APPENDIX 1

IOT BASED WEATHER STATION MONITORING USING NODEMCU

A PROJECT REPORT

Submitted by

SANTHYA M (2017504037)

NIKKILESHH M (2017504062)

SIVARAMAKRISHNA D (2017504614)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER 2020

APPENDIX 2

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this project titled **"IOT BASED WEATHER STATION MONITORING USING NODEMCU"** is the bonafide work of **Mr. NIKKILESHH M (2017504062), Ms. SANTHYA (2017504037), and Mr. SIVARAMAKRISHNA D (2017504614)** who carried out under my supervision.

SIGNATURE	**SIGNATURE**
Dr. A. VIJI	**Mrs. R. RAMYA**
Faculty and Guide,	Faculty and Guide,
Department of Electronics Engineering,	Department of Electronics Engineering,
Madras Institute of Technology,	Madras Institute of Technology,
Anna University,	Anna University,
Chennai - 600 044.	Chennai - 600 044.

ABSTRACT:

Weather is the most important element that affects farm production. It can influence crop growth, total yield, pest occurrence, water and fertilizer need and all farm activities carried out during the growing season. In other words, farming under open sky is greatly reliant upon weather. Climate change can lead to unpredictable weather which is beyond human control. A farmer cannot predict weather conditions; however he can adapt to the given situation and take additional farm management practices to minimize the crop losses. Therefore, accurate information regarding weather is important so that farming activities can be planned without any adverse effects. The only way to manage and ensure profitable crop production is by accurate weather forecasts. Available on any device (radio, TV, mobile or computer), it gives great value to farmers whose success largely depends on weather. Our project aims at monitoring weather conditions at a particular place and making the data accessible in Blynk application and Thingspeak cloud by interfacing sensors with Nodemcu.

APPENDIX 3

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SYMBOL	DESCRIPTION
MCU	Microcontroller Unit
IDE	Integrated Development Environment
WMS	Warehouse Management System
GPIO	General Purpose Input and Output
BMP	Barometric Pressure Sensor
LED	Light Emitting Diode
LDR	Light Dependent Resistor
ADC	Analog to Digital Convertor
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
GND	Ground
V	Voltage
UART	Universal Asynchronous Receiver Transmitter

CHAPTER 1
INTRODUCTION

In this chapter, IOT technology, its trends and characteristics are explained. Weather station monitoring and integrating IOT in our project are also discussed here.

1.1 INTRODUCTION TO IOT

With the advent of high speed internet, more and more humans around the globe are interconnected. Internet of Things (IOT) takes this a step further and connects not only humans but electronic devices which can speak amongst themselves. With falling costs of Wifi enabled devices this trend will only gather more momentum. The main concept behind IoT is to connect various electronic devices through a network and then retrieve the data from these devices (sensors) which can be distributed in any fashion, upload them to any cloud service where one can analyze and process the gathered information. In the cloud service one can utilize these data to alert people by various means such as using a buzzer or sending them an email or SMS.

As mentioned earlier, IoT enables not only Human-Human interaction, but also Human-Device interaction as well as Device-Device interaction. This particular development in the shape of new avenues of interactions will impact essentially every industry such as transportation 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 broader level changes in energy consumption, which is going to be an integral part of any nation's energy policy. Looking beyond the aforementioned energy example, there are many areas of interests where IoT can make a meaningful impact such as Smart Homes, which involve IoT to heighten the degree of automation; Wearable technologies such as smart watches and fitness bands; One of the biggest areas of potential in IoT is connected healthcare. Many global electronics behemoths have already invested deeply in the Internet of Things infrastructure. With players like Intel, Rockwell Automation, Siemens, Cisco and General Electric the market is on the cusp of an explosion, 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.

IOT is the future technology of connecting the entire world at one place. All the objects, things and sensors can be connected to share the data obtained in various locations and process/analyses that data for coordinating the applications like traffic signalling, mobile health monitoring in medical applications and industrial safety ensuring methods, etc. As per the estimation of technological experts, 50 billion objects will be connected in IOT by 2025. IOT offers a wide range of connectivity of devices with various protocols and various properties of applications for obtaining the complete machine to machine interaction.

1.2 WEATHER MONITORING SYSTEM

This project measures environmental parameters such as temperature, humidity, rainfall level, barometric pressure and light intensity, and uploads these values to a cloud service. In the cloud the data is analyzed and if the retrieved data are above or below a certain threshold limit, depending on the value, a message is displayed.

Earlier people staying at home and busy in their household chores or people busy with their office workload had no idea about the environmental parameters outside their home or office. They have no idea if the temperature 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 model which we have designed can come in quite handy in these situations. It will notify us whenever the temperature is too low or too high through Blynk application. The project can also be extended in a way that it will automatically notify whenever there is a downpour in the surrounding and remind us to carry an umbrella or a raincoat. LDR which measures the light intensity of the surrounding environment displays the message about the weather. The core of the project is the ESP8266 based Nodemcu which is a low cost wifi module and all the other sensors are connected to this device. The C code is 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 starts working. The code has to be uploaded only once.

1.3 APPLICATIONS

The IOT based Weather Monitoring System project is used to get Live reporting of weather conditions. It monitors temperature, barometric pressure, humidity, light intensity and rainfall level. Suppose scientists/nature analysts want to monitor changes in a particular environment like a volcano or a rain-forest and if these people are from different places in the world, in this case, SMS based weather monitoring systems have some limitations as it sends SMS to a few numbers and time for sending SMS increases as the number of mobile users increases. In order to know the information about the weather at a particular place, they can either visit the website or even see it through the Blynk app. Another important application is in Farm management and crop production. Farming based on weather data is crucial to successful farm management. More importantly, it ensures sustainable farming, thus protecting the environment.

1.4 PROJECT MOTIVATION

Today, the weather variables have a great effect on our modern lifestyle. Weather affects a wide range of man's activities, including agriculture and transportation. Combination of various sensors forms one weather station. The Weather Monitoring System can be either wireless or wired one. The primary motivation behind taking up this project is the large utility of wireless weather monitoring in varied areas ranging from agricultural growth and development to industrial development using NodeMCU which has inbuilt Wifi module(ESP8266) to make our analysis much easier. Weather variables like temperature, atmospheric pressure, humidity, rainfall are sensed by sensors and further processed by NodeMCU using Arduino IDE.

1.5 REPORT ORGANIZATION

Introduction to IOT, Weather station monitoring, its scope and applications are discussed in this chapter. Hardware specifications and working of sensors will be explained in Chapter 2 and software requirements will be dealt in Chapter 3. Results will be discussed in Chapters 4. Limitations and Challenges are discussed in Chapter 5. Conclusions and Future scope will be covered in Chapter 6.

CHAPTER 2
HARDWARE REQUIREMENTS

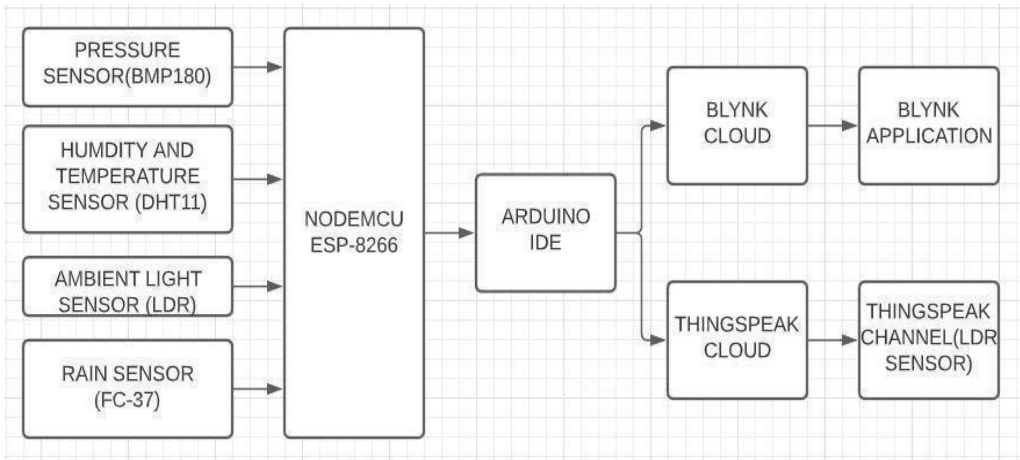
2.1 PROPOSED METHODOLOGY

In this chapter the proposed methodology, the components required, system design and implementation setup considered in this project are discussed.

2.1.1 COMPONENTS REQUIRED

1. NodeMCU ESP8266
2. Temperature and Humidity Sensor (DHT11)
3. Barometric Pressure Sensor (BMP280)
4. LDR sensor
5. Rainfall sensor (FC-37)
6. Resistors 10 kilo ohms
7. Smartphone with Blynk application installed
8. Laptop with Arduino IDE installed

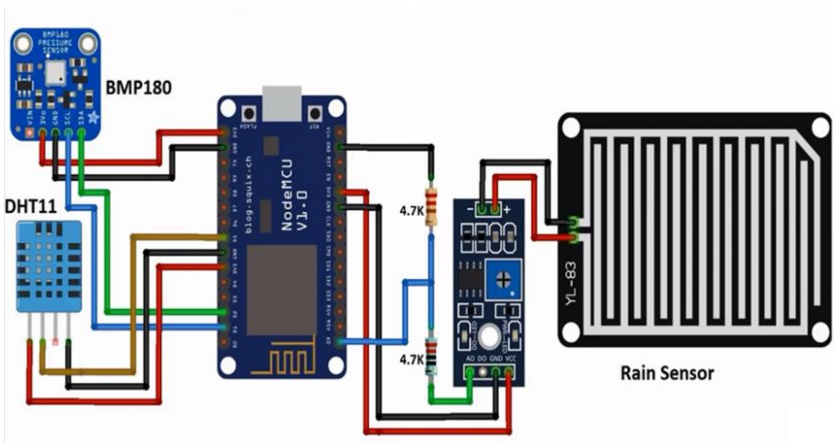
2.1.2 SYSTEM DESIGN



In this project, Nodemcu with the help of sensors measures weather parameters such as temperature, humidity, barometric pressure (Absolute and sea-level), altitude, light intensity and rainfall level using 4 sensors. These sensors are rain sensor, DHT 11, BMP280 and LDR sensor. Rain sensor is used to measure the rainfall intensity level. DHT 11 sensor is used to measure temperature and humidity. BMP 280 sensor is used for measuring barometric pressure (both Absolute and sea level) and altitude. LDR sensor is used to measure light intensity which is used to predict whether the atmosphere is sunny or cloudy. These sensors are then connected to the Nodemcu board. The sensed parameters are processed by coding in Arduino IDE. The values are then displayed in Blynk application and Thingspeak.

2.1.3 IMPLEMENTATION SETUP

The complete setup of the project is shown below:



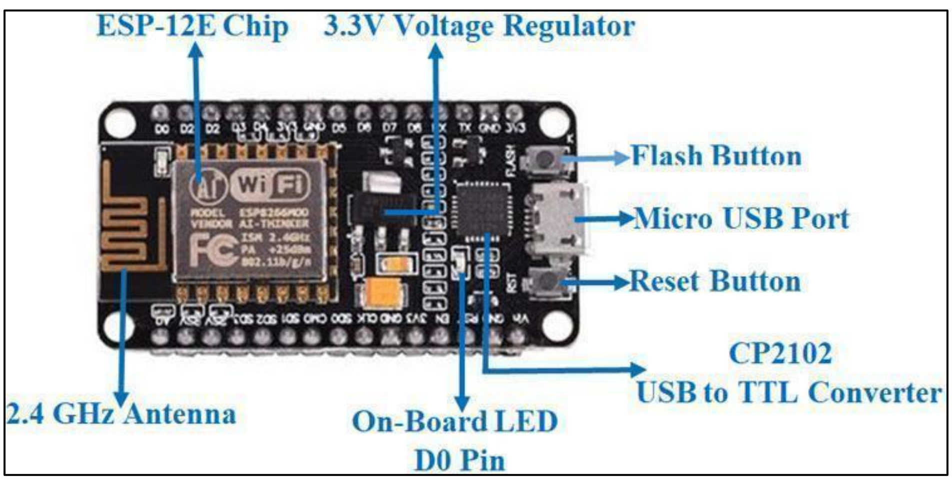
2.2 NODEMCU

NodeMCU has an in-built microcontroller and we can also use Arduino IDE to program Nodemcu. It has 4 Megabytes storage (flash memory) capacity. The module supports both 5v (micro USB) and 3.3v (GPIO) general purpose input/output pins. The ESP8266 Wi-Fi module is a low-cost Wifi microchip with full TCP/IP and microcontroller capability. This small module allows nodemcu to connect to a Wi-Fi network and make simple TCP/IP connections.

2.2.1 SPECIFICATIONS AND FEATURES

1. Microcontroller: 32 bit
2. Operating Voltage: 3.3V
3. Digital I/O Pins (DIO): 16
4. Analog Input Pins (ADC): 1
5. UARTs: 4 pins
6. SPIs: 4 pins
7. I2Cs: 2 pins
8. Flash Memory: 4 MB
9. SRAM: 64 KB
10. Clock Speed: 80 MHz

2.2.2 PARTS OF NODEMCU

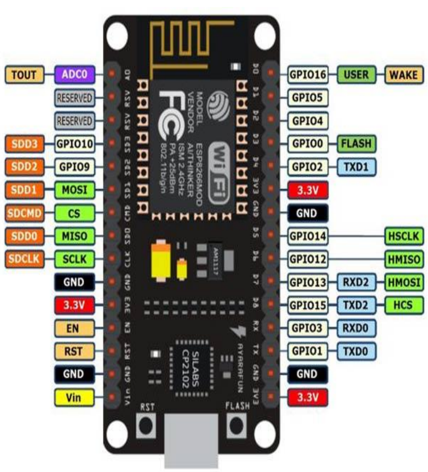


The heart of nodemcu is ESP8266 Wifi module. The board also consists of:

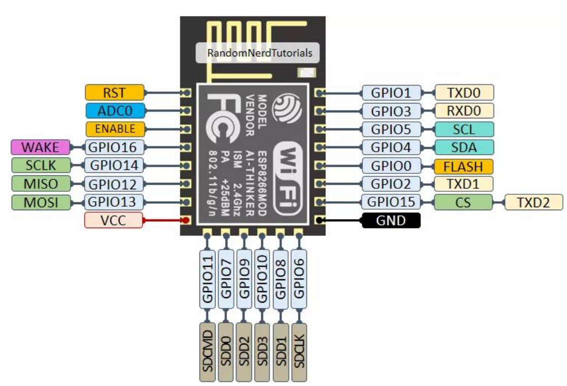
1. 3.3 V regulator
2. 2.4GHz Antenna
3. On board LED
4. Flash button
5. Micro USB port
6. RESET button
7. USB to TTL convertor

USB to TTL converter is used to provide connectivity between USB and serial UART interface.

2.2.3 PIN CONFIGURATION



NODEMCU PIN CONFIGURATION



ESP8266 PIN CONFIGURATION

Nodemcu has 30 pins (15 on left and 15 on right). ESP8266 has 16 GPIO pins. Of these, only 13 pins are used for digital I/O. 30 pins of nodemcu are classified into:

1. ADC pins
2. SPI pins
3. 12C pins
4. UART pin
5. Grnd pins
6. 3.3 V pins

2.2.4 MAPPING OF PINS

GPIO number of ESP does not match with the label on nodemcu board. Thus a clear understanding of mapping of pins is required.

NODE MCU	ESP8266
D0	GPIO16
D1	GPIO5
D2	GPIO4
D3	GPIO0
D4	GPIO2
D5	GPIO14
D6	GPIO12
D7	GPIO13
D8	GPIO15
D9/RX	GPIO3
D10/TX	GPIO1
D11/SD2	GPIO9
D12/SD3	GPIO10

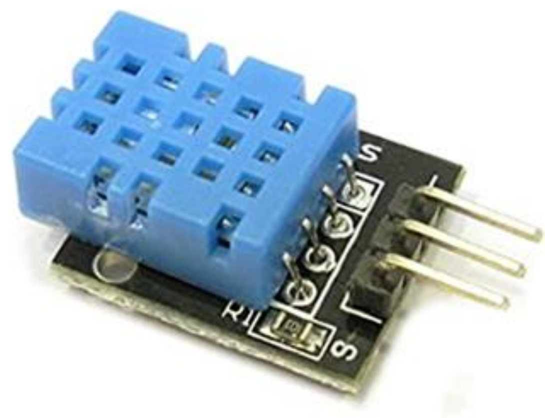
2.3 TEMPERATURE AND HUMIDITY SENSOR (DHT11)

2.3.1 OVERVIEW

Humidity is the measure of water vapour present in the air. The level of humidity in air affects various physical, chemical and biological processes. In industrial applications, humidity can affect the business cost of the products, health and safety of the employees. So, in semiconductor industries and control system industries measurement of humidity is very important. Humidity measurement determines the amount of moisture present in the gas that can be a mixture of water vapour, nitrogen, argon or pure gas etc. Humidity sensors are of two types based on their measurement units. They are a relative humidity sensor and Absolute humidity sensor. DHT11 is a digital temperature and humidity sensor.

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc. to measure humidity and temperature instantaneously.DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor. To measure the surrounding air this sensor uses a thermistor and a capacitive humidity sensor.

2.3.2 STRUCTURE AND WORKING PRINCIPLE OF DHT11 SENSOR



STRUCTURE OF DHT11 SENSOR

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measures, processes this changed resistance values and changes them into digital form. DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The

IC measures, processes this changed resistance values and changes them into digital form.

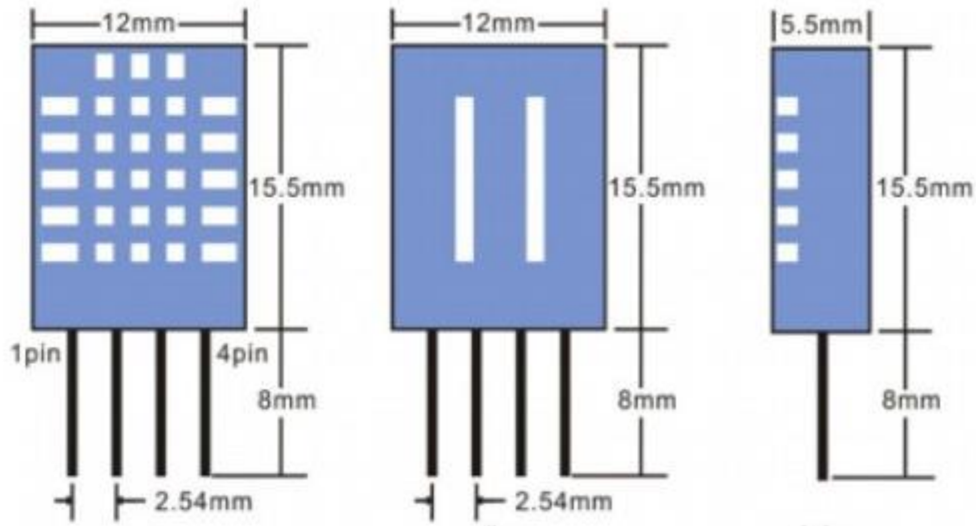
2.3.3 DHT11 SPECIFICATIONS

1. Operating Voltage: 3.5V to 5.5V
2. Operating current: 0.3mA (measuring) 60uA (standby)
3. Output: Serial data
4. Temperature Range: 0°C to 50°C
5. Humidity Range: 20% to 90%
6. Resolution: Temperature and Humidity both are 16-bit
7. Accuracy: ±1°C and ±1%

2.3.4 PIN CONFIGURATION

S.NO	PIN NAME	PIN DESCRIPTION
1.	Vcc	Power supply 3.5 to 5.5V
2.	Data	Output pins
3.	NC	No connection
4.	Ground	Connected to the ground

2.3.5 2D–MODEL OF THE SENSOR



2.3.6 APPLICATIONS

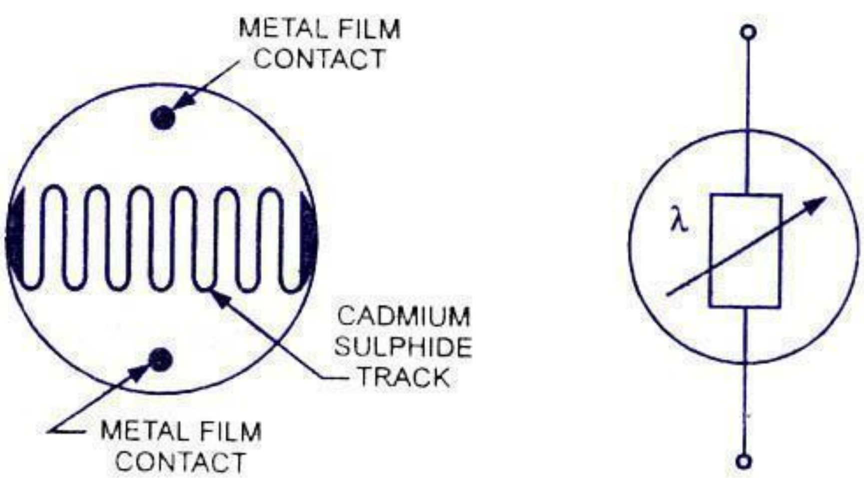
This sensor is used in various applications such as measuring humidity and temperature values in heating, ventilation and air conditioning systems. Weather stations also use these sensors to predict weather conditions. The humidity sensor is used as a preventive measure in homes where people are affected by humidity. Offices, cars, museums, greenhouses and industries use this sensor for measuring humidity values and as a safety measure. Its compact size and sampling rate made this sensor popular among hobbyists.

2.4 LIGHT DEPENDENT RESISTOR (LDR)

2.4.1 OVERVIEW

A Light Dependent Resistor (LDR) is also called a photoresistor or a cadmium sulfide (CdS) cell. It is also called a photoconductor. It is basically a photocell that works on the principle of photoconductivity. The passive component is basically a resistor whose resistance value decreases when the intensity of light decreases. This optoelectronic device is mostly used in light varying sensor circuits, and light and dark activated switching circuits.

2.4.2 STRUCTURE AND WORKING PRINCIPLE OF LDR



(a) Basic Structure *(b) Symbol*

LDR

The snake-like track shown below is the Cadmium Sulphide (CdS) film which also passes through the sides. On the top and bottom are metal films which are connected to the terminal leads. It is designed in such a way as to provide maximum possible contact area with the two metal films. The structure is housed in a clear plastic or resin case, to provide free access to external light.

As explained above, the main component for the construction of LDR is cadmium sulphide (CdS), which is used as the photoconductor and contains no or very few electrons when not illuminated. In the absence of light it is designed to have a high resistance in the range of mega ohms. As soon as light falls on the sensor, the electrons are liberated and the conductivity of the material increases. When the light intensity exceeds a certain frequency, the photons absorbed by the semiconductor give band electrons the energy required to jump into the conduction band. This causes the free electrons or holes to conduct electricity and thus dropping the resistance dramatically (< 1 Kilo ohm).

2.4.3 LDR FEATURES

1. Can be used to sense Light
2. Easy to use on Breadboard or Perf Board
3. Easy to use with Microcontrollers or even with normal Digital/Analog IC
4. Small, cheap and easily available

5. Available in PG5 ,PG5-MP, PG12, PG12-MP, PG20 and PG20-MP series

2.4.4 LDR TECHNICAL SPECIFICATIONS

Parameter	Conditions	Min	Typ	Max	Unit
Cell resistance	1000 LUX 10 LUX	- -	400 9	- -	Ohm K Ohm
Dark Resistance	-	-	1	-	M Ohm
Dark Capacitance	-	-	3.5	-	pF
Rise Time	1000 LUX 10 LUX	- -	2.8 18	- -	ms ms
Fall Time	1000 LUX 10 LUX	- -	48 120	- -	ms ms
Voltage AC/DC Peak		-	-	320	V max
Current		-	-	75	mA max
Power Dissipation				100	mW max
Operating Temperature		-60	-	+75	Deg. C

2.4.5 APPLICATIONS

Light-dependent resistors are simple and low-cost devices. These devices are used where there is a need to sense the presence and absence of light is necessary. These resistors are used as light sensors and the applications of LDR mainly include alarm clocks, street lights, light intensity meters, burglar alarm circuits. For a better understanding of this concept, here we have explained one project namely; power conserving of intensity controlled street lights using LDR.

2.5 RAINFALL SENSOR (FC-37)

2.5.1 OVERVIEW

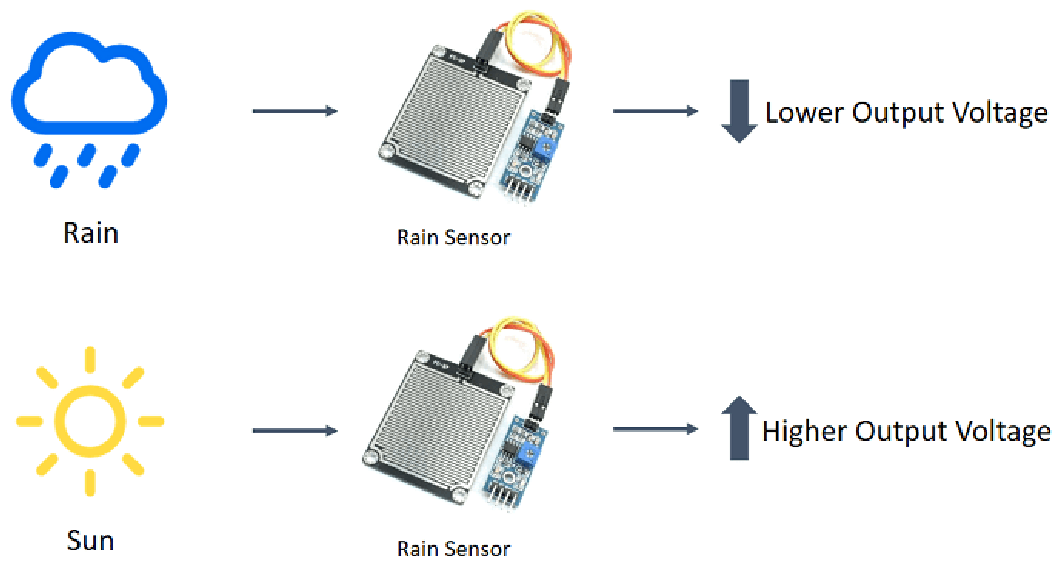
Raindrop Sensor is a tool used for sensing rain. It consists of two modules, a rain board that detects the rain and a control module, which compares the analog value, and converts it to a digital value. The raindrop sensors can be used in the automobile sector to control the windshield wipers automatically, in the agriculture sector to sense rain and it is also used in home automation systems.

2.5.2 STRUCTURE AND WORKING PRINCIPLE OF FC-37

Basically, the resistance of the collector board varies accordingly to the amount of water on its surface.

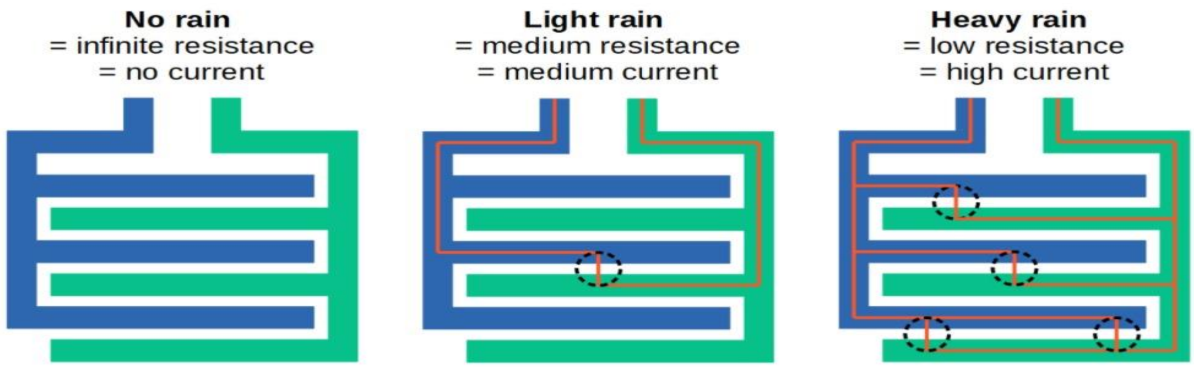
When the board is:

1. Wet: the resistance increases, and the output voltage decreases
2. Dry: the resistance is lower, and the output voltage is higher



2.5.3 RAIN SENSOR FEATURES

1. Working voltage 5V
2. Output format: Digital switching output (0 and 1), and analog voltage output AO
3. Potentiometer adjust the sensitivity
4. Uses a wide voltage LM393 comparator
5. Comparator output signal clean waveform is good, driving ability, over 15mA
6. Anti-oxidation, anti-conductivity, with long use time
7. With bolt holes for easy installation
8. Small board PCB size: 3.2cm x 1.4cm



2.5.4 PIN CONFIGURATION

S.NO	PIN NAME	PIN DESCRIPTION
1.	Vcc	Connects supply voltage -5V
2.	GND	Connected to ground
3.	D0	Digital pin to get digital output
4.	A0	Analog pin to get analog output

2.5.5 APPLICATIONS

1. Weather monitoring
2. Automatic windshield wipers
3. Smart Agriculture
4. Home-Automation

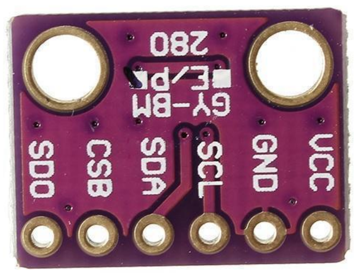
2.6 BAROMETRIC PRESSURE SENSOR (BMP280)

2.6.1 OVERVIEW

Robert Bosch is the world market leader for pressure sensors in automotive and consumer applications. Bosch’s proprietary APSM (Advanced Porous Silicon Membrane) MEMS manufacturing process is fully CMOS compatible and allows a hermetic sealing of the cavity in an all silicon process. The BMP280 is based on Bosch’s proven Piezo-resistive pressure sensor technology featuring high EMC robustness, high accuracy and linearity and long term stability. The BMP280 is an absolute barometric pressure sensor especially designed for mobile applications.

The sensor module is housed in an extremely compact 8-pin metal-lid LGA package with a footprint of only 2.0 × 2.5 mm2 and 0.95 mm package height. Its small dimensions and its low power consumption of 2.7 µA @1Hz allow the implementation in battery driven devices such as mobile phones, GPS modules or watches. As the successor to the widely adopted BMP180, the BMP280 delivers high performance in all applications that require precise pressure measurement. The BMP280 operates at lower noise, supports new filter modes and an SPI interface within a footprint 63% smaller than the BMP180.

2.6.2 STRUCTURE AND WORKING PRINCIPLE OF BMP280



STRUCTURE OF BMP280

1. The working principle of BMP280 is based on weight of air.
2. At higher temperatures, air is not as dense and heavy, so it applies less pressure on the sensor.
3. At lower temperatures, air is denser and weighs more, so it exerts more pressure on the sensor.
4. The sensor uses real-time temperature measurement to compensate the pressure readings for changes in air density.

2.6.3 FEATURES OF BMP280

1. The BMP280 sensor is an environmental sensor with temperature, barometric pressure that is the next generation upgrade to the BMP085/BMP180/BMP183.
2. This sensor is great for all sorts of weather sensing and can even be used in both I2C and SPI.
3. This precision sensor is the best low-cost, precision sensing solution for measuring barometric pressure with ±1 hPa absolute accuracy, and temperature with ±1.0°C accuracy. Because pressure changes with altitude and the pressure measurements are so good, we can also use it as an altimeter with ±1 meter accuracy.

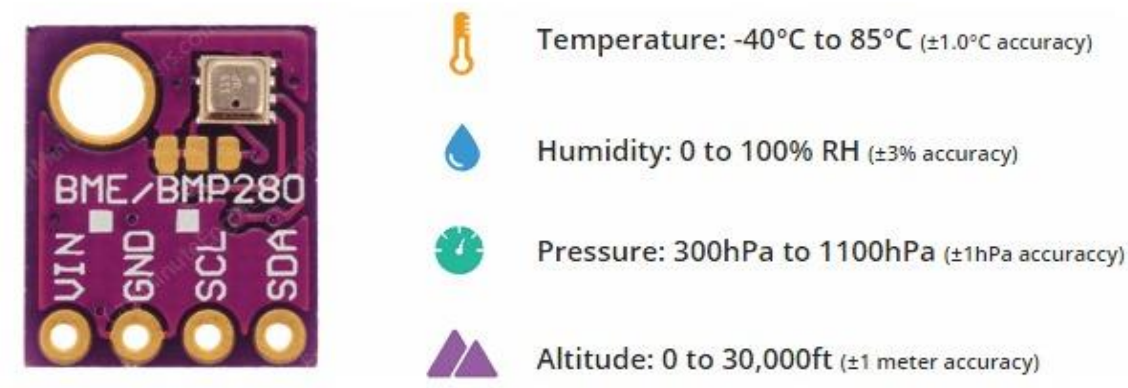
2.6.4 TECHNICAL SPECIFICATIONS OF BMP280

1. Model: BMP280 -3.3
2. Chip: BMP280
3. Power supply: 3V/3.3V DC
4. Peak current: 1.12mA
5. Air pressure range : 300-1100hPa (equivalent to +9000m to -500m above sea level)
6. Temperature range: -40 °C to +85 °C
7. Digital interfaces: I²C (up to 3.4 MHz) and SPI (3 and 4 wire, up to 10 MHz)

8. Current consumption of sensor BMP280: 2.7µA @ 1 Hz sampling rate
9. Pin pitch: 2.54mm
10. Module size: 11.5mm*15mm

2.6.5 PIN CONFIGURATION

S.NO.	PIN NAME	DESCRIPTION
1.	Vcc	Power source of 3.3VDC
2.	GND	Ground
3.	SLC	Serial Clock
4.	SDA	Serial Data
5.	CSB	CSB pin to GND to have SPI and to V_{cc} (3.3V) for I2C. It's an input to the chip
6.	SDO	**S**erial **D**ata **O**ut / **M**aster **I**n **S**lave **O**ut pin, for data sent from the BMP280 to NodeMCU



2.6.6 APPLICATIONS

1. Enhancement of GPS navigation (e.g. time-to-first-fix improvement, dead-reckoning, slope detection)
2. Indoor navigation (floor detection, elevator detection)
3. Outdoor navigation, leisure and sports applications
4. Weather forecast, Home weather stations
5. Health care application
6. Vertical velocity indicator (e.g. risk/sink speed)
7. Handsets such as mobile phones, tablet PCs, GPS devices
8. Flying toys

CHAPTER 3
SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

The Software Requirements for this project include:

1. Arduino IDE
2. Thingspeak
3. Blynk

3.1 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

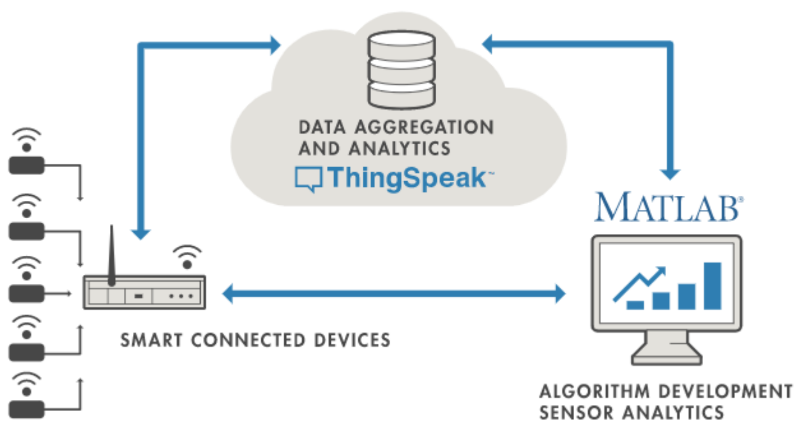
The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. 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 tool chain, also included with the IDE distribution.

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. By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards.

Using the IDE we upload the data from sensors to Blynk server and ThingSpeak cloud platform to analyze and visualize weather data in real-time.

3.2 THINGSPEAK

ThingSpeak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT 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.



It's an IoT analytics platform service that allows us to aggregate, visualize, and analyze live data streams in the cloud. We can send data to ThingSpeak from our devices, create instant visualization of live data, and send alerts. ThingSpeak was originally launched by ioBridge in 2010 as a service in support of IoT applications. ThingSpeak has integrated support from the numerical computing software MATLAB from MathWorks, allowing ThingSpeak users to analyze and visualize uploaded data using Matlab without requiring the purchase of a Matlab license from Mathworks.

ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak, you can perform online analysis and process data as it comes in. ThingSpeak is often used for prototyping and proof-of-concept IoT systems that require analytics.

We can send data from any internet-connected device directly to ThingSpeak using a Rest API or MQTT. In addition, cloud-to-cloud integrations with The Things Network, Senet, the Libelium Meshlium gateway, and Particle.io enable sensor data to reach ThingSpeak over LoRaWAN® and 4G/3G cellular connections.

With ThingSpeak, we can store and analyze data in the cloud without configuring web servers, and you can create sophisticated event-based email alerts that trigger based on data coming in from connected devices.

3.3 BLYNK

Blynk is an Internet of things (IoT) company which provides a platform for building mobile (IOS and Android) applications that can connect electronic devices to the Internet and remotely monitor and control these devices. Blynk was founded by Pavel Bayborodin, a user experience (UX) expert in mobile and automotive space. The IoT platform was launched in 2014. Blynk platform is used by engineers to connect MCUs and prototyping development boards like Arduino, ESP8266 or SBCs like Raspberry Pi over Wi-Fi, Ethernet or the cellular to the Internet and build custom mobile applications to remotely monitor and control electronic equipment. Blynk Cloud is open-source.

3.3.1 BLYNK ARCHITECTURE

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, and it can store, visualize and analyze data.

There are three major components in the platform:

Blynk App - allows us to create amazing interfaces for projects using various widgets we provide.

Blynk Server - responsible for all the communications between the smartphone and hardware. We can use the Blynk Cloud or run a private Blynk server locally. Its open-source could easily handle thousands of devices and can even be launched on a Raspberry Pi.

Blynk Libraries - for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands.

All the aforementioned components communicate with each other to build a fully functional IoT application that can be controlled from anywhere through a preconfigured connectivity type. We can control your hardware from the Blynk app running on mobile devices through the Blynk Cloud or Blynk's personal server. It works the same in the opposite direction by sending rows of processed data from hardware to your Blynk app.

Examples of platform applications are Smart Home, environmental monitoring, industrial equipment remote control and Weather monitoring system.

CHAPTER 4
ALGORITHMS

4.1 WEATHER STATION MONITORING CODE USING BLYNK

```
#include <Adafruit_Sensor.h>      //In order to use DHT sensor we have to include this library first
#define BLYNK_PRINT Serial

//Include Libraries
#include <DHT.h>                  //Including the DHT library
#include <ESP8266WiFi.h>      //Including the ESP8266 WiFi library in order to use them
#include <BlynkSimpleEsp8266.h>   //library for linking up Blynk with ESP8266
#include <Wire.h>                 //For using I2C connection of BMP180 in order to connect it to the board
#include <Adafruit_BMP280.h>       //Including the library for BMP180

Adafruit_BMP280 bmp;              /            /Defining the object bmp
#define I2C_SCL 12    //Connect SCL pin of BMP180 to GPIO12 (D6) of Nodemcu
```

```
#define I2C_SDA 13     //Connect SDA pin of BMP180 to GPIO13 (D7) of Nodemcu
#define led D1
#define sensor A0
int RainSensor = A0;
int RS=0;       //variable to store Raindrop Sensor Module's output value
float dst,bt,bp,ba;
char dstmp[20],btmp[20],bprs[20],balt[20];
bool bmp280_present=true;

char auth[] = "brSKF-2o8kDY0xC_izqZJn670edawayX"; //Authentication Key will be there in the Blynk App

//Mention the SSID and Password
char ssid[] = "vivo 1920"; //SSID of the WiFi hotspot available
char pass[] = "santhya2000";  //Password of the WiFi

#define DHTPIN 2    //Connect the DHT11 sensor data pin to GPIO2(D4) of Nodemcu
#define DHTTYPE DHT11   //Mention the type of sensor we are using
DHT dht(DHTPIN, DHTTYPE); //Defining the pin and the dhttype
BlynkTimer timer;
```

```
void sendSensor()
{

//Check the working of BMP180 sensor
      if (!bmp.begin())
      {
         Serial.println("Failed to read from BMP Sensor!");
         while (1) {}
      }

//Getting the Humidity and temperature value from DHT11
      float h = dht.readHumidity();
     float t = dht.readTemperature(); // or dht.readTemperature(true) for
Fahrenheit

//Check the working of DHT11 sensor
      if (isnan(h) || isnan(t))
      {
         Serial.println("Failed to read from DHT sensor!");
         return;
      }
```

//Measuring the Dew Point

double gamma = log(h/100) + ((17.62*t) / (243.5+t));

double dp = 243.5*gamma / (17.62-gamma);

//Checking for Rain

RS = analogRead(RainSensor);

RS = constrain(RainSensor, 150, 440);

RS = map(RainSensor, 150, 440, 1023, 0);

//Reading the value of Pressure, Temperature, Altitude from the BMP280

float bp = bmp.readPressure()/100; // Division by 100 makes it in millibars

float ba = bmp.readAltitude();

float bt = bmp.readTemperature();

float dst = bmp.readSealevelPressure()/100;

//LDR SENSOR CODE

int LDR = analogRead(sensor);

if(LDR <150)

{

digitalWrite(led, HIGH);

Blynk.notify("Light ON");

```
  }
else
  {
    digitalWrite(led, LOW);
  }

  Blynk.virtualWrite(V7, LDR); // Give pin which is free in node mcu, make sure v7 is not overlapped

 //Printing the values of the above read value on to the Virtual Pins in the Blynk App
          Blynk.virtualWrite(V5 , h);
          Blynk.virtualWrite(V6 , t);
          Blynk.virtualWrite(V10, bp);
          Blynk.virtualWrite(V11, ba);
          Blynk.virtualWrite(V13, bt);
          Blynk.virtualWrite(V13, dst);
          Blynk.virtualWrite(V14, dp);
          Blynk.virtualWrite(V15, RS);

}
```

```
void setup()
{

    Serial.begin(9600); //Initializing the Serial Monitor with a Baud Rate of 9600

    Blynk.begin(auth, ssid, pass);
    dht.begin();  //Initializing the DHT sensor
    Wire.begin(I2C_SDA, I2C_SCL); //Initializing the I2C connection
     delay(10);
     timer.setInterval(1000L, sendSensor);

    // FOR LDR
      pinMode(led, OUTPUT);
      Serial.begin(9600);
      Blynk.begin(auth, ssid, pass);
      timer.setInterval(500L, sendSensor);
}
void loop()
{
  Blynk.run();
  timer.run();
}
```

4.2 LIGHT INTENSITY DETECTION CODE USING THINGSPEAK

```
#include <ESP8266WiFi.h>;
#include <WiFiClient.h>;
#include <ThingSpeak.h>;
const char* ssid = "vivo 1920";                        //Network SSID
const char* password = "santhya2000";           //Network Password
int val;
int LDRpin = A0; //LDR Pin Connected at A0 Pin

WiFiClient client;
unsigned long myChannelNumber = 1196476 ; //Channel Number
(Without Brackets)
const char * myWriteAPIKey = "Y8GM6OPBLRC07UUS"; //Your
Write API Key

void setup()
{
Serial.begin(9600);
delay(10);


// Connect to WiFi network
```

```
WiFi.begin(ssid, password);
ThingSpeak.begin(client);
}

void loop()
{
val = analogRead(LDRpin); //Read Analog values and Store in val variable

Serial.print(val);//Print on Serial Monitor
Serial.println('\t');
if ( val < 600 )
{
Serial.print(" Sunny Weather");
Serial.println('\n');
}
else if ( val > 600 && val < 1024)
{
Serial.print( "Gloomy Weather");
Serial.println('\n');
}
else if ( val == 1024)
```

```
{
Serial.print( "No ambient light is detected");
Serial.println('\n');
}
delay(1000);

ThingSpeak.writeField(myChannelNumber, 1,val, myWriteAPIKey);
//Update in ThingSpeak

delay(100);

}
```

CHAPTER 5
OUTPUTS AND RESULTS

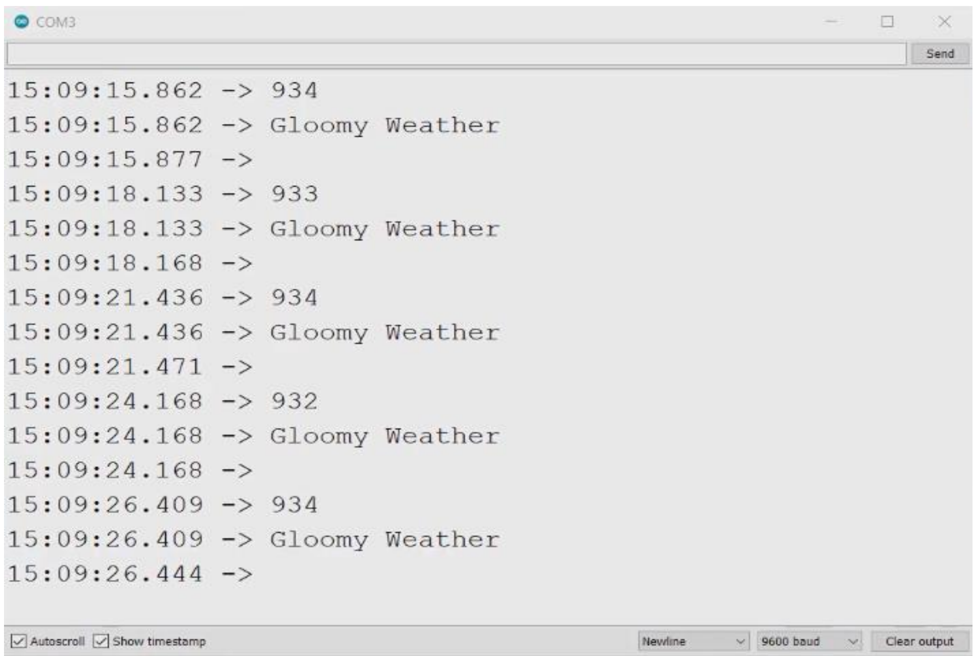
5.1 BLYNK OUTPUTS

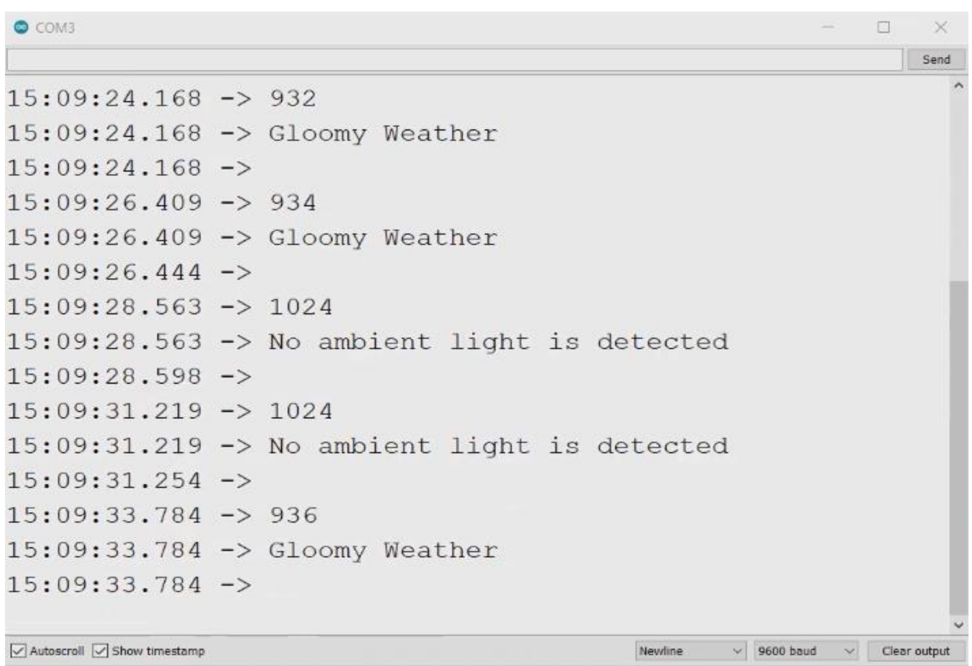
The outputs are taken under varying atmospheric conditions using Blynk android application.

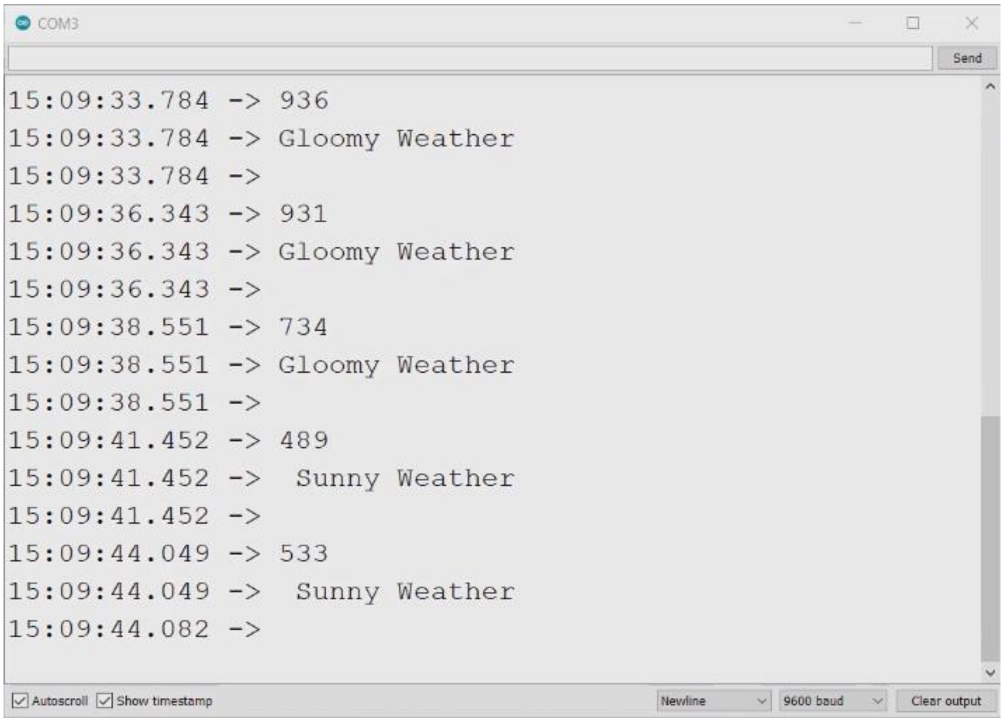

DHT11
TEMPERATURE
V6
29.400°C
0
100
HUMIDITY
V5
84.000%
0
100
PRESSURE
V10
1006.21
0
100
ALTITUDE
V11
58.81
0
2000
SEALEVEL
V13
30.09
0
1023
RAIN
V15
1492
0
200
DHT11
TEMPERATURE
V6
29.200°C
0
0
HUMIDITY
V5
85.000%
0
0
PRESSURE
V10
1006.48
0
100
ALTITUDE
V11
56.52
0
2000
SEALEVEL
V13
29.93
0
1023
RAIN
V15
1492
0
200

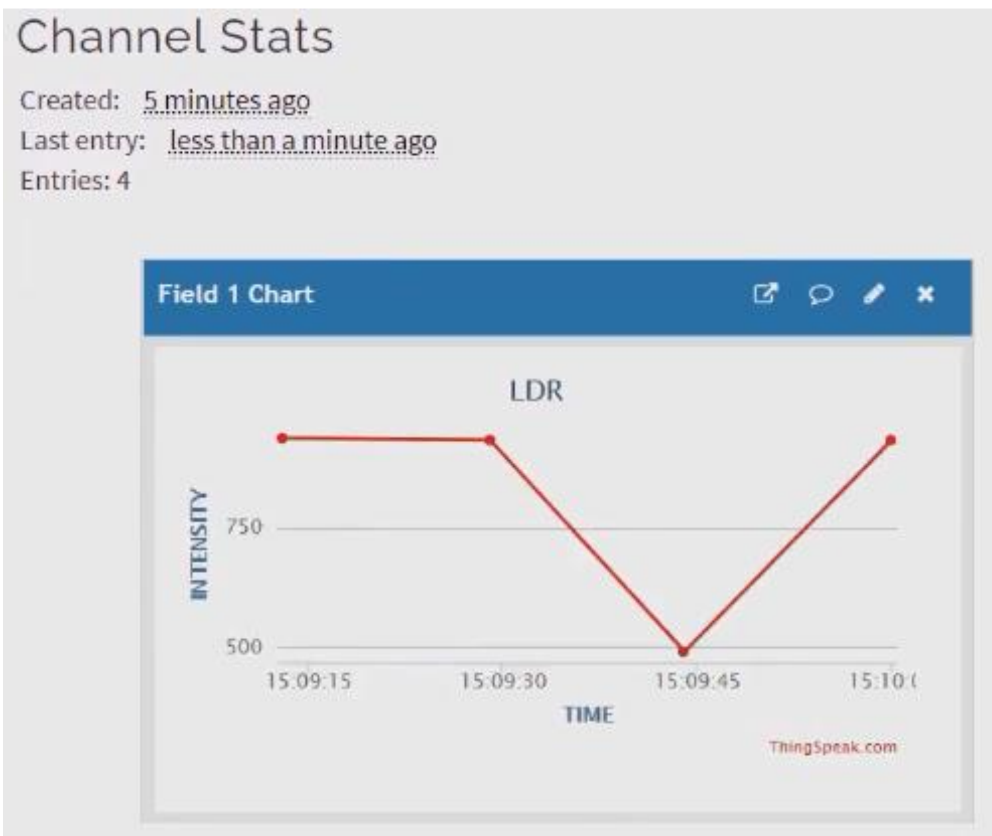
5.2 THINGSPEAK OUTPUTS

The outputs are taken under varying ambient light conditions in a serial monitor and a graph of light intensity vs. time is plotted.






COM3
Send
15:09:33.784 -> 936
15:09:33.784 -> Gloomy Weather
15:09:33.784 ->
15:09:36.343 -> 931
15:09:36.343 -> Gloomy Weather
15:09:36.343 ->
15:09:38.551 -> 734
15:09:38.551 -> Gloomy Weather
15:09:38.551 ->
15:09:41.452 -> 489
15:09:41.452 -> Sunny Weather
15:09:41.452 ->
15:09:44.049 -> 533
15:09:44.049 -> Sunny Weather
15:09:44.082 ->
Autoscroll
Show timestamp
Newline
9600 baud
Clear output


Channel Stats
Created: 5 minutes ago
Last entry: less than a minute ago
Entries: 4
Field 1 Chart
LDR
INTENSITY
750
500
15:09:15
15:09:30
15:09:45
TIME
ThingSpeak.com

5.3 VIDEO LINKS FOR OUTPUTS

1. Video demonstration of interfacing NodeMCU with sensors using Blynk application is given below:

   https://youtu.be/WesfCsRzISk

2. Video demonstration of interfacing NodeMCU with LDR using Thingspeak is given below:

   https://youtu.be/8eArtzslDYU

CHAPTER 6
LIMITATIONS AND CHALLENGES

The following are the limitations faced during the course of the project:

1. Due to inconsistencies in the sensors, accurate prediction of weather parameters were not possible at all times.
2. An active internet connection is always required for the sensors to upload the data to cloud services and monitor the data in real time.
3. Due to lack of materials for a tough exterior cover for the system, the device was not able to function in harsh weather conditions.
4. Due to energy constraint in Blynk application, only six widgets could be displayed. To display more than six parameters a paid subscription was required.
5. Project cost was more expensive than predicted as some of the components failed to work properly.

CHAPTER 7
CONCLUSIONS AND FUTURE SCOPE

CONCLUSIONS

To implement this, we need to deploy the sensor devices in the environment for collecting the data and analysis. By deploying sensor devices in the environment it will record real time data. It can cooperate with other objects through the network. Then the collected data and analysis results will be available to the end user through the Wi-Fi. The smart way to monitor the environment and an efficient, low cost entrenched system is presented with different models in this Paper.

In the proposed architecture purposes of different modules were discussed. The noise and air pollution monitoring system with Internet of Things (IoT) concept experimentally tested for monitoring two parameters. It also sent the sensor parameters to the cloud (Google Spreadsheets). This data will be cooperative for future analysis and it can be easily shared to other end users. This model can be further expanded to monitor the developing cities and manufacturing zones for pollution monitoring. To protect the public health from pollution, this model provides an efficient and low cost solution for unceasing monitoring of the environment.

Additions that can be made to improve the system:

1. Powering the device using solar panels.
2. Suspending the device from a weather balloon so that it can be used to record atmospheric parameters at high altitudes and remote and inaccessible areas.
3. Use of a tough exterior cover for the system that will act as a protective cover enabling the device to function in harsh weather conditions.
4. Designing a method to mount the weather monitoring device onto a buoyant platform like a buoy hence enabling the system to measure weather changes over the sea. This data can also be shared to Cargo ships and other nautical industries conducting operations within the area.
5. Using silica gel to prevent condensation on the exterior cover as condensation might affect the sensors readings.

FUTURE SCOPE

One can implement a few more sensors and connect it to the satellite as a global feature of this system. Adding more sensors to monitor other environmental parameters such as CO_2, Pressure and Oxygen Sensor In aircraft, navigation and military there is a great scope of this real-time system. It can also be implemented in hospitals or medical institutes for the research & study in “Effect of Weather on Health and Diseases”, hence to provide better precaution alerts.

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.

REFERENCES

[1] Internet of Things (IOT) Based Weather Monitoring
system. AUTHOR : Bulipe Srinivas Rao, Prof. Dr. K. SrinivasaRao,
Mr. N. Ome

[2] IOT BASED ENVIRONMENT MONITORING SYSTEM .
AUTHOR : Snehal R. Shinde , A. H. Karode, Dr S. R. Suralkar

[3] Mobile APP & IoT Based Station Weather Station.
AUTHOR : K. N. V. SATYANARAYANA , S. R. N. REDDY , K. N.
V. SURESH VARMA & P. KANAKA RAJU

[4] Arduino Based Weather Monitoring System . AUTHOR :
Karthik Krishnamurthi, Suraj Thapa, Lokesh Kothari, Arun
Prakash

[5] A Low-Cost Microcontroller-based Weather Monitoring System, Kamarul Arifin Noordin, Chow CheeOnn and
Mohamad Faizal Ismail, CMU. Journal (2006) Vol. 5(1).

[6] Microcontroller based Real Time weather monitoring device with GSM, K C Goudal, Preetham V R and M N
Shanmukha Swami, IJSETR Vol. 3, Issue 7, July 2014.

[7] "Arduino." [Online]. Available: http://www.arduino.cc/download/

[8] "Blynk."[Online]. Available: https://docs.blynk.cc/

[9] "ThingSpeak."[Online] - an IoT analytics platform service. Available: https://www.mathworks.com/help/thingspeak/