IOT Based WEATHER STATION SYSTEM

[A Project Report for Industrial Internship/ Minor Project]

Submitted by Swarnava Gayen

In the partial fulfillment for the award of the ***degree of*** **B.Tech** in the **Electronics & Communication Engineering** Department.

Academy of Technology

At **Ardent Computech Pvt. Ltd.**

ARDENT

COMPUTECH PVT. LTD

Knowledge is power, power leads to growth, growth is life

ARDENT
COMPUTECH PVT LTD
Knowledge is power, power leads to growth, growth is life

CERTIFICATE FROM SUPERVISOR

This is to certify that **Swarnava Gayen,** successfully completed the project titled **"IOT based Weather Station System"** under my supervision during the period from March 25,2021 to April 19,2021 that is in partial fulfillment of requirements for the award of the B.Tech and submitted to the B.Tech ECE Department of —Academy of Technology.

Signature of the Supervisor

Date: 11th August, 2022

Name of the Project Supervisor: SHOUVIK SARKAR

Faculty details- 9 yrs experience MTech ,B.Tesch,

MCCAPC,CSSP,FIE,Meber of research IEI,IAENG,FOSE

Acknowledgement

The achievement that is associated with the successful completion of any task would be incomplete without mentioning the names of those people whose endless cooperation made it possible. Their constant guidance and encouragement made all our efforts successful. We take this opportunity to express our deep gratitude towards our project mentor, SHOUVIK SARKAR for giving such valuable suggestions, guidance and encouragement during the development of this projectwork.

Last but not the least we are grateful to all the faculty members of Ardent Computech Pvt. Ltd. for their support.

ABSTRACT

The proposed method in this study uses the IOT technology to collects data related to the weather and environment using raindrop sensor.

IOT Live Weather Station Monitoring Using NodeMCU ESP8266. This project is all about IoT based Live Weather Station Monitoring Using NodeMCU ESP8266. We will interface FC37 Rain Sensor with NodeMCU ESP8266-12E Wifi Module. We will measure temperature, rainfall and upload the data to a ThinkSpeak Cloud server.

INDEX

WEATHER STATION SYSTEM using IoT

Introduction

A weather station can be described as an instrument or device, which provides us with the information of the weather in our neighbouring environment. A weather station is a device that collects data related to the weather and environment using raindrop sensor

Weather Station is also a facility that can be used for rainfall forecasts and to study the weather and climate of the world in real time. The weather station using internet of things is proposed to help user access data about the weather anywhere in real-time

Components List

Hardware:

- **NodeMCU 8266**
- **FC-37 Raindrop Sensor**
- **Breadboard**
- **Jumper wires**
- **Micro USB Cable**

Software:

- **Arduino IDE.**
- **Windows Operating System.**
- **Cloud Platform used -Thing Speak**
- **Jupyter Notebook**

INTERNET OF THINGS

The Internet of things (IoT) is known as connecting objects like cell phones, personal computer and other devices to the world wide web, which introduces a new era in the area of communication, where objects communicate with each other without human intervention. The establishment of IoT has led to increased research in the area of IoT and development of social distance maintaining and smart monitoringsystemand is becoming really famous nowadays. Most of the equipments and gadgets are controlled and monitored to help and assist humans. Moreover, various wireless technologies assist in communicating with remote places which play a great role in the intelligence of shopping malls,hospitals,hotels,resturants,stadiums. IoT is a sophisticated network.

HARDWARE COMPONENTS DESCRIPTION

NODEMCU

NodeMCU is a low-cost open source IoT platform.It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module.Later, support for the ESP32 32-bit MCU was added. NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit).The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits Both the firmware and prototyping board designs are open source.

The firmware uses the Lua scripting language. The firmware 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 . Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

The prototyping hardware typically used is a circuit board functioning as a dual in-line packagee (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards The design was initially based on the ESP-12 module of the ESP8266 which is a Wi-Fi SoC integrated with a TensilicaXtensa LX106 core, widely used in IoTapplications

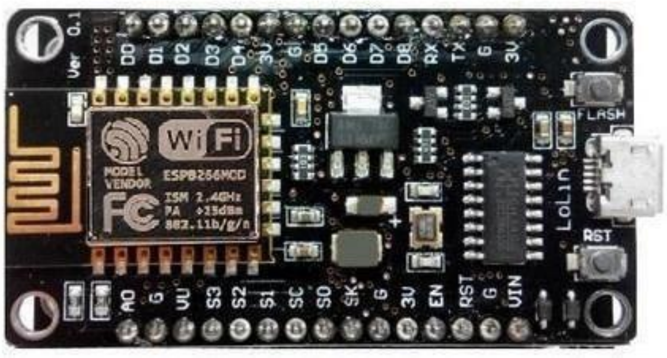
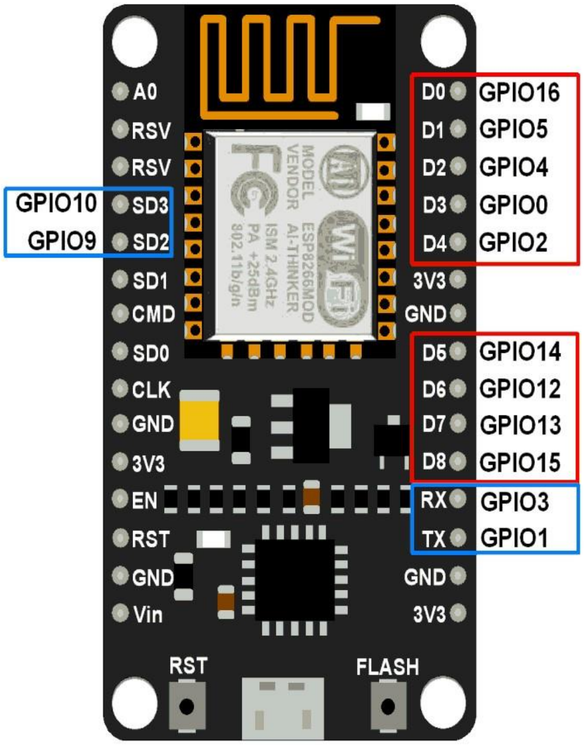

Ver 0.1
D0 D1 D2 D3 D4 3V G D5 D6 D7 D8 RX TX G 3V
FLASH
WiFi
ESP8266MOD
ISM 2.4GHz
PA +25dBm
802.11b/g/n
LoLin
RST
A0 G VU S3 S2 S1 SC S0 SK G 3V EN RST G VIN

Figure.1.1 **ESP8266 BOARD**

The **NodeMCU ESP8266 development board** comes with the ESP-12E module containing ESP8266 chip having TensilicaXtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoTprojects.

NodeMCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

NodeMCU Pinout and Functions Explained



Category		
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to	NodeMCU has 16 general purpose input-output pins on its board

Pin **Name** **Description**

	GPIO16	
SPI Pins	SD1, CMD, SD0 CLK	,NodeMCU has four pins available for SPI communication.
UART Pins UART1	TXD0, RXD	), NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and
	TXD2, RXD2	(RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C

NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU XtensaLX106
- Operating Voltage:3.3V
- Input Voltage:7-12V
- Digital I/O Pins (DIO):16
- Analog Input Pins (ADC):1
- UARTs:1
- SPIs:1
- I2Cs: 1
- Flash Memory: 4MB
- SRAM: 64KB
- Clock Speed: 80MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug nPlay
- PCBAntenna
- Small Sized module to fit smartly inside your IoTproject

Applications of NodeMCU

- Prototyping of IoTdevices
- Low power battery operatedapplications
- Networkproject.
- Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities.

Insight Into ESP8266 NodeMCU Features & Using It With Arduino IDE

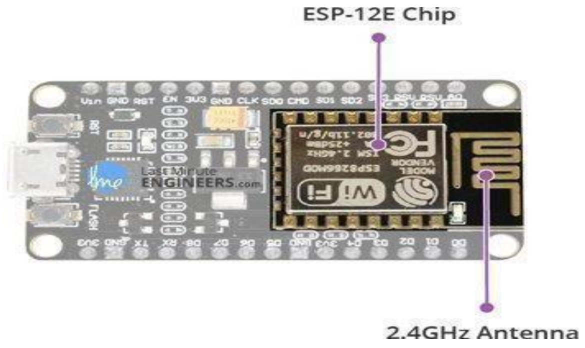
The Internet of Things (IoT) has been a trending field in the world of technology. It has changed the way we work. Physical objects and the digital world are connected now more than ever. Keeping this in mind, Espressif Systems (A Shanghai-based Semiconductor Company) has released an adorable, bite- sized WiFi enabled microcontroller – ESP8266, at an unbelievable price! For less than $3, it can monitor and control things from anywhere in the world – perfect for just about any IoT project.

ESP-12E Module

The development board equips the ESP-12E module containing ESP8266 chip having TensilicaXtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency and supports RTOS.

ESP-12E Chip

- · TensilicaXtensa® 32-bitLX106
- · 80 to 160 MHz ClockFreq.
- · 128kB internalRAM
- · 4MB external flash
- · 802.11b/g/n Wi-Fitransceiver



There’s also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays.

.

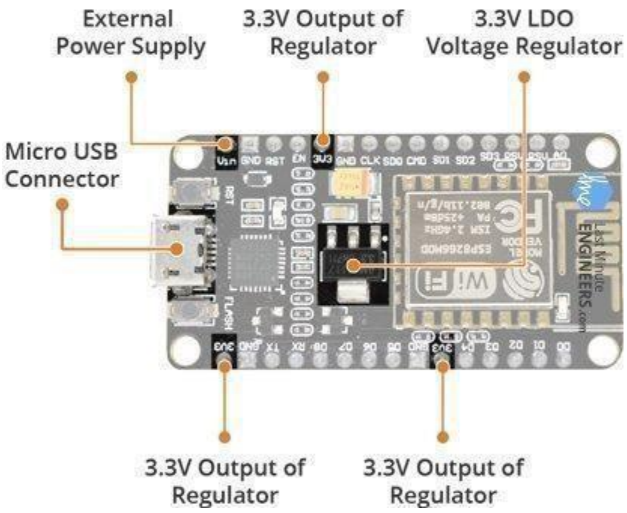
The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devicesto connect directly to it. This makes the ESP8266 NodeMCU even moreversatile.

Power Requirement

As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as 80mA during RF transmissions. The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.

Power Requirement

- · Operating Voltage: 2.5V to3.6V
- · On-board 3.3V 600mAregulator
- · 80mA OperatingCurrent
- · 20 µA during SleepMode



Power to the ESP8266 NodeMCU is supplied via the on-board MicroB USB connector. Alternatively, if you have a regulated 5V voltage source, the VIN pin can be used to directly supply the ESP8266 and its peripherals.

Warning:

The ESP8266 requires a 3.3V power supply and 3.3V logic levels for communication. The GPIO pins are not 5V-tolerant! If you want to interface the board with 5V (or higher) components, you'll need to do some level shifting.

Peripherals and I/O

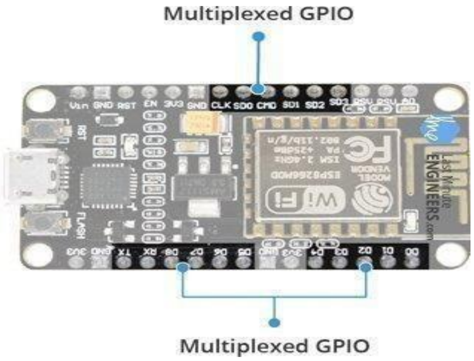
The ESP8266 NodeMCU has total 17 GPIO pins broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:

· ADC channel – A 10-bit ADCchannel.

· UART interface – UART interface is used to load codeserially.
· PWM outputs – PWM pins for dimming LEDs or controllingmotors.
· SPI, I2C & I2S interface – SPI and I2C interface to hook up all sorts of sensors andperipherals. · I2S interface – I2S interface if you want to add sound to yourproject.

Multiplexed I/O

- 1 ADCchannels.
- UART interfaces.
- PWM outputs.
- SPI, I2C & I2Sinterface.



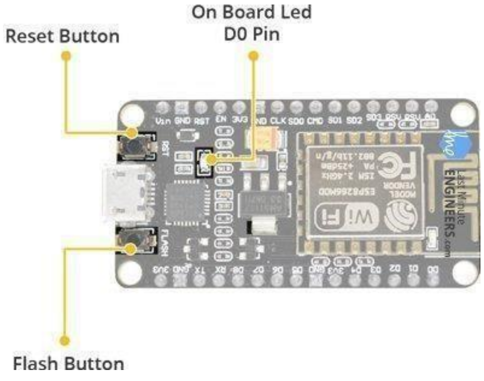
Thanks to the ESP8266's pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as PWM/UART/SPI.

On-board Switches & LED Indicator

The ESP8266 NodeMCU features two buttons. One marked as RST located on the top left corner is the Reset button, used of course to reset the ESP8266 chip. The other FLASH button on the bottom left corner is the download button used while upgradingfirmware.

Switches & Indicators

· RST – Reset the ESP8266chip
· FLASH – Download newprograms
· Blue LED – UserProgrammable

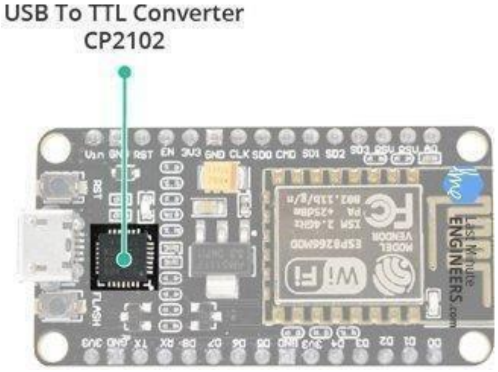


The board also has a LED indicator which is user programmable and is connected to the D0 pin of the board.

Serial Communication

The board includes CP2102 USB-to-UART Bridge Controller from Silicon Labs, which converts USB signal to serial and allows your computer to program and communicate with the ESP8266 chip.

· Serial Communication
· CP2102 USB-to-UART converter
· 4.5 Mbps communicationspeed
· Flow Control support

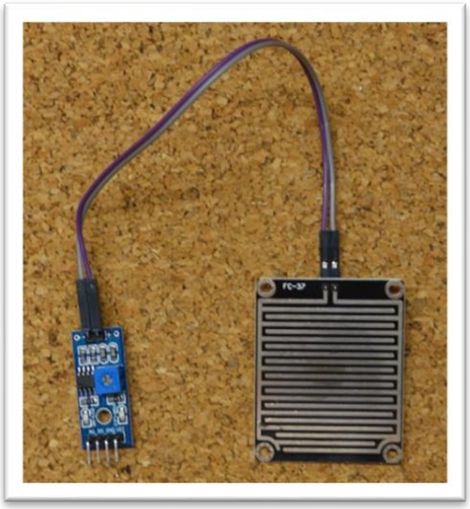

USB To TTL Converter
CP2102
Vin GND RST EN 3V3 GND CLK SD0 CMD SD1 SD2 SD3 RSV RSV A0
RST
FLASH
3V3 GND TX RX D8 D7 D6 D5 GND 3V3 D4 D3 D2 D1 D0
Last Minute ENGINEERS.com

If you have an older version of CP2102 driver installed on your PC, we recommend upgrading now.

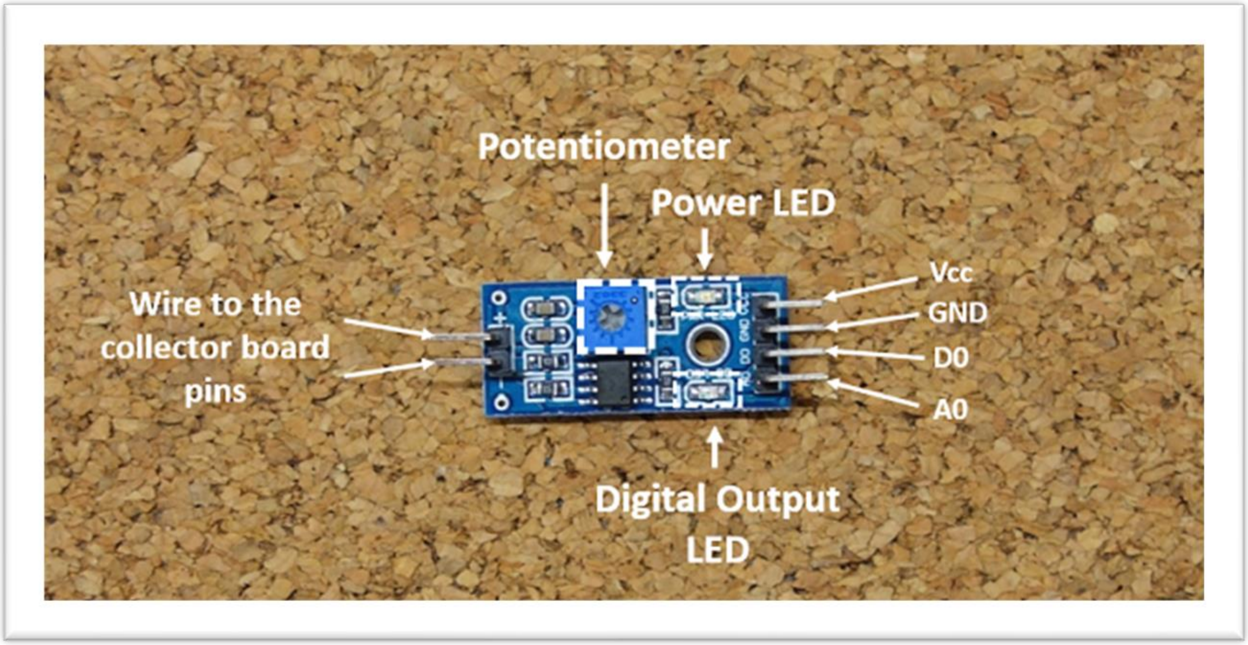
FC-37 RAINDROP SENSOR

The rain sensor is used to detect water and it can detect beyond of what a humidity sensor do. This article explains how to use the FC-37 rain sensor module with the Arduino.

The FC-37 rain sensor (or other versions like YL-83) is set up by two pieces: the electronic board (at the left) and the collector board (at the right) that collects the water drops, as you can see in the following figure:



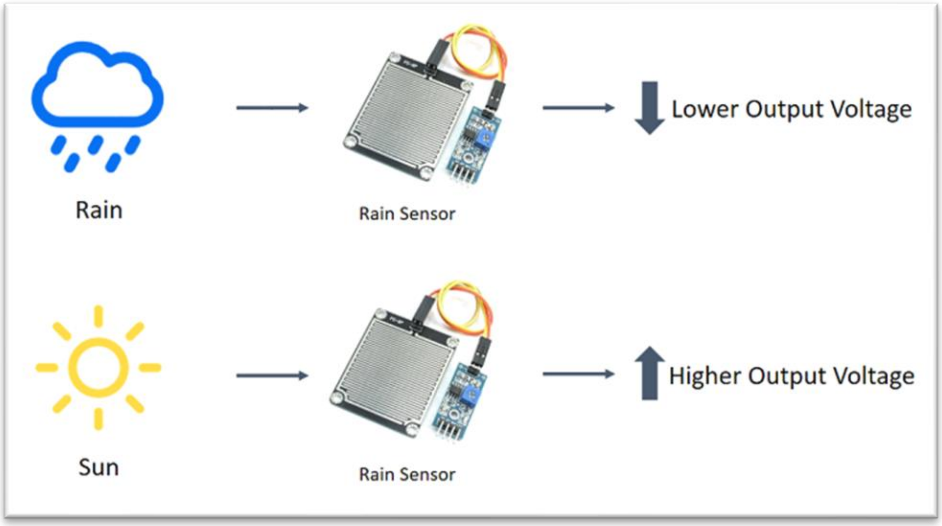
The rain sensor has a built-in potentiometer for sensitivity adjustment of the digital output (D0). It also has a power LED that lights up when the sensor is turned on and a digital output LED.



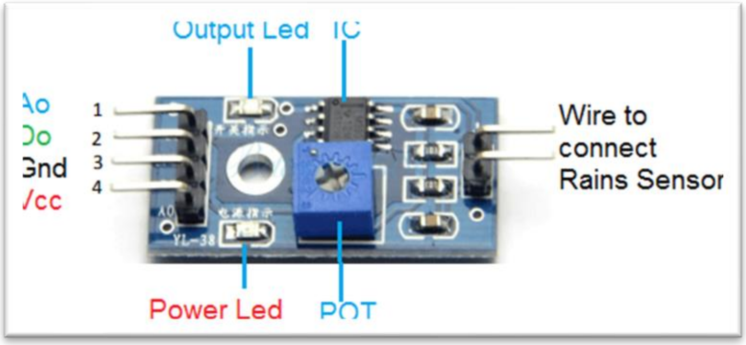
How does it work?

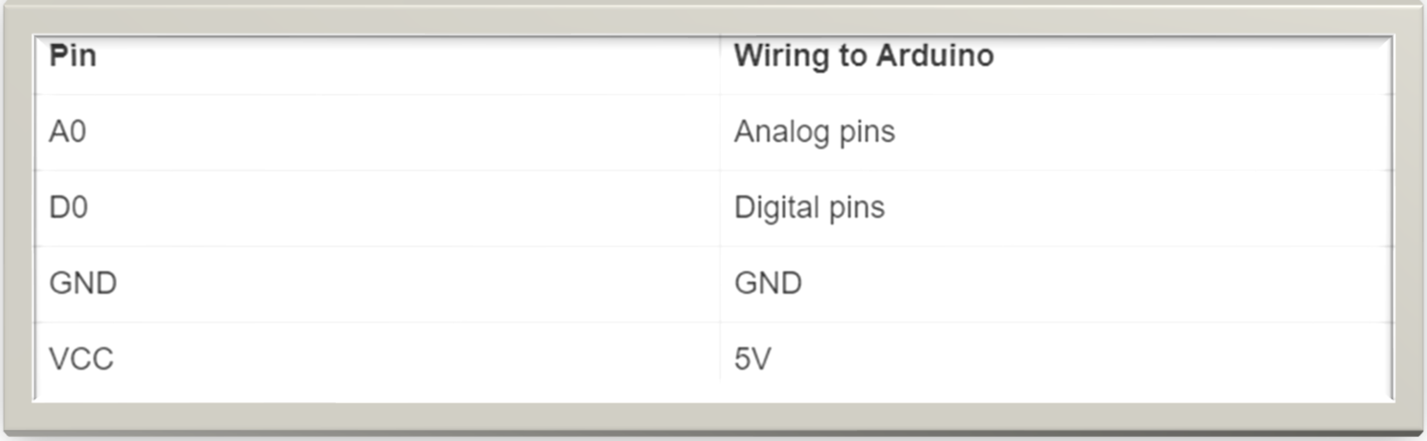
Basically, the resistance of the collector board varies accordingly to the amount of water on its surface.
When the board is:

- **Wet:** the resistance increases, and the output voltage decreases
- **Dry:** the resistance is lower, and the output voltage is higher



PIN DETAILS OF RAIN SENSOR

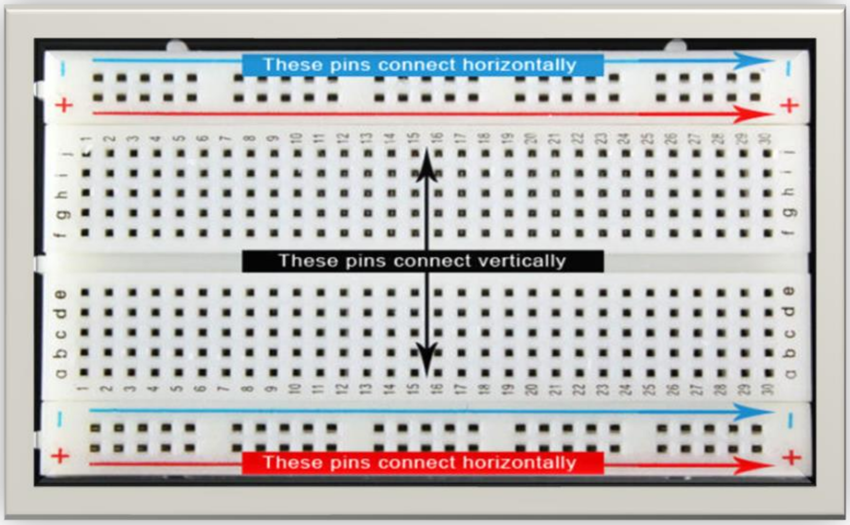




Pin	Wiring to Arduino
A0	Analog pins
D0	Digital pins
GND	GND
VCC	5V

BREAD BOARD

A breadboard is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used for slicing bread.[1] In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these. Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. Older breadboard types did not have this property. A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).Compared to more permanent circuit connection methods, modern breadboards have high parasitic capacitance, relatively high resistance, and less reliable connections, which are subject to jostle and physical degradation. Signaling is limited to about 10 MHz, and not everything works properly even well below that frequency. A common use in the system on a chip (SoC) era is to obtain an microcontroller (MCU) on a pre-assembled printed circuit board (PCB) which exposes an array of input/output (IO) pins in a header suitable to plug into a breadboard, and then to prototype a circuit which exploits one or more of the MCU's peripherals, such as general-purpose input/output (GPIO) UART/USART serial transceivers, analog-to digital converter (ADC), digital-to-analog converter (DAC), pulse-width modulation .A breadboard is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used for slicing bread.[1] In the 1970s on a pre-assembled printed circuit board (PCB) which exposes an array of input/output (IO) pins in a header suitable to plug into a breadboard, and then to prototype a circuit which exploits one or more of the MCU's peripherals, such as general-purpose input/output (GPIO) UART/USART serial transceivers, analog-to digital converter (ADC), digital-to-analog converter (DAC), pulse- width modulation.


These pins connect horizontally
These pins connect vertically
These pins connect horizontally

JUMPER WIRE

A jump wire (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

MICRO USB CABLE

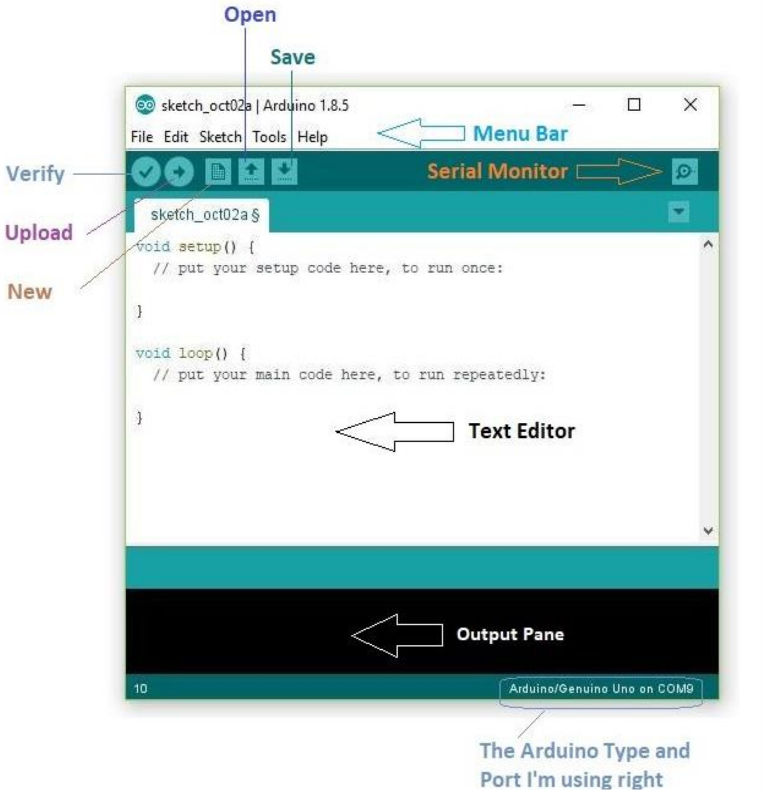
Universal Serial Bus (USB) was developed in the 1990s in an effort to simplify the connections between computers and peripheral devices. It has become widely popular due to its compatibility with many platforms and operating systems, its low cost of implementation, and its ease of use. Most computers that are built today come with several USB ports, and USB is the interface of choice for most home and office peripherals including printers, cameras, modems, and portable storage devices.

USB standards are developed and maintained by an industry body called the USB Implementers Forum (USB-IF). In its original specification, USB defined only two connector types: A and B. Revisions to the specification and demands on manufacturers have expanded the breadth of connectors used for USB devices, but the majority of USB products still use these A and B connector interfaces

SOFTWARE COMPONENTS DESCRIPTION

ARDUINO IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, and Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

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 toolchain, 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'sfirmware.

WINDOWS OPERATING SYSTEM

Microsoft Windows, also called Windows and Windows OS, computer operating system (OS) developed by Microsoft Corporation to run personal computers (PCs). Featuring the first graphical userinterface(GUI) for IBM-compatible PCs, the Windows OS soon dominated the PC market.The first version of Windows, released in 1985, was simply a GUI offered as an extension of Microsoft’s existing disk operatingsystem.

THINGSPEAK

ThingSpeak is a platform providing various services exclusively targeted for building IoT applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plugins and apps for collaborating with web services, social network and other APIs. We will consider each of these features in detail below.

The core element of ThingSpeak is a ‘ThingSpeak Channel’. A channel stores the data that we send to ThingSpeak and comprises of the below elements:

- 8 fields for storing data of any type - These can be used to store the data from a sensor or from an embedded device.
- 3 location fields - Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.
- 1 status field - A short message to describe the data stored in the channel.

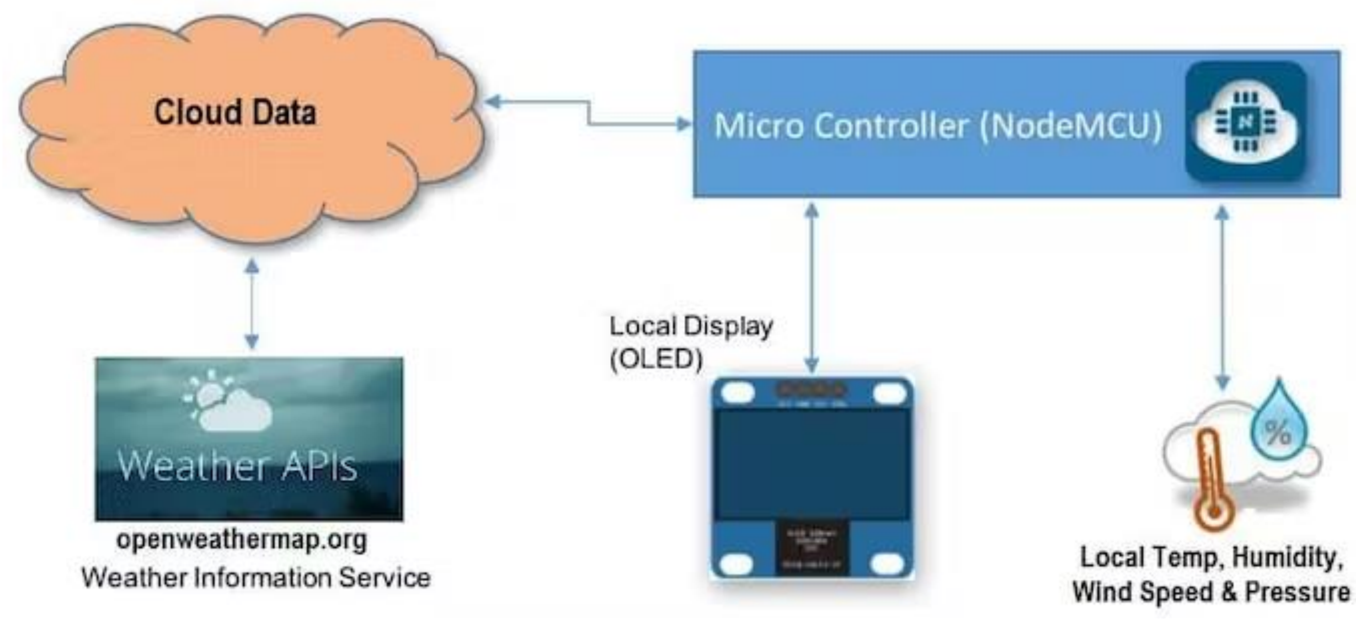
To use ThingSpeak, we need to signup and create a channel. Once we have a channel, we can send the data, allow ThingSpeak to process it and also retrieve the same.

JUPYTER NOTEBOOK

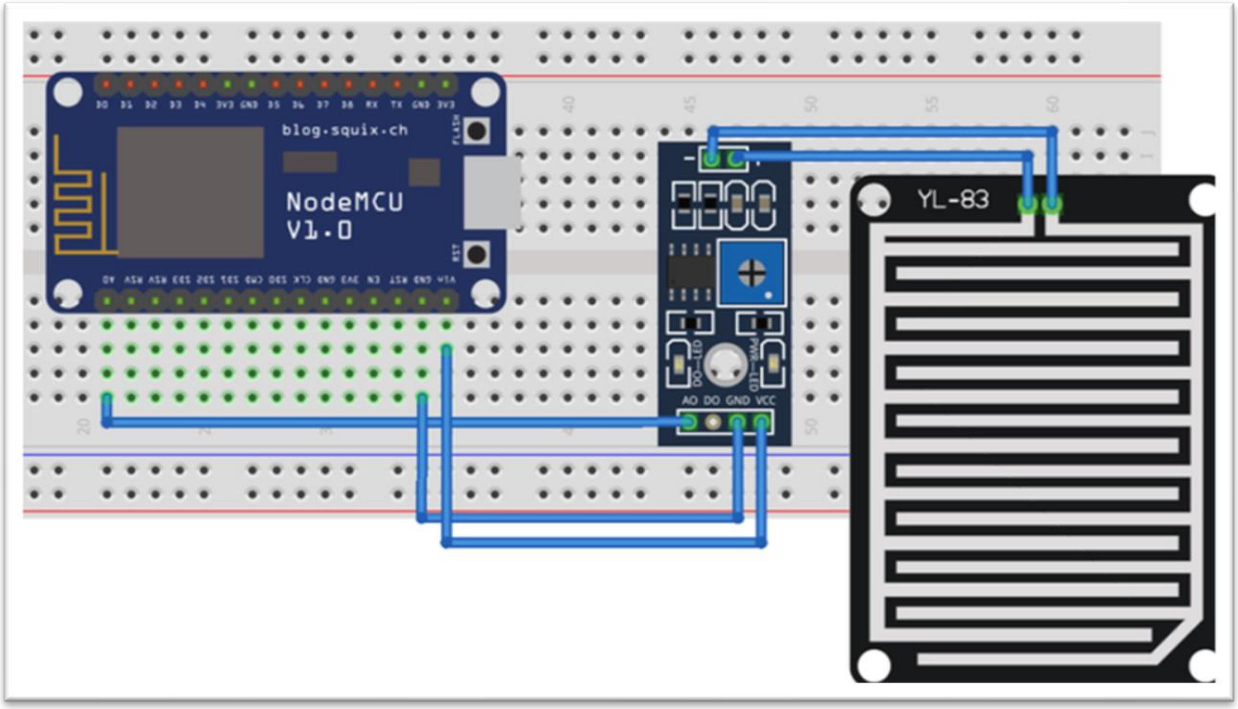
The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

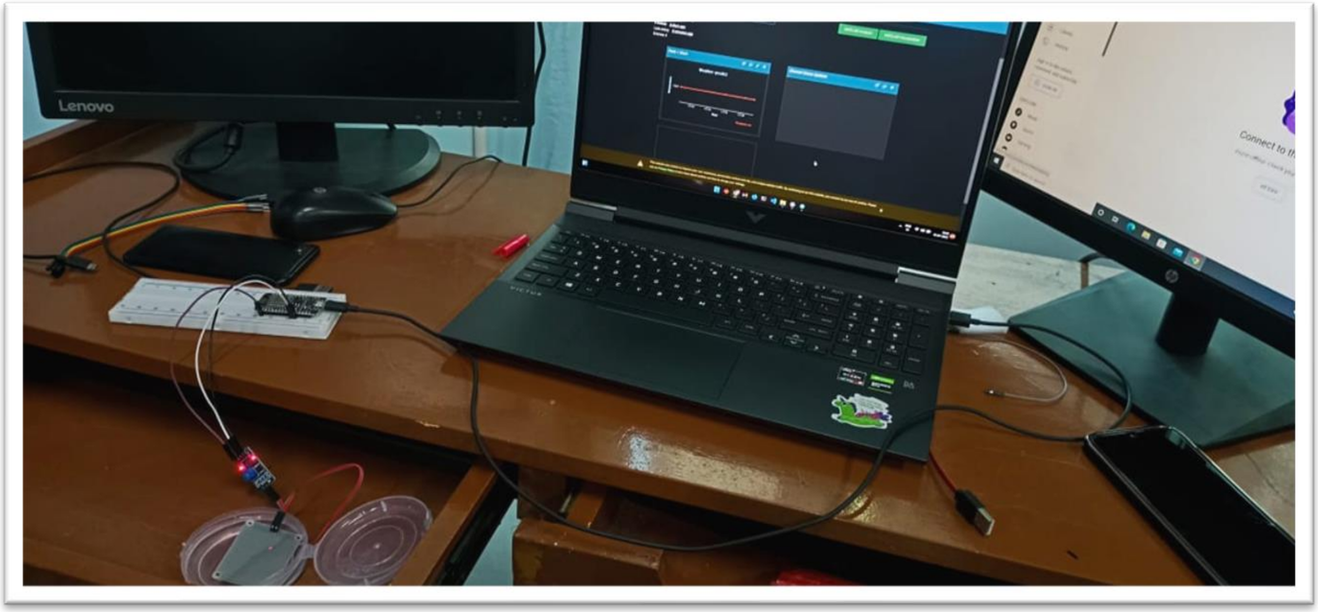
PROJECT BLOCK DIAGRAM



CIRCUIT DIAGRAM


D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
blog.squix.ch
FLASH
NodeMCU
V1.0
RST
A0 RSV RSV SD3 SD2 SD1 CMD SD0 CLK GND 3V3 EN RST GND Vin
40
45
50
55
60
20
50
DO-LED
PWR-LED
AO DO GND VCC
YL-83

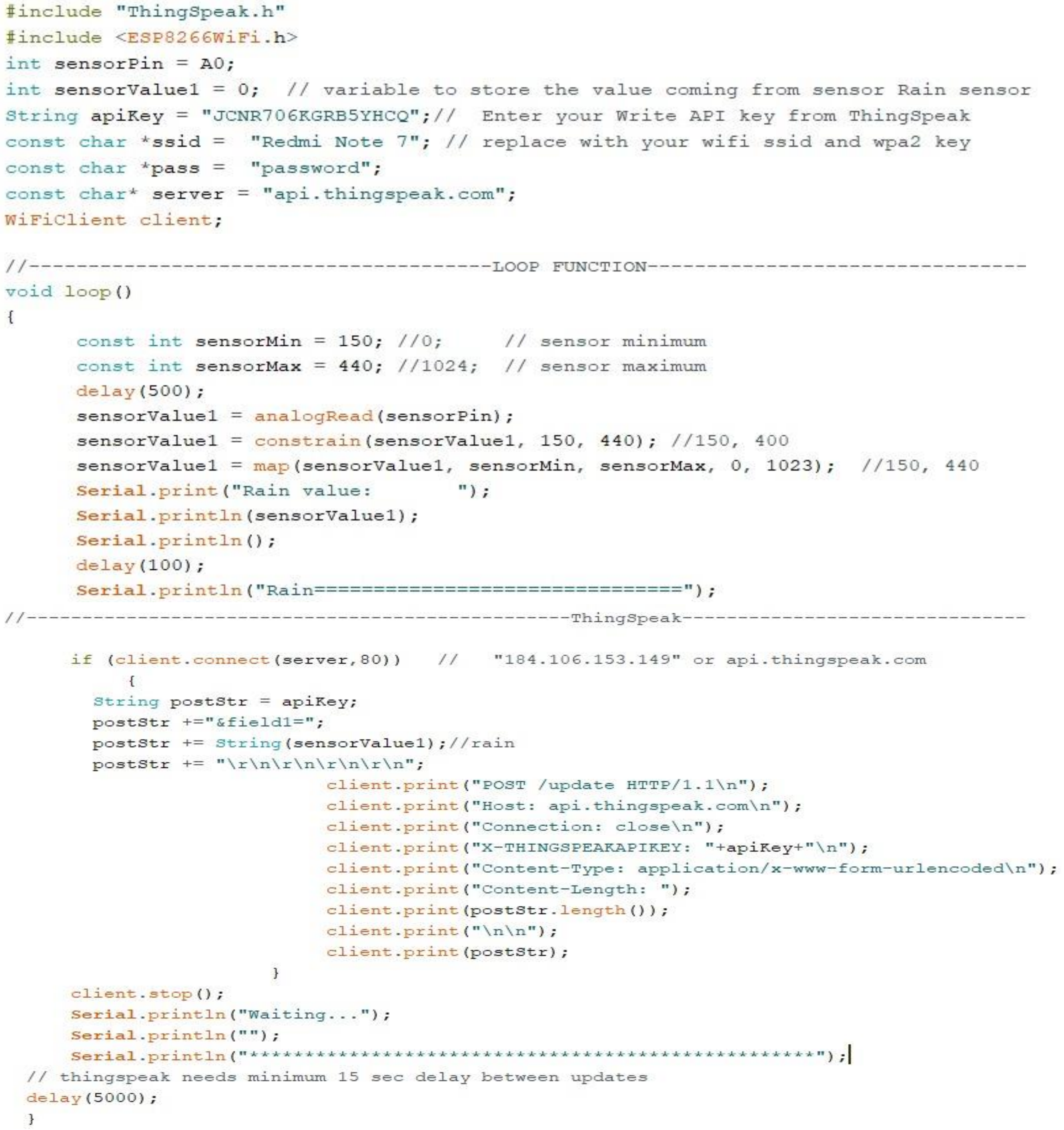
PROJECT PICTURE



WORKING PRINCIPLE

1. In the whole process, we implemented IoT based Live Weather Station Monitoring system using NODEMCU ESP8266.
2. We will interface FC37 raindrop sensor and upload the data to the ThingSpeak cloud server to the smart phone via WiFi module inbuilt in NodeMCU ESP8266.
3. A Integrated TCP/IP protocol stack is used for transmitting and receiving sensor information. Depending on a status of weather information to the wireless remote location. The NodeMCU 8266 controls the entire Weather Prediction system status on a Thing Speak Cloud server.
4. Next, to send data to ThingSpeak, we need a **unique API key**, which we will use later in our code to upload our sensor data to Thingspeak Website.
5. In this way, a secure, flexible, trust-able and economical system is developed to solve mentioned weather parameters.
6. Also, we can check whether data through anywhere using Internet as we hosted this server publicly.

PROJECT CODE



```
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
int sensorPin = A0;
int sensorValue1 = 0;  // variable to store the value coming from sensor Rain sensor
String apiKey = "JCNR706KGRB5YHCQ";//  Enter your Write API key from ThingSpeak
const char *ssid =  "Redmi Note 7"; // replace with your wifi ssid and wpa2 key
const char *pass =  "password";
const char* server = "api.thingspeak.com";
WiFiClient client;

//-----------------------------------------LOOP FUNCTION--------------------------------
void loop()
{
      const int sensorMin = 150; //0;     // sensor minimum
      const int sensorMax = 440; //1024;  // sensor maximum
      delay(500);
      sensorValue1 = analogRead(sensorPin);
      sensorValue1 = constrain(sensorValue1, 150, 440); //150, 400
      sensorValue1 = map(sensorValue1, sensorMin, sensorMax, 0, 1023);  //150, 440
      Serial.print("Rain value:        ");
      Serial.println(sensorValue1);
      Serial.println();
      delay(100);
      Serial.println("Rain================================");
//-----------------------------------------------------ThingSpeak-------------------------------

     if (client.connect(server,80))   //   "184.106.153.149" or api.thingspeak.com
          {
       String postStr = apiKey;
       postStr +="&field1=";
       postStr += String(sensorValue1);//rain
       postStr += "\r\n\r\n\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);
                         }
     client.stop();
     Serial.println("Waiting...");
     Serial.println("");
     Serial.println("***************************************************");
  // thingspeak needs minimum 15 sec delay between updates
  delay(5000);
  }
```

CODE DESCRIPTION

- ESP8266WIFI library provides ESP8266 specific WiFi routines that we are calling to connect to the network.

- Const char* ssid means the SSID (name) of the WiFi network you want to connect to const char*

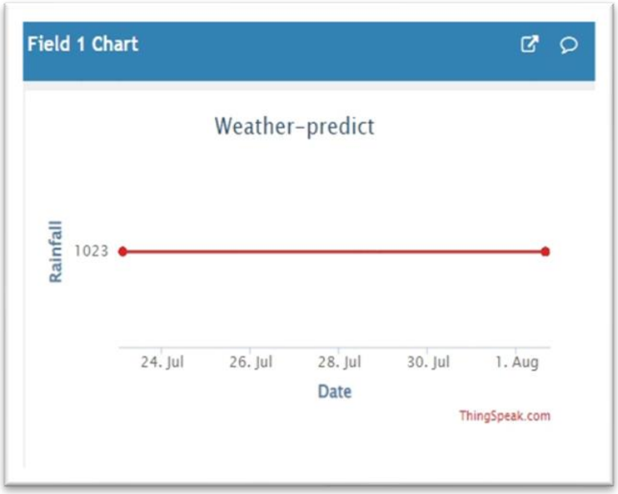
- Const char* password- the password() function verifies or changes the username password or the password phrase. The oldpass is the current password or password phrase for user username ,and is a NULL-terminated character string of a password or a password phrase.

- Serial.println-prints data to the serial port as human readable ASCII text followed by a carriage return character(ASCII 13,or “\r”) and a newline character(ASCII 10,or “\n”).

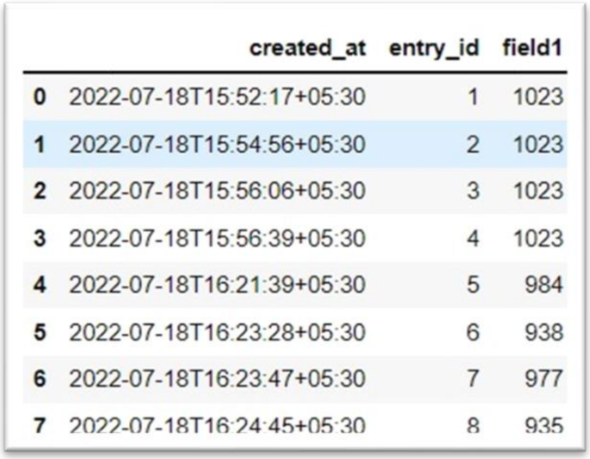
- Delay(5000) represents the time in milliseconds the program has to wait until moving to the next line of code.

RESULTS

WEATHER CHART:



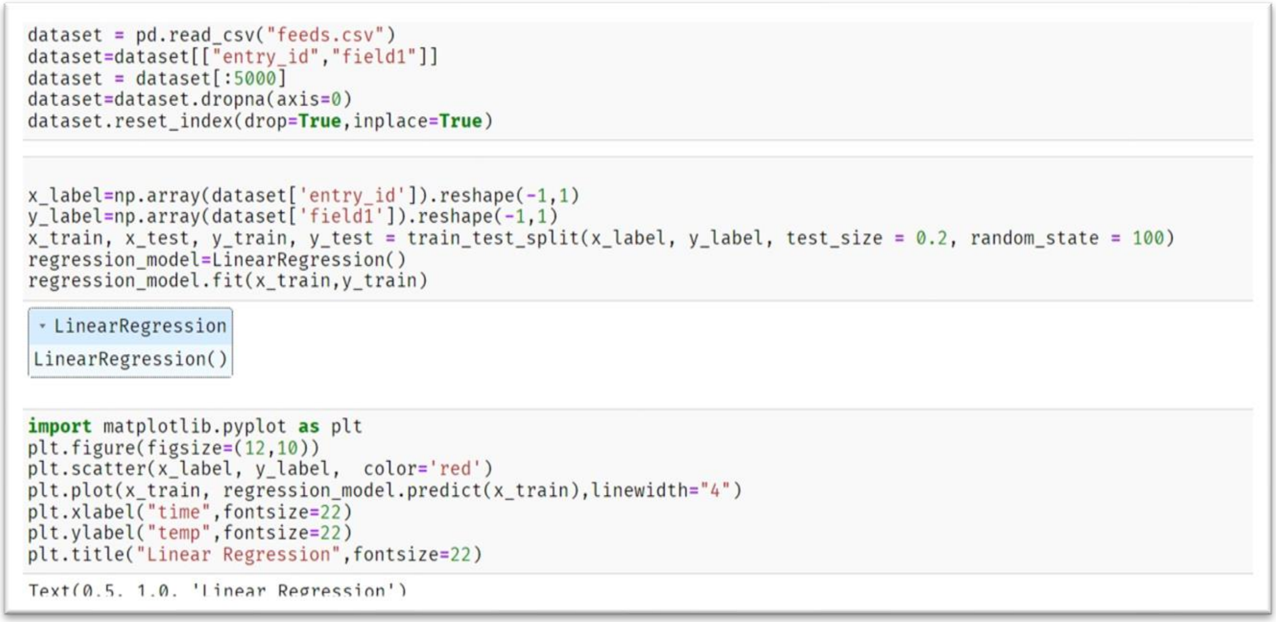
DATAFRAME:



	created_at	entry_id	field1
0	2022-07-18T15:52:17+05:30	1	1023
1	2022-07-18T15:54:56+05:30	2	1023
2	2022-07-18T15:56:06+05:30	3	1023
3	2022-07-18T15:56:39+05:30	4	1023
4	2022-07-18T16:21:39+05:30	5	984
5	2022-07-18T16:23:28+05:30	6	938
6	2022-07-18T16:23:47+05:30	7	977
7	2022-07-18T16:24:45+05:30	8	935

MACHINE LEARNING MODEL

DATASET USING MATPLOT LIBRARY:



```
dataset = pd.read_csv("feeds.csv")
dataset=dataset[["entry_id","field1"]]
dataset = dataset[:5000]
dataset=dataset.dropna(axis=0)
dataset.reset_index(drop=True,inplace=True)
```

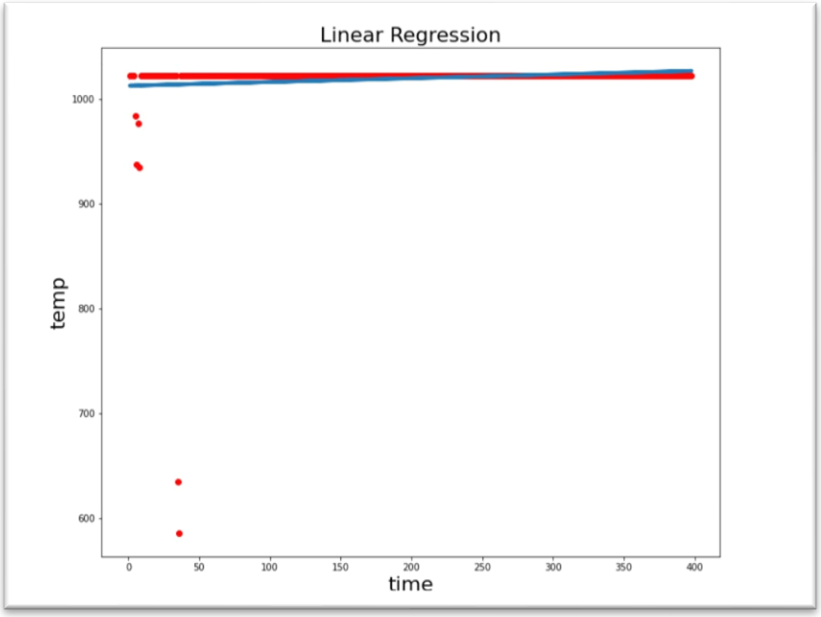
```
x_label=np.array(dataset['entry_id']).reshape(-1,1)
y_label=np.array(dataset['field1']).reshape(-1,1)
x_train, x_test, y_train, y_test = train_test_split(x_label, y_label, test_size = 0.2, random_state = 100)
regression_model=LinearRegression()
regression_model.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(12,10))
plt.scatter(x_label, y_label,  color='red')
plt.plot(x_train, regression_model.predict(x_train),linewidth="4")
plt.xlabel("time",fontsize=22)
plt.ylabel("temp",fontsize=22)
plt.title("Linear Regression",fontsize=22)
```

```
Text(0.5, 1.0, 'Linear Regression')
```

Linear Regression:



ADVANTAGES

- ✓ The Smart Weather Monitoring System Is Quite Small As Compared To Conventional Weather Monitoring System And Can Be Installed Easily.
- ✓ Power Requirements For Smart Weather Monitoring Systems Are Quite Low.
- ✓ Sensors Used In The Smart Weather Monitoring System Are Quite Cheaper Which Makes This Project Quite Cost Effective.
- ✓ Data From The Sensors Can Be Sent To The Web Page Also Which Can Be Accessed From Anywhere In The World.
- ✓ Maintenance Cost Of The Smart Weather Monitoring System Is Also Quite Low Due To The Presence Of Fewer Parts.
- ✓ The Data Collected And Analyzed By The Sensors Of The Smart Weather Monitoring System Predicts An Accurate Outcome Of The Weather.
- ✓ Due To The High Speed Of These Sensors, They Can Also Easily Detect Any Sudden Change In The Forecast As Well.
- ✓ Prior Alert Of The Weather Conditions Is Also Possible In Case Of Smart Weather Monitoring System. This Prior Alert Can Be Seen Easily By Visiting A Website Of An IoT Portal.

DISADVANTAGES

- ✓ The bulky machinery of conventional Weather Monitoring Systems requires constant monitoring and should be changed routinely. This bulky machinery also increases the cost of installation.
- ✓ This system also has a high-power consumption which increases its cost.
- ✓ Data is transferred manually.
- ✓ The existing Weather System predicts the weather and the sudden change in the forecast with some delay.

FUTURE SCOPES

The IOT Based Weather Station System will proposed to Real time Applications. It doesn’t need of any data centers physically because of we are creating a data server in Cloud so that it doesn’t require any datacenter further. So it reduces the cost of equipment. Many of the innovative researchers are interested towards The IoT based real time spplications. So this system will helps to the researcher for their further investigation of weather details.

The IOT based Weather Station System not only displays the weather parameters like Temperature, Altitude, Humidity and Pressure etc., but it also displays the weather location, Industry, Time and other weather information from this we can forecasts the weather details

.

CONCLUSION

In this paper, to make this idea, one need to take genuine help of electronic sensor devices which are needed to place in the environment. By using this sensor, we can stream real-time data over the web server using ESP8266. We also required one dedicated public IP to available this server over the open Internet. The excellent and low-cost weather are monitoring real-time system presented in this paper. A integrated TCP/IP protocol stack is used for transmitting and receiving sensor.

REFERENCES

1. https://forum.arduino.cc/t/help-with-a-thingspeak-arduino-nodemcu-8266-program/998947
2. https://circuitdigest.com
3. https://www.hackster.io/pranavkhatale/iot-based-weather-station-using-thingspeak-iot-analytics-ef1174
4. https://electronics-project-hub.com/category/internet-of-things-iot/
5. https://pianalytix.com/weather-reporting-system-using-iot/#:~:text=These%20limitations%20are%3A,Data%20is%20transferred%20manually.
6. https://nothans.com/thingspeak-tutorials/introduction-to-the-internet-of-things-and-thingspeak
7. https://social.technet.microsoft.com/wiki/contents/articles/34245.iot-based-weather-informative-system.aspx