CHARACTERISTICS OF THE SENSORS USED

	DHT11 (Temperature)	**DHT11** (Humidity)	**BMP180** (Pressure)
RANGE	0 ◦C - 50 ◦C	20 – 90 % RH	300 – 1100 hPa
ACCURACY	±2 ◦C	±4% RH (Max ±5% RH)	± 0.12 hPa
SENSITIVITY	0.1 ◦C	1 % RH	0.01 hPa
REPEATABILITY	± 1 ◦C	± 1 % RH	± 0.1 hPa
RESOLUTION	16 bits	16 bits	16 bits
HYSTERESIS	-	±1 % RH	-

The NodeMCU is programmed using the Arduino IDE and the corresponding libraries for the sensors and MQTT client were downloaded and also used in the code. The topics for the readings from the sensors are published to the open-source MQTT broker called (broker.mqttdashboard.com (port 1883)) ,once there is internet connection for the NodeMCU. An MQTT client is now created on an Android smartphone to subscribe to those topics for monitoring from anywhere around the world once the username and password for the MQTT broker are logged into. The block diagram for this simple system is illustrated below :-

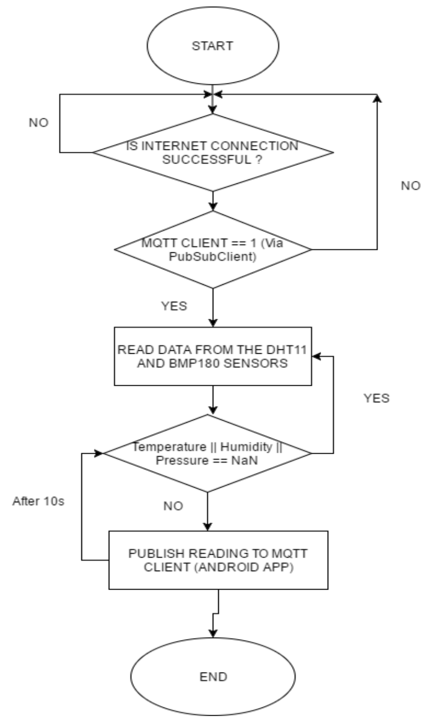


Figure 1 :- BLOCK DIAGRAM OF THE ENTIRE SYSTEM

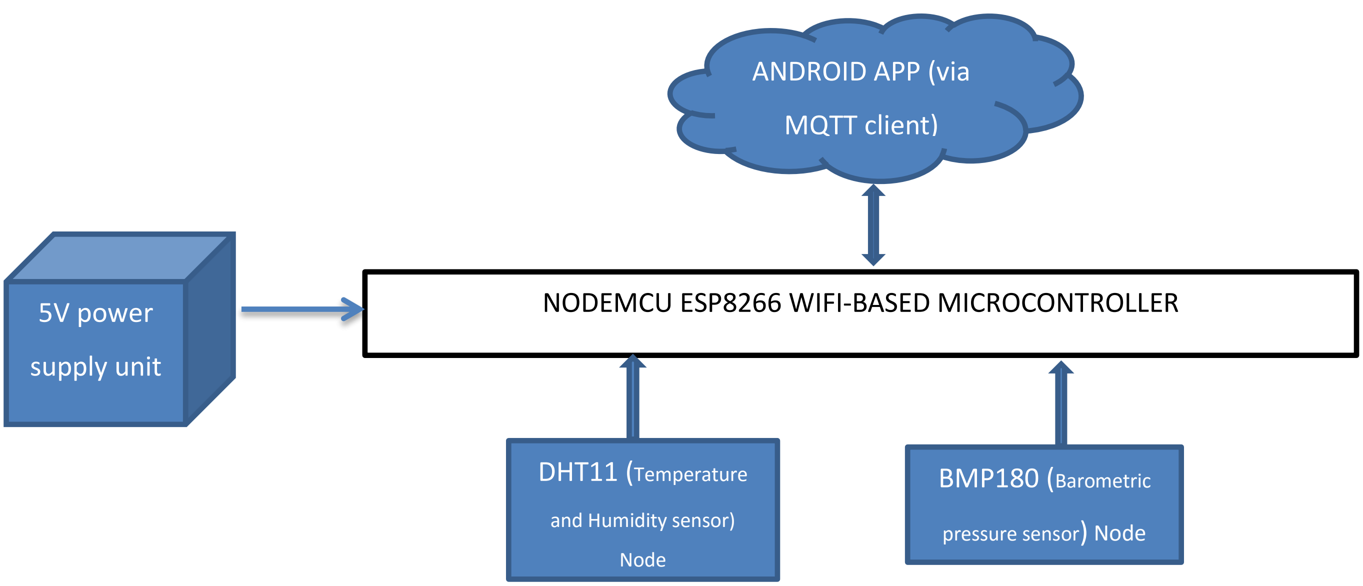


Figure 2 :- BLOCK DIAGRAM SHING ALL THE NODES IN THE CIRCUIT

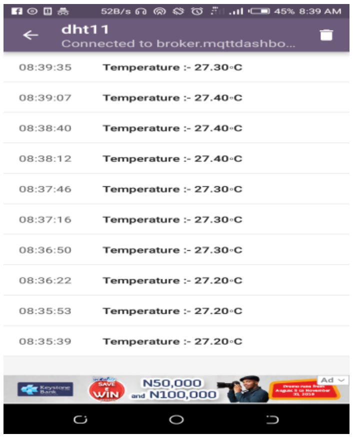
CALIBRATING THE DHT11 AND BMP180 SENSOR

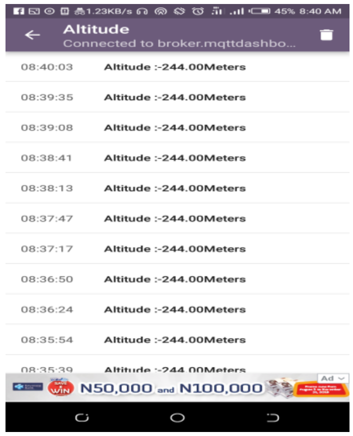
Calibration is a very essential aspect of measurement and instrumentation. It basically involves 'comparison' between the output of an instrument (The sensors in this case) to the output of a pre-defined standard when the same input or measurand is applied to it. It is very important to ensure uniformity and standard accuracy of an instrument and also to prevent systematic errors. The two sensors used in this project are already factory calibrated . The temperature monitoring part of the DHT11 is calibrated using a standard Thermometer and the humidity part is calibrated using a humidity measuring device to obtain the actual humidity of the surrounding. The DHT11 temperature range is from 0 to 50 ◦C with an accuracy of ±2 ◦C and, humidity range is from 20% to 90% both with a good resolution (16 bits ADC). Unfortunately, the DHT sensors do not support calibrating of features but the accuracy and repeatability of the DHT11 sensor when tested was not really bad.

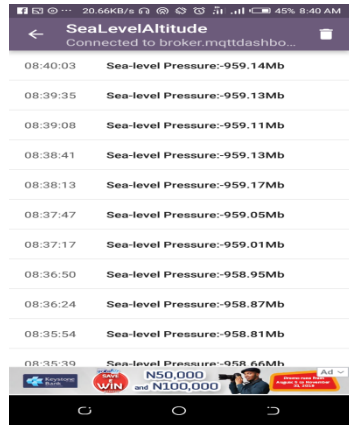
On the other hand, The BMP180 sensor can be calibrated using a polynomial approach to obtain the optimal calibration parameters based on the RMSE (Root mean squared error). This was not done however due to the absence of a standard barometric pressure measuring instrument.

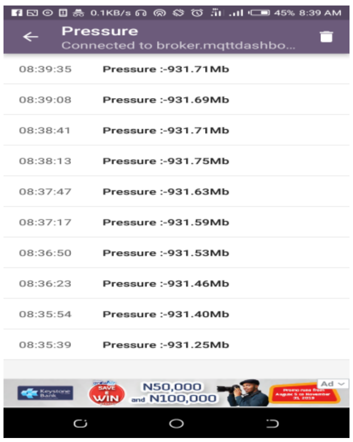
RESULTS AND DISCUSSIONS

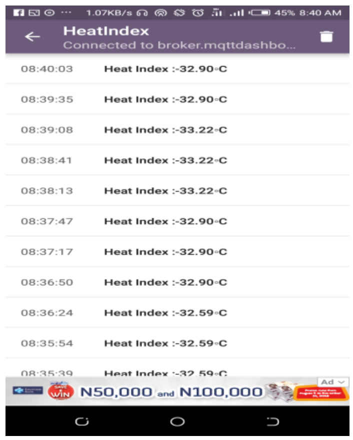
Once the setup was completed, the results of the readings were published to the MQQT broker at broker.mqtt-dashboard.com (Port 1883) . An Android App running an MQTT client published or subscribed to the right topics as specified in the code and connected to the internet can be used to show the real-time readings from the sensors. The following test was carried out on in a room in Awolowo Hall and pictures of the readings are shown below :-

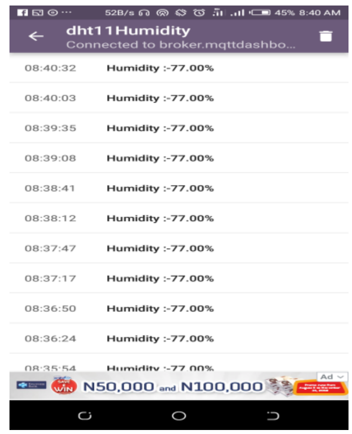












COMPUTING THE MEAN, STANDARD DEVIATION, VARIANCE

The mean, standard deviation and variance were calculated from the readings obtained to reduce Random errors in the readings. This was done using Data Analysis tools in Python programming language :-

WEATHER STATION RESULTS

WEATHER STATION STATISTICAL ANALYSIS (std- Standard Deviation, Variance = std^2)

```
weather_station_results
```

	Temperature (oC)	Humidity (%)	Pressure (mB)	HeatIndex (oC)	SeaLevelPressure (mB)
0	27.3	77	931.71	32.90	959.14
1	27.4	77	931.69	32.90	959.13
2	27.4	77	931.71	33.22	959.11
3	27.4	77	931.75	33.22	959.13
4	27.3	77	931.63	33.22	959.17
5	27.3	77	931.59	32.90	959.05
6	27.3	77	931.53	32.90	959.01
7	27.2	77	931.46	32.90	958.95
8	27.2	77	931.40	32.59	958.87
9	27.2	77	931.25	32.59	958.81

```
weather_station_results.describe()
```

	Temperature (oC)	Humidity (%)	Pressure (mB)	HeatIndex (oC)	SeaLevelPressure (mB)
count	10.00000	10.0	10.000000	10.000000	10.000000
mean	27.30000	77.0	931.572000	32.934000	959.037000
std	0.08165	0.0	0.161576	0.232818	0.124191
min	27.20000	77.0	931.250000	32.590000	958.810000
25%	27.22500	77.0	931.477500	32.900000	958.965000
50%	27.30000	77.0	931.610000	32.900000	959.080000
75%	27.37500	77.0	931.705000	33.140000	959.130000
max	27.40000	77.0	931.750000	33.220000	959.170000

CONCLUSION AND RECOMMENDATION

This project basically involves the design and development of an IoT Local Weather station for monitoring several weather parameters in a specific area and sending the real-time readings to the Internet using low-cost microcontrollers and sensors. We however faced some slight challenges with the recalibration of the sensors used and also with the accuracy of the readings. This design can however be improved to include some other features such as :-

- An OLED for display in the cases of no internet
- A GPS module to show the location of the area
- More sensors i.e Rain, Light intensity, Moisture to make the design more robust
- A control unit – to control a switch when any of the values goes above a specific threshold

This weather station can be useful in many applications i.e

- Farming – to help farmers plan their planting and harvesting periods
- Weather forecasting