IOT BASED ANEMOMETER

A PROJECT REPORT (EIF 452)

Submitted by

P.V.N. S. SATHWIK	**(15127001)**
D.MAHESH	**(15127064)**
D.VISHNU VARDHAN	**(15127066)**
MANNURU AJAY KUMAR	**(15127146)**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

MECHANICAL ENGINEERING

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

DEPARTMENT OF MECHANICAL ENGINEERING

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

PADUR, CHENNAI - 603 103

NOVEMEBER 2018

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

BONAFIDE CERTIFICATE

This is to certify that this project report **"IOT BASED ANEMOMETER"** is the bonafide work of **"P V N S SATHWIK (15127001), D MAHESH (15127064), D VISHNU VARDHAN (15127066), MANNURU AJAY KUMAR (15127146)"** who carried out the project work as a part of the subject **"EIF 452 – IoT for Elecrical Engineers"** under my supervision during the academic year 2018-2019.

SIGNATURE
Dr. G. Muthukumaran
SUPERVISOR
Dept. of EIE,
Hindustan Institute of Technology & Science,
No. 1, Rajiv Gandhi Salai (OMR),
Padur,
Chennai - 603103

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Name: ________________________ Name: ________________________

ABSTRACT

Many wind speed measurement projects report poor performance as low as 60% of the predicted performance. The reason for this is poor resource assessment and the use of new untested technologies and systems in remote locations. Predictions about the potential of an area for wind speed measurement projects (through simulated models) may vary from the actual potential of the area. Hence, introducing accurate site assessment techniques will lead to accurate predictions of energy production from a particular area. We solve this problem by using an INTERNET OF THINGS (IOT) technology to periodically analyze the data from anemometers installed in that area. In this project we are developing a prototype of this system. The anemometer blades rotate with the wind speed and direction. The voltage generated during the anemometer blades rotation will be converted into the wind speed. The data will be continuously updated in the cloud. This data is further calculated to give the average, minimum and maximum speed of the wind in that particular area.

ACKNOWLEDGEMENT

First and foremost, we would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

We would like to thank our internal guide **DR. MUTHUKUMARAN**, Center for Sensors for continually guiding and actively participating in our project, giving valuable suggestions to complete the project work.

We wish to express our heartfelt gratitude to **Dr. G. RAVI KUMAR SOLOMON**, Head, Department of Mechanical Engineering for much of his valuable support encouragement in carrying out this work.

It's a matter of pride and privilege for us to express our deep gratitude to the management of HITS for providing us the necessary facilities and support.

We are highly elated in expressing our sincere and abundant respect to the Vice Chancellor Prof. **(Dr.) KUNCHERIA P. ISSAC** for giving me this opportunity to bring out and implement our ideas in this project.

We would like to thank all the technical and teaching staff of the Electronics and Instrumentation Sciences, who extended directly or indirectly all support.

Last, but not the least, we are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION:

Wind isn't something you can see very easily, so you can hardly time it with a stopwatch like you'd measure the speed of an Olympic sprinter or a race car! Fortunately, scientists are amazingly inventive people and they've come up with some pretty clever ways of measuring wind speed with gadgets called anemometers. The invention of anemometer helped man to measure wind speed. But being a weather station equipment the data speaks about what to forecast. There are many primitive techniques to do this but there is no proper tracking of the wind speed and it even requires operators and maintenance people.

Now with the help of IOT we would like to contribute for weather data tracking. We are interfacing the microcontroller, sensors and the internet to track the wind speed continuously which will reduce the man time and accurate results are obtained periodically as the tracking is done continuously.

1.2 ANEMOMETER:

An anemometer is a device used for measuring wind speed, and is also a common weather station instrument. The term is derived from the Greek word anemos, which means wind, and is used to describe any wind speed instrument used in meteorology. The first known description of an anemometer was given by Leon Battista Alberti in 1450. [1] The anemometer has changed little since its development in the 15th century. Leon Battista Alberti (1404–1472) is said to have invented the first mechanical anemometer around 1450. In following centuries, numerous others, including Robert Hooke (1635–1703), developed their own versions, with some being mistakenly credited as the inventor.

Fig.1.1 Anemometer

1.3 WIND SPEED MEASUREMENT:

Wind is produced by the conversion of atmosphere ‘s potential energy to kinetic energy which sets the air in the atmosphere in motion, generating wind. Wind energy is a by-product of solar energy. The atmospheric layer that is of most important to wind energy applications extends to around 100 m over the level of ground and here the wind is primarily affected by surface friction. This atmospheric layer where the surface (roughness) plays a major role is known as the surface layer. Wind can be defined as a vector quantity which has direction and speed to its attributes. In accord to meteorological convention, wind is the horizontal component of air motion.

A wind measurement provides data to improve wind resource assessment and to increase confidence in site evaluation. Thus in order to find out the feasibility of site for taking up wind related projects the information about meteorological data especially wind data along with other environmental data and it analysis is important. They are the following parameters:

Parameters which define measurement

- Type and quality of equipment
- Measurement levels of sensors
- Accuracy of Measurement, it’s duration and data recovery
- Sampling of data and intervals in which it is recorded
- Storage format for data
- Processing procedures applied to data
- Reports on data

CHAPTER 2

LITERATURE SURVEY

2.1 HISTORY:

NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core,[citation needed] widely used in IoT applications (see related projects). NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9.[12] Later that month, Tuan PM ported MQTT client library from Contiki to the ESP8266 SoC platform,[13] and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib[14] to NodeMCU project,[15] enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.

A research article mentioning the Internet of things was submitted to the conference for Nordic Researchers in Logistics, Norway, in June 2002, [24] which was preceded by an article published in Finnish in January 2002. [25] The implementation described there was developed by Kary Främling and his team at Helsinki University of Technology and more closely matches the modern one, i.e. an information system infrastructure for implementing smart, connected objects. [26]

The anemometer has changed little since its development in the 15th century. Leon Battista Alberti (1404–1472) is said to have invented the first mechanical anemometer around 1450. In following centuries, numerous others, including Robert Hooke (1635–1703), developed their own versions, with some being mistakenly credited as the inventor. In 1846, John Thomas Romney Robinson (1792–1882) improved upon the design by using four hemispherical cups and mechanical wheels. In 1926, Canadian meteorologist John Patterson (January 3, 1872 – February 22, 1956) developed a three-cup anemometer, which was improved by Brevoort and Joiner in 1935. In 1991, Derek Weston added the ability to measure wind direction. In 1994, Andrews Pflitsch developed the sonic anemometer. [1]

2.2 PREVIOUS WORK:

With the reference to the project 'Design and Construction of a Digital Anemometer' by iaz Ahmad1, Hari Krishna Kumar Sah, Md. Saddam Hossain1, Mohammad Ariful Islam Undergraduate Student, Department of Mechanical Engineering, our model is miniature version of this which directly gives us the data just by simply using the Anemometer is a measuring device used for measuring wind speed. Since the beginning of its invention, it has been modified several times. Many types of new generation of Anemometers are available nowadays. But these anemometers complicated and very high in price. Our objective was to develop an anemometer of simple design and construction with ease of use. In this paper, design, construction and testing of a three cup anemometer has been described. A three cup anemometer was designed as three cup anemometers are simpler and yet has less errors. Microcontroller was used to make it digitalized. It was tasted with a standard anemometer to perform necessary calibration. After calibration, it was tested again on various wind speeds and the errors were found to be 3.04% and the minimum air that it can measure is 3km/hour.[27]

They have used a magnetic reed switch and the similar cup anemometers. They have used Arduino Uno as the microcontroller where as we have used NodeMcu.

CHAPTER 3

HARDWARE AND SOFTWARE COMPONENTS

3.1 HARDWARE COMPONENTS:

This IOT based anemometer is made by interfacing both hardware components and software components. The hardware components used are a microcontroller, Dynamo and the anemometer blades.

3.1.1 Microcontroller: (NodeMcu ESP -12E):

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the DevKit. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.

Features:

- Version: DevKit v1.0.
- Breadboard Friendly.
- Light Weight and small size.
- 3.3V operated, can be USB powered.
- Uses wireless protocol 802.11b/g/n.
- Built-in wireless connectivity capabilities.
- Built-in PCB antenna on the ESP-12E chip.
- Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.
- Uses CP2102 USB Serial Communication interface module.
- Arduino IDE compatible (extension board manager required).
- Supports Lua (alike node.js) and Arduino C programming language.

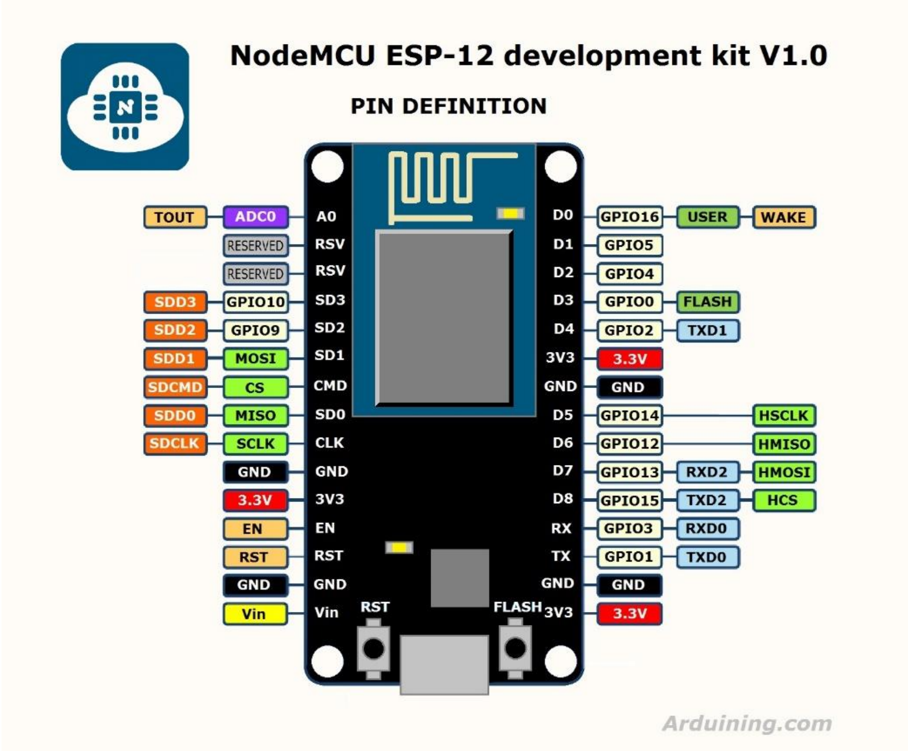

Fig.3.1 NodeMcu Pin configuration

Wireless Standard	IEEE 802.11 b/g/n
Frequency Range	2.412 - 2.484 GHz
Power Transmission	802.11b : +16 ± 2 dBm (at 11 Mbps) 802.11g : +14 ± 2 dBm (at 54 Mbps) 802.11n : +13 ± 2 dBM (at HT20, MCS7)
Receiving Sensitivity	802.11b : -93 dBm (at 11 Mbps, CCK) 802.11g : -85 dBm (at 54 Mbps, OFDM) 802.11n : -82 dBm (at HT20, MCS7)
Wireless Form	On-board PCB Antenna
IO Capability	UART, I2C, PWM, GPIO, 1 ADC
Electrical Characteristic	3.3 V Operated 15 mA output current per GPIO pin 12 - 200 mA working current Less than 200 uA standby current
Operating Temperature	-40 to +125 ℃
Serial Transmission	110 - 921600 bps, TCP Client 5

Wireless Network Type	STA / AP / STA + AP
Security Type	WEP / WPA-PSK / WPA2-PSK
Encryption Type	WEP64 / WEP128 / TKIP / AES
Firmware Upgrade	Local Serial Port, OTA Remote Upgrade
Network Protocol	IPv4, TCP / UDP / FTP / HTTP
User Configuration	AT + Order Set, Web Android / iOS, Smart Link APP

Table.3.1 Specifications of ESP-12E WiFi Module

3.1.2 Dynamo:

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.

Working Principle of DC Motor:

A machine that converts DC power into mechanical power is known as a DC motor. Its operation is based on the principle that when a current carrying conductor is placed in a magnetic field, the conductor experiences a mechanical force. The direction of this force is given by Fleming's left hand rule and magnitude is given by;

$$F = BIL \text{ Newton}$$

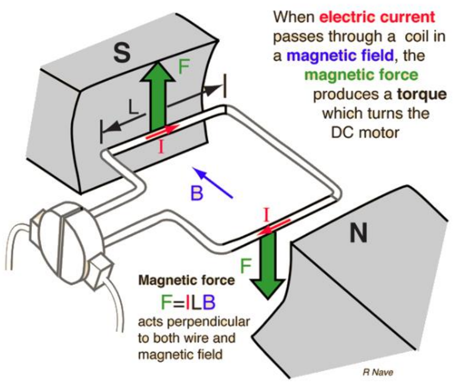

Fig.3.2 Dc Motor Principle

Motor Specifications:

- Standard 130 Type DC motor
- Operating Voltage: 4.5V to 9V
- Recommended/Rated Voltage: 6V
- Current at No load: 70mA (max)
- No-load Speed: 9000 rpm
- Loaded current: 250mA (approx)
- Rated Load: 10g*cm
- Motor Size: 27.5mm x 20mm x 15mm
- Weight: 17 grams

Fig.3.3 Dc Motor 5v

3.1.3 Cup type Anemometer Blades:

A simple type of anemometer was invented in 1845 by Dr. John Thomas Romney Robinson, of Armagh Observatory. It consisted of four hemispherical cups mounted on horizontal arms, which were mounted on a vertical shaft. The air flow past the cups in any horizontal direction turned the shaft at a rate that was roughly proportional to the wind speed. Therefore, counting the turns of the shaft over a set time interval produced a value proportional to the average wind speed for a wide range of speeds. It is also called a rotational anemometer.

Fig.3.4 Cup type Anemometer

3.2 SOFTWARE COMPONENTS:

The software components used for the project are Arduino IDE for writing and uploading the code to the NodeMcu. The ThingSpeak cloud platform for plotting the data.

3.2.1 Arduino IDE:

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board. [13]

The source code for the IDE is released under the GNU General Public License, version 2. [3] The Arduino IDE supports the languages C and C++ using special rules of code structuring. [14] The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.[15] The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.[16]

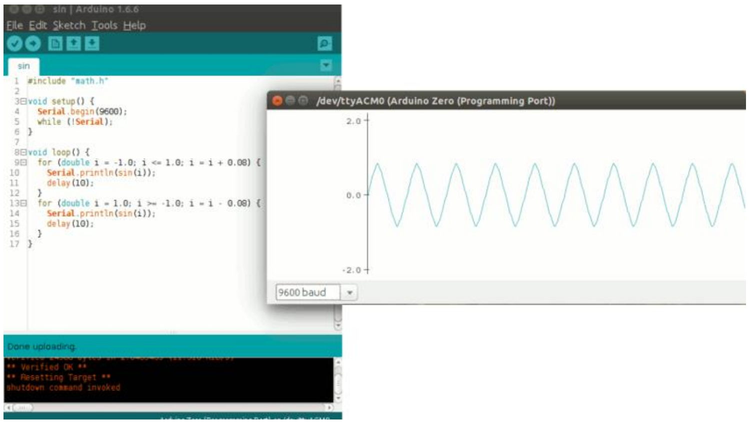

Fig.3.5 Arduino IDE

3.2.2 ThingSpeak:

According to its developers, "ThingSpeak is an open source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates”. [17]

ThingSpeak has integrated support from the numerical computing software MATLAB from MathWorks, [18] allowing ThingSpeak users to analyze and visualize uploaded data using Matlab without requiring the purchase of a Matlab license from Mathworks.

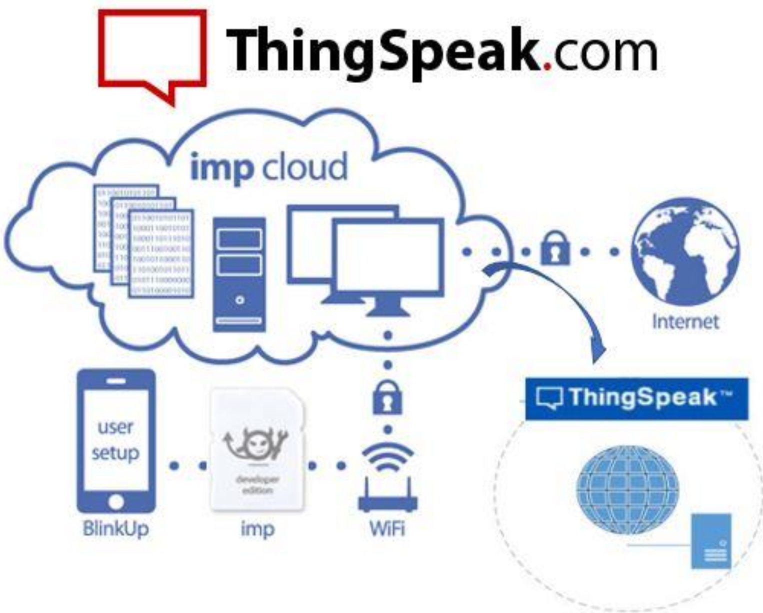

Fig.3.6 ThingSpeak Block diagram

CHAPTER 4

ANEMOMETER AND IOT INTERFACING

4.1 INTERNET OF THINGS (IOT):

The Internet of things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these things to connect, collect and exchange data. [19] [20] [21] [22]

IoT involves extending Internet connectivity beyond standard devices, such as desktops, laptops, smartphones and tablets, to any range of traditionally dumb or non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the Internet, and they can be remotely monitored and controlled. With the arrival of driverless vehicles, a branch of IoT, i.e. the Internet of Vehicles starts to gain more attention. [23]

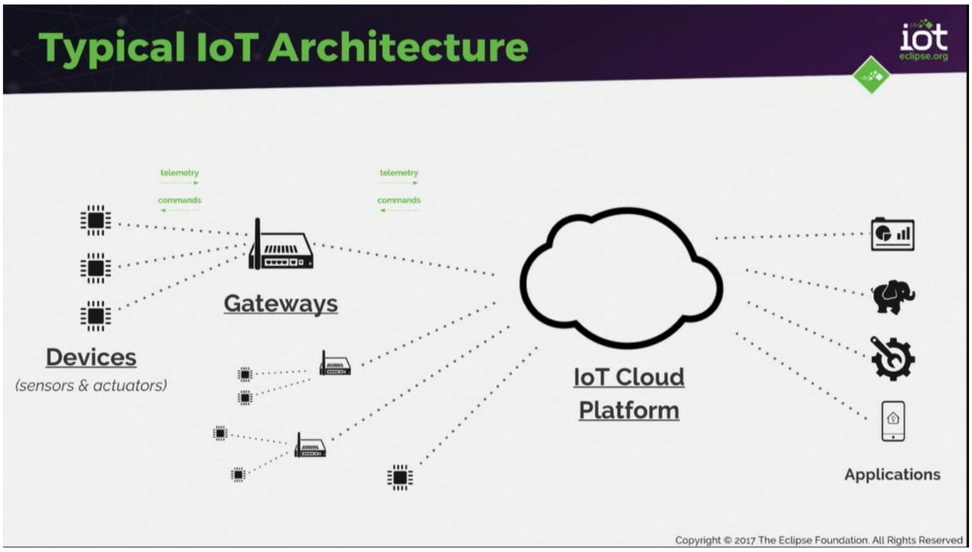

Fig.4.1 IOT Architecture

4.2 WORKING OF IOT BASED ANEMOMETER:

The wind on striking the cup type Anemometer blades rotates the shaft of the Dynamo. The mechanical work done by the wind by rotating the shaft of the Dynamo is converted into electrical energy. This is the principle of Dynamo. The voltage output from the dynamo is given as an input to the NodeMcu microcontroller which will convert the Voltage into the wind speed. This wind speed is sent to the ThingSpeak cloud platform by the NodeMcu built in Wi-Fi module. This data can be viewed by logging into the ThingSpeak Cloud.

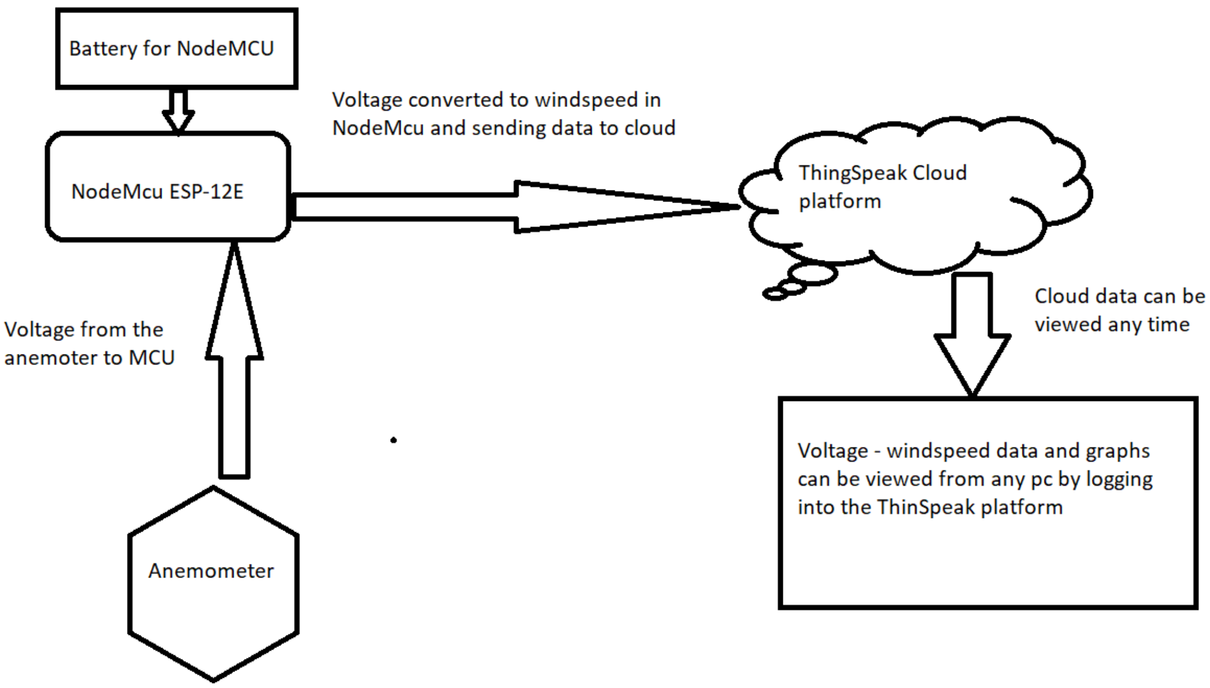

Fig.4.2 Block Diagram of IOT based Anemometer

4.3 NodeMcu Code:

```
#include <ESP8266WiFi.h>
String apiKey = "S83GPH1VKLDWANRE";     //  Enter your Write API key from
ThingSpeak
const char *ssid =  "vishnu";     // replace with your wifi ssid and wpa2 key
const char *pass =  "dadi1234";
const char* server = "api.thingspeak.com";
int ledPin = 9;
WiFiClient client;
void setup() {
 Serial.begin(9600);

       Serial.println("Connecting to ");
       Serial.println(ssid);
```

```
WiFi.begin(ssid, pass);
 while (WiFi.status() != WL_CONNECTED)
    { delay(500);
          Serial.print(".");
    }
     Serial.println("");
     Serial.println("WiFi connected");
     }
void loop() {
 float sensorvalue = analogRead(A0);
 float k= (sensorvalue*(51.0/1023.0)*50);
  if (client.connect(server,80))   //   "184.106.153.149" or api.thingspeak.com
                    {      String postStr = apiKey;
                          postStr +="&field1=";
                          postStr += String(sensorvalue);
                          postStr +="&field2=";
                          postStr += String(k);
                          postStr += "\r\n\r\n";
                          client.print("POST /update HTTP/1.1\n");
                          client.print("Host: api.thingspeak.com\n");
                          client.print("Connection: close\n");
                          client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
                          client.print("Content-Type: application/x-www-form-urlencoded\n");
                          client.print("Content-Length: ");
                          client.print(postStr.length());
                          client.print("\n\n");
                          client.print(postStr);
                           Serial.print("voltage: ");
                          Serial.print(sensorvalue);
                          Serial.print(" wind speed ");
```

```
                Serial.print(k);
                Serial.println("%. Send to Thingspeak.");
            }
      client.stop();

      Serial.println("Waiting...");

  // thingspeak needs minimum 15 sec delay between updates, i've set it to 30 seconds
  delay(10000);
}
```

code-*-

4.4 ThingSpeak NodeMcu Interfacing:

ThingSpeak provides instant visualizations of data posted by ESP8266 to ThingSpeak. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics. Uploading of ESP8266 sensor data is done using Internet. It is three step process.

- Connect to your WiFi hot spot having internet access.
- Read Sensor data
- Upload data to ThingSpeak

<u>Step 1</u>: Sign up ThingSpeak

It's simple just enter your email id and verify your account.

<u>Step 2</u>: Configuring ThingSpeak

Configuration is just few clicks job.

<u>Step 2.1</u>: Create New Channel

Click on New Channel

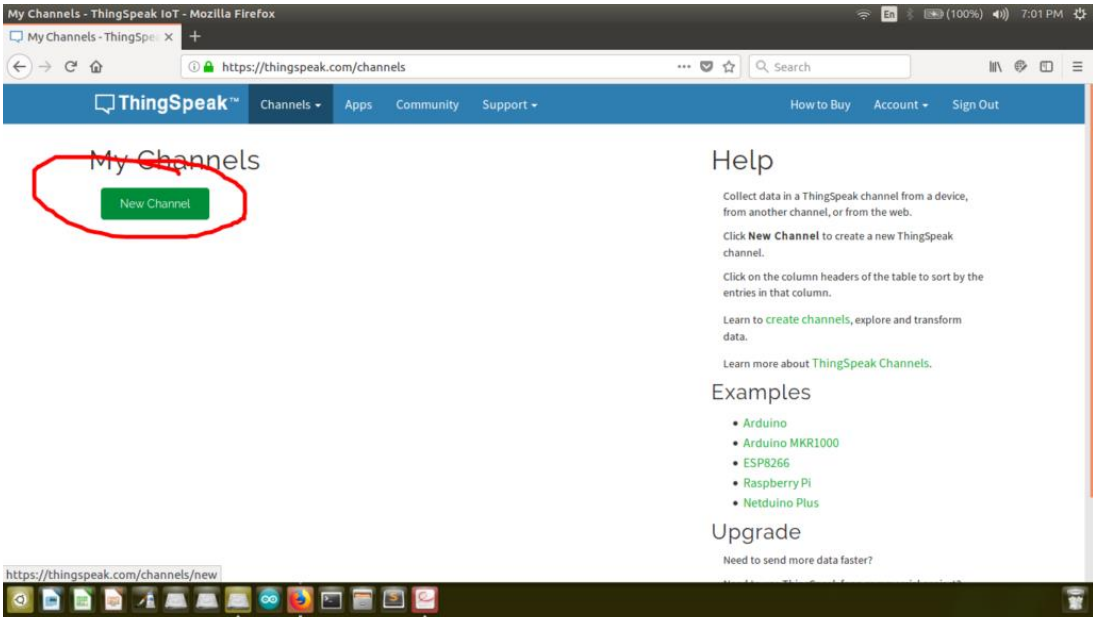

Fig.4.3 Step 2.1

Enter Name and Field. You may have multiple Fields depending on number of sensor create multiple fields such as Light, Temperature, Humidity, etc.

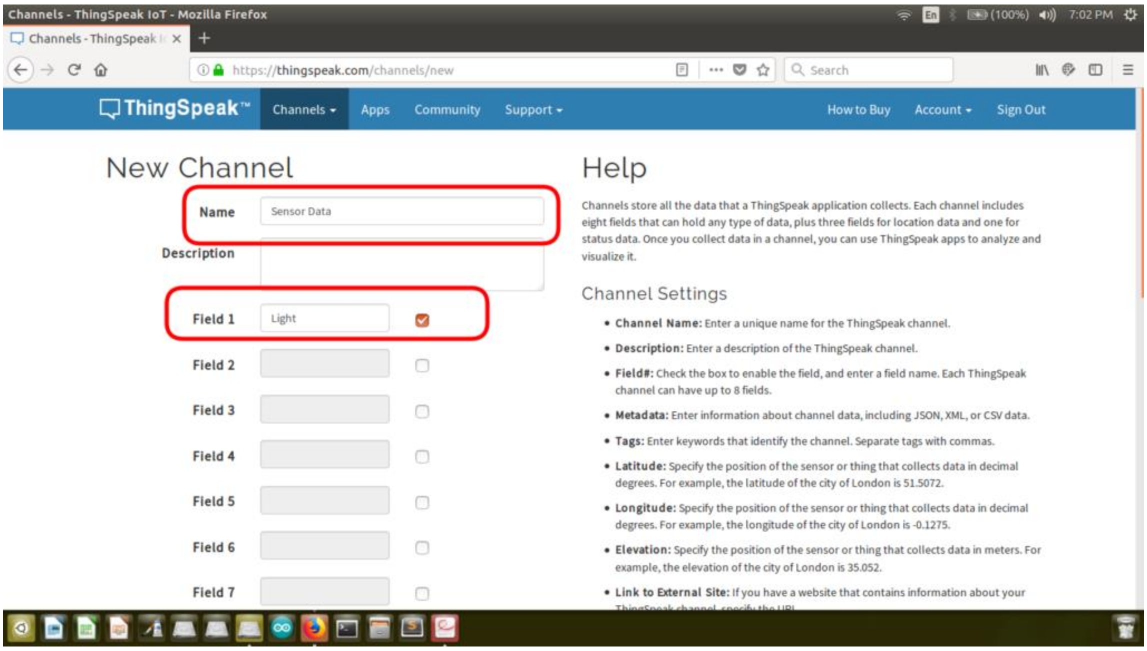

Fig.4.4 Step 2.2

Keep everything else as it is. Blank or default values. and click on Save Channel.

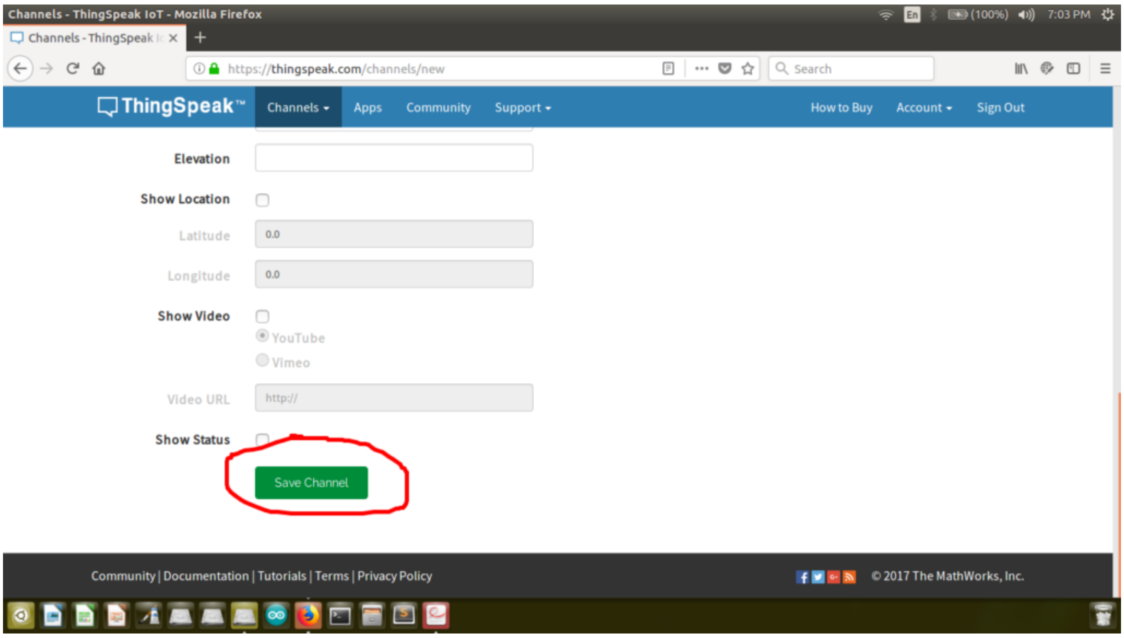

Fig.4.5 ThingSpeak Steps

Step 2.2: Getting API Key

Click on API Key Tab and look for these two fields Write Api Key and Update channel feed line.

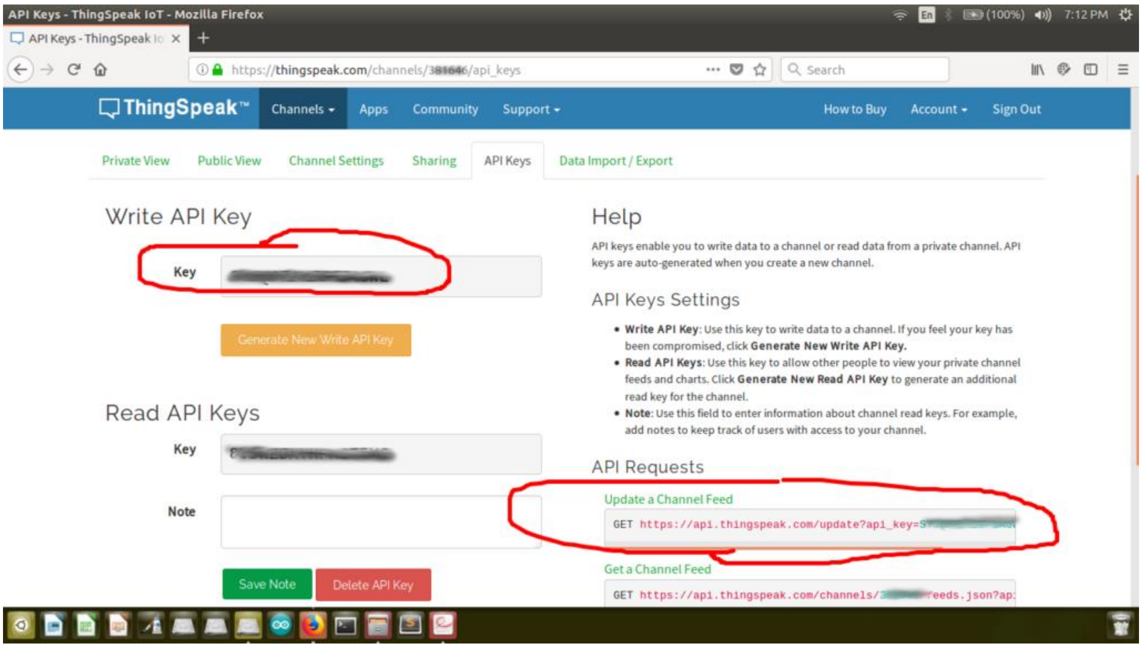

Fig.4.6 Step 2.2

This line is important for data upload to cloud server:

```
GET <span class="str">https://api.thingspeak.com/update?api_key=<span
class="customcode">akjshfajkhfowei</span>&amp;field1=<span
class="customcode">0</span></span>hhhhh
```

First parameter is Api Key. Do not share API key it makes decision of which user it is. In most cases I create own cloud server as per requirements. Second parameter is field1=0 here you can pass the ADC value ex. field1=1234

value in front of field1=0 is your sensor data for example if your adc value is 1234 then you call this GET request with https://api.thingspeak.com/update?api_key=yourapikey&field1=1234

Step 3: Check Data on ThingSpeak Server

Open Your ThingSpeak Account and Click on Private View of Your Channel.

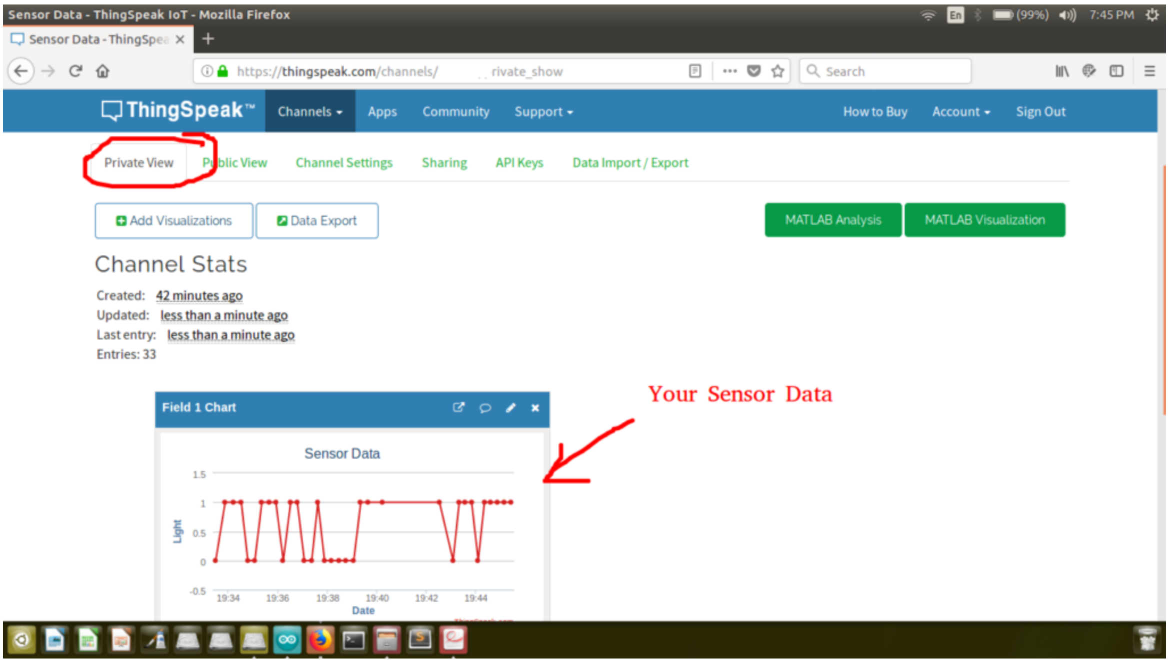

Fig.4.7 Step 3

CHAPTER 5

RESULTS AND CONCLUSION

5.1 RESULTS:

The serial monitor output while the code is running i.e.; when NodeMcu is sending the wind speed data to the cloud is shown below. Following screenshots were taken during the code execution:

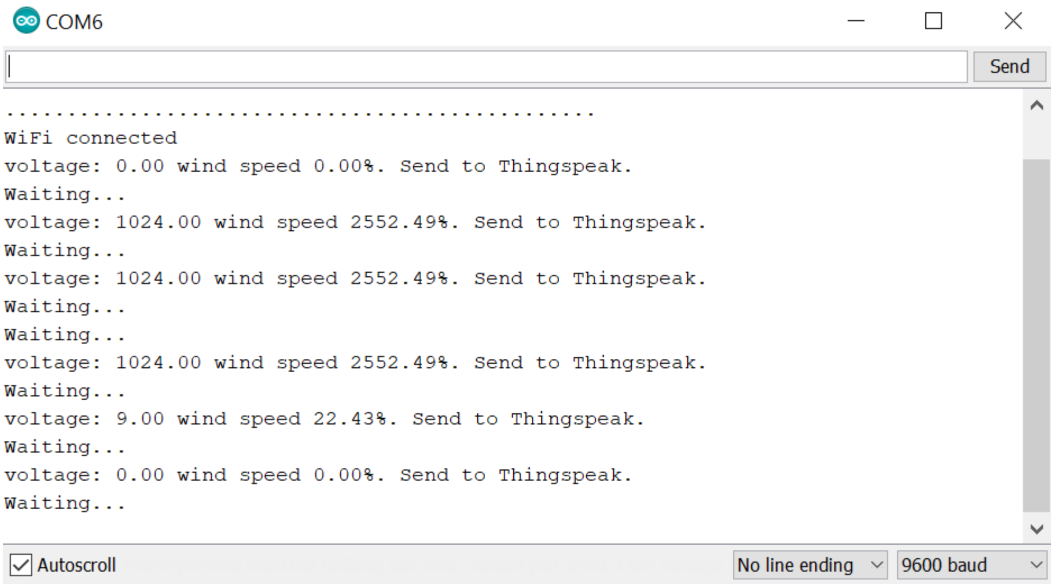

Fig.5.1 Serial Monitor Output

The data being stored in the ThingSpeak is plotted with the matlab plot options in the ThingSpeak. The following screen shots are obtained in the ThingSpeak platform.

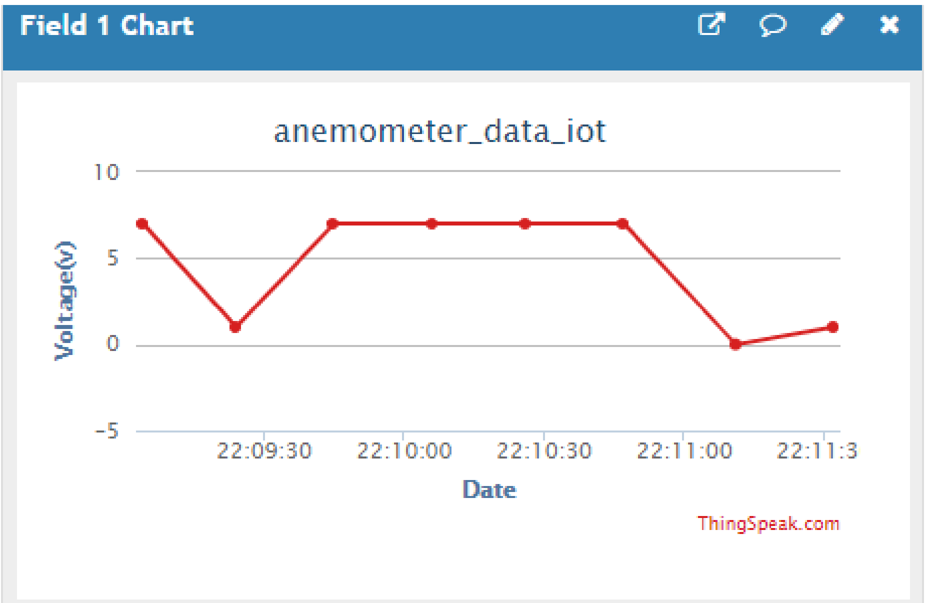

Fig.5.2 Voltage Vs Date plot

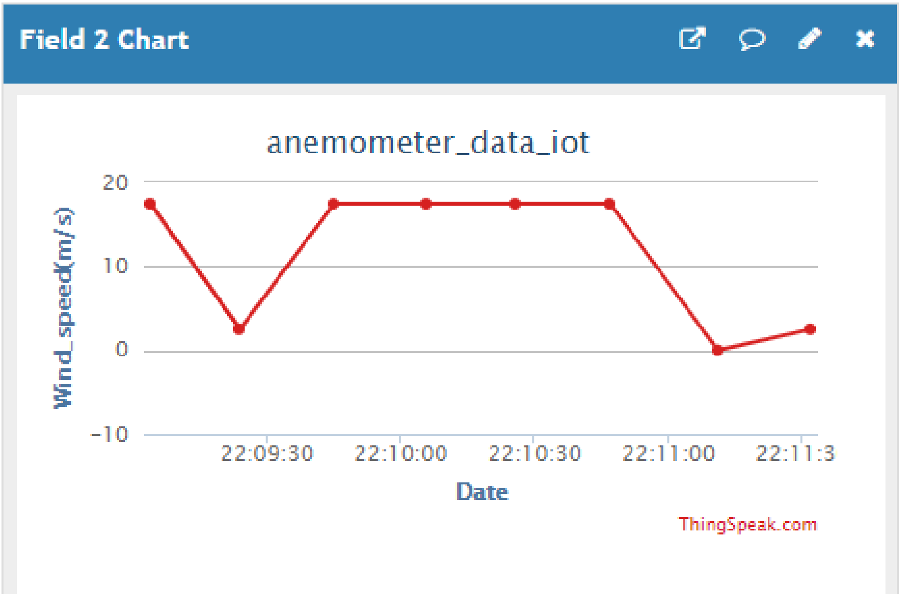

Fig.5.3 Wind_speed Vs Date graph

The graphs are plotted with Voltage and wind speed on the Y-axis in the respective graphs and the Time(Date) is taken on the X-axis in the respective graphs. The graphs tell us that the peak of the curves is obtained when the wind is striking the anemometer blades continuously and the minima points are the points where the wind speed is zero. The wind speed from the graph is 17 m/s that is peak value and the 0.1 m/s is the point where the curve is at the minima.

5.2 CONCLUSION:

The data stored in the ThingSpeak cloud platform can be obtained in a CSV file. These values are taken and Minimum, Maximum and the Average values are found. These values can be verified on a daily base and this values will help in better prediction of weather forecast. Thus, giving the justification for our contribution we would like to make towards betterment of the weather stations data tracking using Internet of Things (IOT) Technology.

REFERENCES

[1]. Wikipedia – anemometers, https://en.wikipedia.org/wiki/Anemometer

[2]. "History of the Anemometer". Logic Energy. Retrieved 14 April 2013.

[3]. *World Meteorological Organization. "Vane anemometer". Eumetcal. Retrieved 6 April 2014.*

[4]. "Hot-wire Anemometer explanation". eFunda. Archived from the original on 10 October 2006. Retrieved 18 September 2006.

[5]. Iten, Paul D. (29 June 1976). "Laser Doppler Anemometer". United States Patent and Trademark Office. Retrieved 18 September 2006.

[7]. Zeroday. "A lua based firmware for wifi-soc esp8266". Github. Retrieved 2 April 2015.

[8]. Hari Wiguna. "NodeMCU LUA Firmware". Hackaday. Retrieved 2 April 2015.

[9]. Systems, Espressif. "Espressif Systems". Espressif-WikiDevi. Retrieved 3 June 2017.

[10]. Brian Benchoff. "A DEV BOARD FOR THE ESP LUA INTERPRETER". Hackaday. Retrieved 2 April 2015.

[11]. Mpx. "Lua CJSON is a fast JSON encoding/parsing module for Lua". Github. Retrieved 2 April 2015.

[12]. Pellepl. "Wear-leveled SPI flash file system for embedded devices". GitHub. Retrieved 2 April 2015.

[13]. "Updated: Arduino announces FPGA board, ATmega4809 in Uno Wi-Fi mk2, cloud-based IDE and IoT hardware". Electronics Weekly. 2018-05-18. Retrieved 2018-06-14.

[14]. "The arduino source code". The Arduino source code.

[15]. Castro, Jorge R. Building a home security system with Arduino: design, build, and maintain a home security system with Arduino Uno. Birmingham, UK. p. 15. ISBN 9781785283802. OCLC 922588951.

[16]. Banzi, Massimo; Shiloh, Michael. Getting started with Arduino (Third ed.). Sebastopol, CA. ISBN 9781449363314. OCLC 898290173.

[17]. https://github.com/iobridge/thingspeak/blob/master/README.textile

[18]. https://www.mathworks.com/hardware-support/thingspeak.html

[19]. Brown, Eric (13 September 2016). "Who Needs the Internet of Things?". Linux.com. Retrieved 23 October 2016.

[20]. Brown, Eric (20 September 2016). "21 Open Source Projects for IoT". Linux.com. Retrieved 23 October 2016.

[21]. "Internet of Things Global Standards Initiative". ITU. Retrieved 26 June 2015.

[22]. Hendricks, Drew. "The Trouble with the Internet of Things". London Datastore. Greater London Authority. Retrieved 10 August 2015.

[23]. Umar Zakir Abdul, Hamid; et al. (2019). "Internet of Vehicle (IoV) Applications in Expediting the Implementation of Smart Highway of Autonomous Vehicle: A Survey". Performability in Internet of Things. Retrieved 28 August 2018.

[24]. HUVIO, Eero, GRÖNVALL, John, FRÄMLING, Kary. Tracking and tracing parcels using a distributed computing approach. In: SOLEM, Olav (ed.) Proceedings of the 14th Annual Conference for Nordic Researchers in Logistics (NOFOMA'2002), Trondheim, Norway, 12–14 June 2002. pp. 29-43.

[25]FRÄMLING, Kary. Tracking of material flow by an Internet-based product data management system (in Finnish: Tavaravirran seuranta osana Internet-pohjaista tuotetiedon hallintaa). Tieke EDISTY magazine, No. 1, 2002, Publication of Tieke (Finnish Information Society Development Centre), Finland, 2002. pp. 24-25.

[26].FRÄMLING, Kary, HOLMSTRÖM, Jan, ALA-RISKU, Timo, KÄRKKAINEN, Mikko. Product agents for handling information about physical objects . Report of Laboratory of Information Processing Science series B, TKO-B 153/03, Helsinki University of Technology, 2003. 20 p.

[27] Design and Construction of a Digital Anemometer, Conference: Conference: International Conference on Mechanical, Industrial and Energy Engineering 2014, At Khulna