**CHAPTER-3**

**INTERNET OF THINGS**

IOT means Internet of Things, the main aim of our project AWS Cloud and Network Security is to retrieve the data of cloud from anywhere at any time, if we have Internet Connection. In this project we are using sensors for getting information about temperature, humidity, rainfall at a particular location.

In this process, we have used different types of sensors they are For temperature and humidity information we are using DHT11 sensor. This is connected with Nodemcu.

# **3.1 DHT11 - Humidity and Temperature Sensor**

The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed).

It’s fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds.

# **Features**

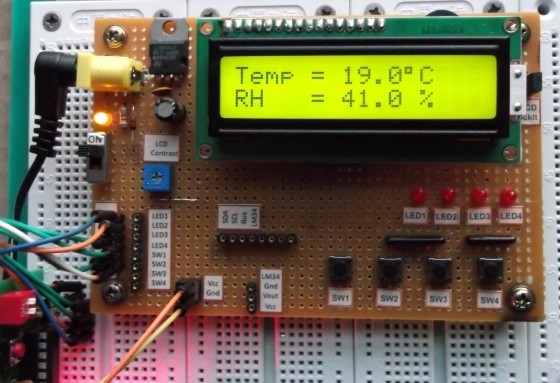
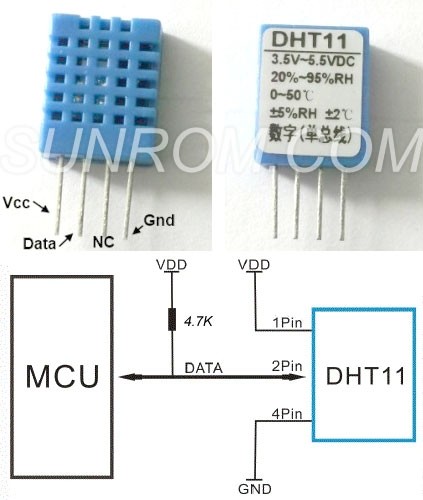
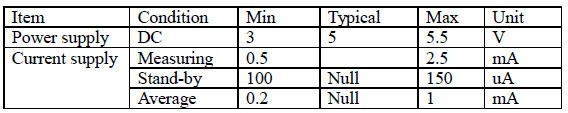
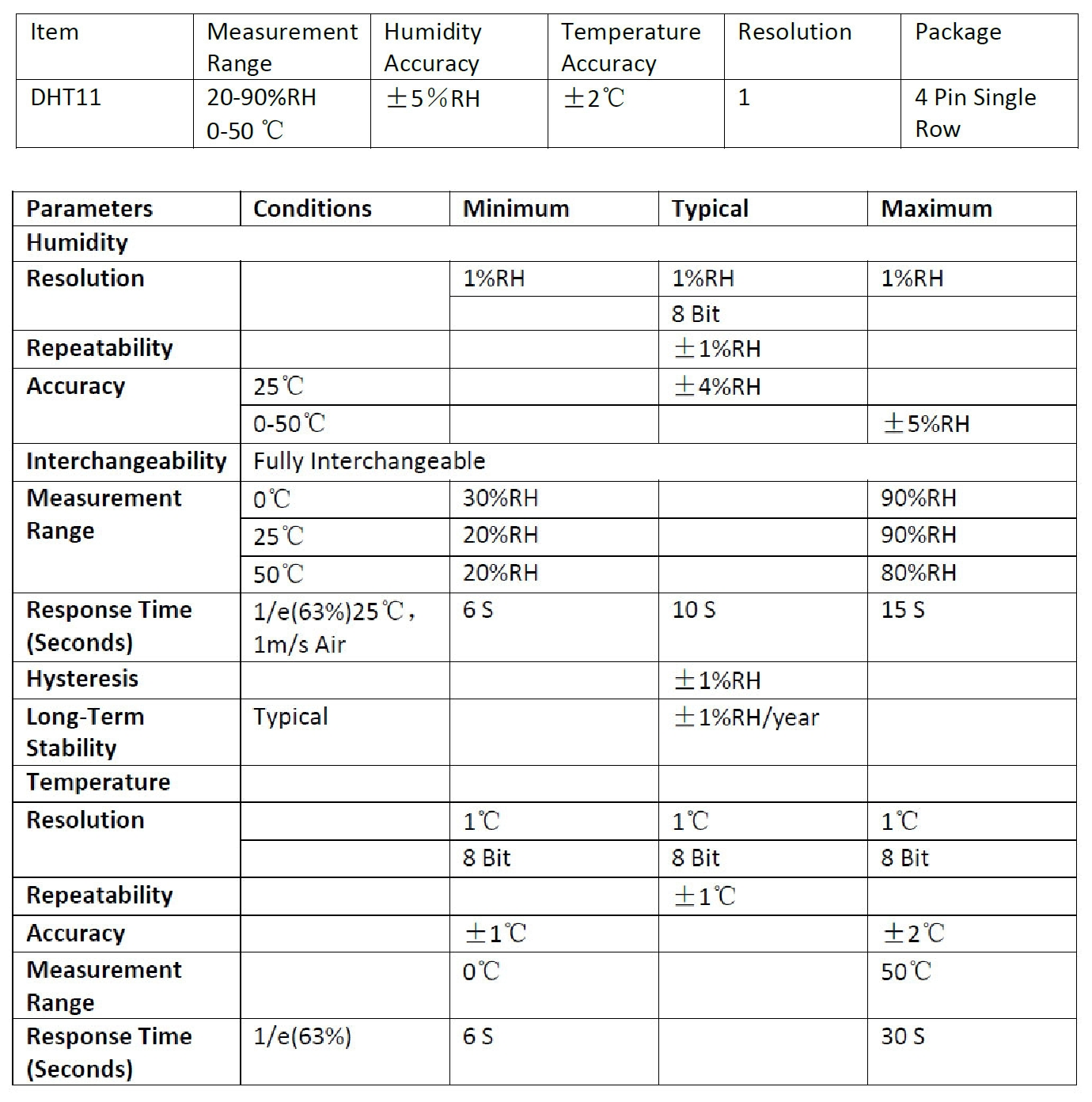
* Full range temperature compensated
* Relative humidity and temperature measurement
* Calibrated digital signal
* Outstanding long-term stability
* Extra components not needed
* Long transmission distance
* Low power consumption
* 4 pins packaged and fully interchangeable

# **Details**

This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness. Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor’s internal signal detecting process.

The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package.

# **3.2 Specifications**



DHT11’s power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

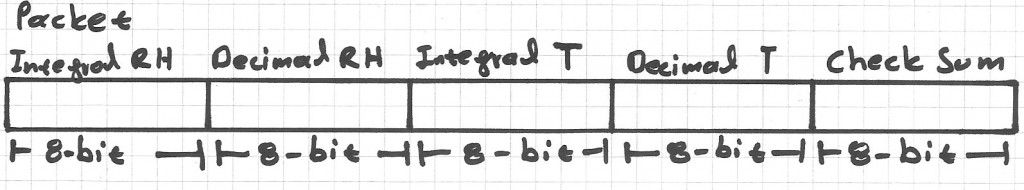
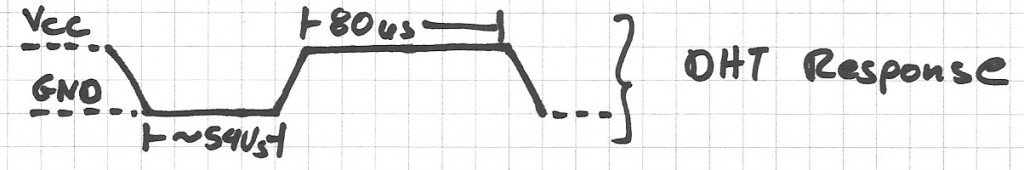
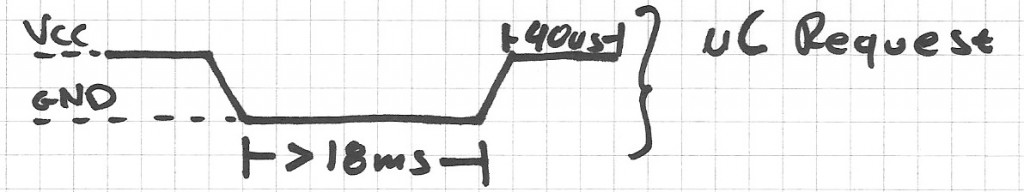
## SDK (Software Development Kit)

Download source code + project articles by clicking following link

<http://www.sunrom.com/files/3732.zip>

It contains details for AVR, PIC and Arduino projects.

# **Communication Process: Serial Interface (Single-Wire Two-Way)**



The interesting thing in this module is the protocol that uses to transfer data. All the sensor readings are sent using a single wire bus which reduces the cost and extends the distance. In order to send data over a bus you have to describe the way the data will be transferred, so that transmitter and receiver can understand what says each other. This is what a protocol does. It describes the way the data are transmitted. On DHT-11 the 1-wire data bus is pulled up with a resistor to VCC. So if nothing is occurred the voltage on the bus is equal to VCC.

Communication Format can be seperated into three stages

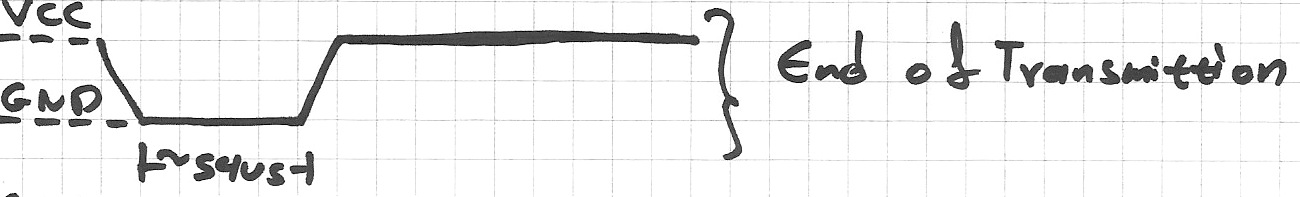
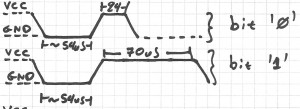
* 1. Request
  2. Response
  3. Data Reading

1. **Request:** To make the DHT-11 to send you the sensor readings you have to send it a request. The request is, to pull down the bus for more than **18ms** in order to give DHT time to understand it and then pull it up for **40uS**.
2. **Response:** What comes after the request is the DHT-11 response. This is an automatic reply from DHT which indicates that DHT received your request. The response is ~54uS low and 80uS high.
3. **Data Reading:** What will come after the response is the sensor data. The data will be packed in a packet of 5 segments of 8-bits each. Totally 5×8 =40bits.

First two segments are Humidity read, integral & decimal. Following two are Temperature read in Celsius, integral & decimal and the last segment is the Check Sum which is the sum of the 4 first

1. Sunrom Technologies Your Source for Embedded Systems Visit us at [www.sunrom.com](http://www.sunrom.com/)

segments. If Check Sum's value isn't the same as the sum of the first 4 segments that means that data received isn't correct.

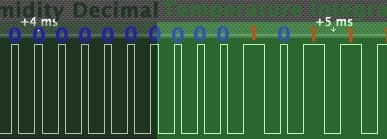
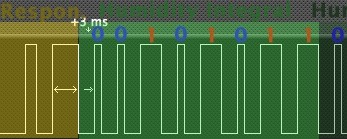


**How to Identify Bits:** Each bit sent is a follow of ~54uS Low in the bus and ~24uS to 70uS High depending on the value of the bit.

Bit '0' : ~54uS Low and ~24uS High Bit '1' : ~54uS Low and ~70uS High

**End Of Frame:** At the end of packet DHT sends a ~54uS Low level, pulls the bus to High and goes to sleep mode.

**Logic Analyzer Snapshots:** In the following image you can see the request sent from the MCU to the DHT and following the packet. Because the request has very long duration as you can see is about 20mS and packet received is in uS we can't view the data bits. So it is exapanded in next view.



If we zoom at the data bits we can read the values. You can see after the Request follows the Response, and Data bits. I have drawn some color notes to be more understandable.

If we decode the above data we have.

Humidity 0b00101011.0b00000000 = 43.0% (43 is integral part and .0 is decimal part) Temperature 0b00010111 = 23 C.

The last two segments can't be seen in this image because of zoom.

## Implementation:

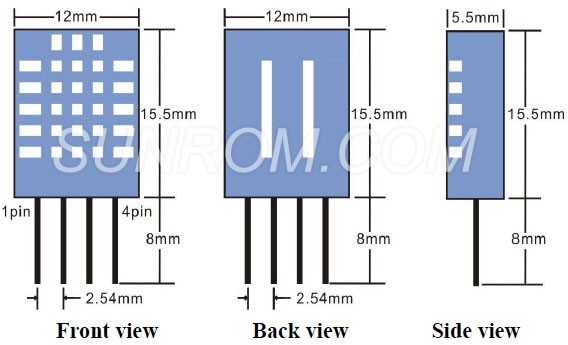
What we have to do to read a DHT-11 sensor is:

* 1. Send request
  2. Read response
  3. Read each data segment and save it to a buffer
  4. Sum the segments and check if the result is the same as CheckSum

If the CheckSum is correct, the values are correct so we can use them. If CheckSum is wrong we discard the packet.

To read the data bits can use a counter and start count uSeconds of High level. For counts > 24uS we replace with bit '1'. For counts <=24 we replace with bit'0'

1. Sunrom Technologies Your Source for Embedded Systems Visit us at [www.sunrom.com](http://www.sunrom.com/)



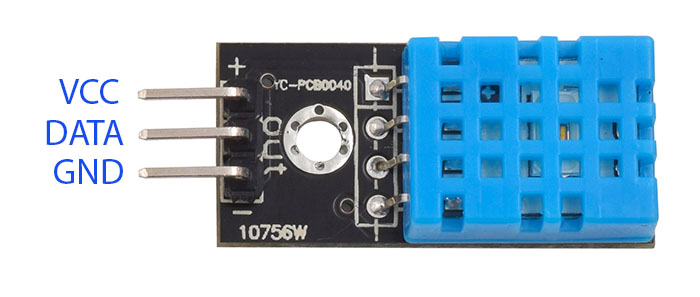
**3.3 DHT11 temperature and Humidity sensor**

DHT11 sensor is used for measuring both humidity and temperature values. It can measure relative humidity in percentage (20 to 90% RH) and temperature in degree Celsius in the range of 0 to 50°C.It has 3 pins. They are:

1. VCC

2. DATA OUT

3. GND



**Fig 6.2: DHT 11**

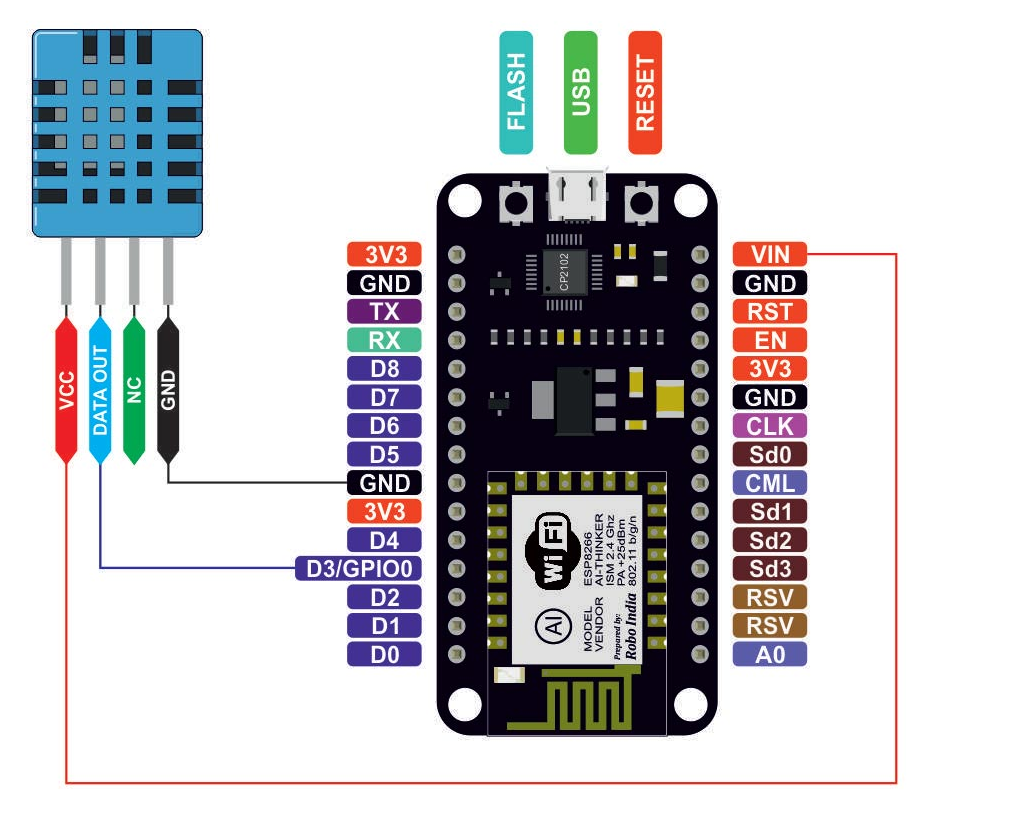
# Nodemcu:



**Fig 6.3: Node MCU**

NodeMCU Dev. Kit/board consist of ESP8266 Wi-Fi enabled chip. The **ESP8266** is a low-cost [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) chip developed by Espressif Systems with TCP/IP protocol. Node MCU Dev. Kit has **Arduino like** Analog (i.e. A0) and Digital (D0-D8) pins on its board. It supports serial communication protocols i.e. UART, SPI, I2C etc. Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc.

# 3.4 DHT11 SENSOR CONNECTION WITH NODEMCU:



**Fig 6.4: Pin Diagram of Node MCU**

**Connection of Node Mcu With Dht11 Sensor:**  
**Pin 1** of the DHT11 goes into **+3v** of the Node MCU.

**Pin 2** of the DHT11 goes into Digital Pin **D3** of the Node MCU.

**Pin 3** of the DHT11 goes into Ground Pin (**GND**) of the Node MCU.

**6.1.3** **Rain Drop Sensor**

The rain sensor module is an easy tool for rain detection. It can be used as a switch when raindrop falls through the raining board and also for measuring rainfall intensity. The module features, a rain board and the control board that is separate for more convenience, power indicator LED and an adjustable sensitivity though a potentiometer

The analog output is used in detection of drops in the amount of rainfall. Connected to 5V power supply, the LED will turn on when induction board has no rain drop, and DO output is high. When dropping a little amount water, DO output is low, the switch indicator will turn on. Brush off the water droplets, and when restored to the initial state, outputs high level.

**Specifications**

• Adopts high quality of RF-04 double sided material.

• Area: 5cm x 4cm nickel plate on side.

• Anti-oxidation, anti-conductivity, with long use time.

• Comparator output signal clean waveform is good, driving ability, over 15mA.

• Potentiometer adjust the sensitivity.

• Working voltage 5V.

• Output format: Digital switching output (0 and 1) and analog voltage output AO.

• With bolt holes for easy installation.

• Small board PCB size: 3.2cm x 1.4cm.

• Uses a wide voltage LM393 comparator.

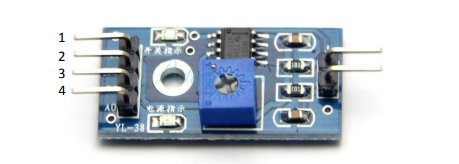
Pin Configuration

1. VCC: 5V DC

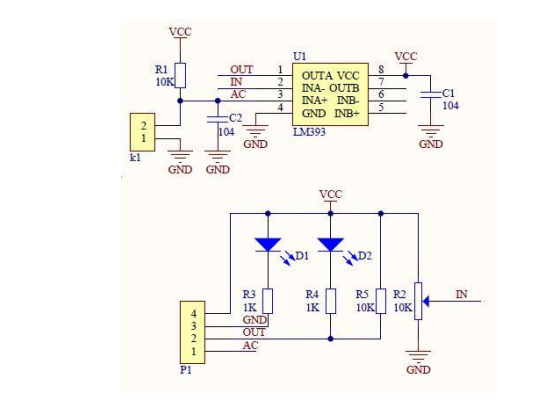
2. GND: ground

3. DO: high/low output

4. AO: analog output 1 2 3 4

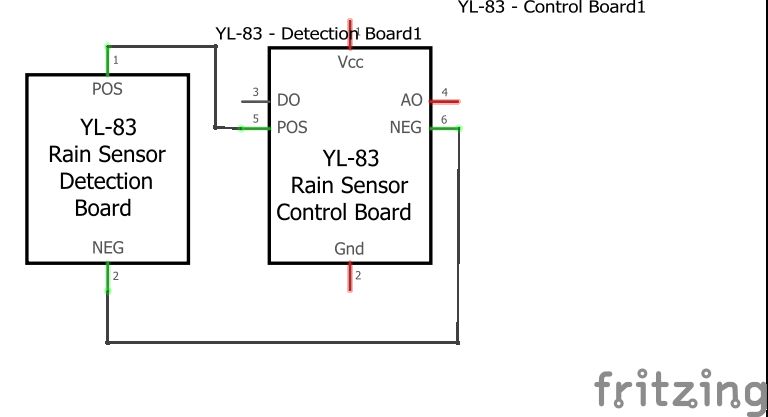


**Schematic Diagram**



**Fig 6.5:** **Schematic Diagram**

**Pin configuration of get spares rain drop sensor:**



**Fig**

**6.6:** **Pin Configuration of Raining Sensor**

**Connecting sensor on bread board:**

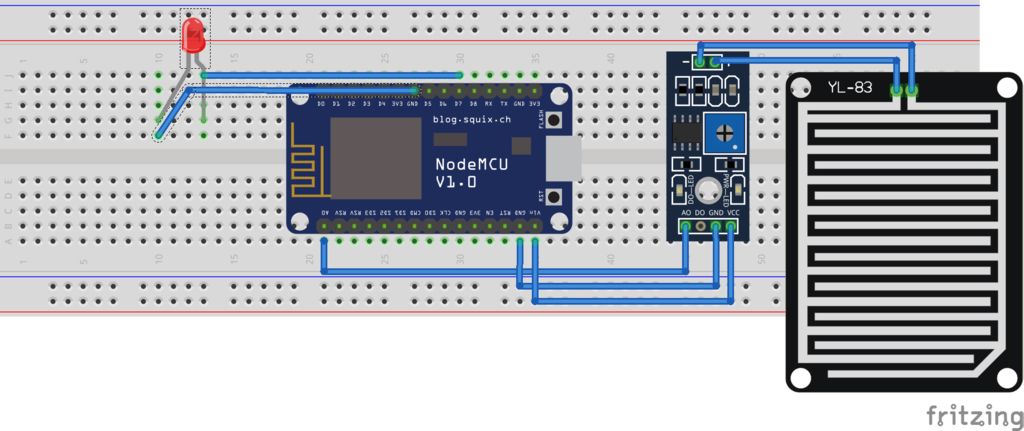


Fig 6.7: Connecting of Rain Sensor with Node MCU

**Working Principle of Raindrop Sensor**

[](https://cdn.instructables.com/FZF/E2NC/JK2UIKLW/FZFE2NCJK2UIKLW.LARGE.jpg)

Raindrop sensor is basically a board on which nickel is coated in the form of lines. It works on the principal of resistance. When there is no rain drop on board. Resistance is high so we get high voltage according to V=IR. When rain drop present it reduces the resistance because water is conductor of electricity and presence of water connects nickel lines in parallel so reduced resistance and reduced voltage drop across it.



[www.vaisala.com](http://www.vaisala.com/)

# Yl-83 Rain Detector

Heating element for keeping sensor free of snow and condensed moisture, and for quick drying

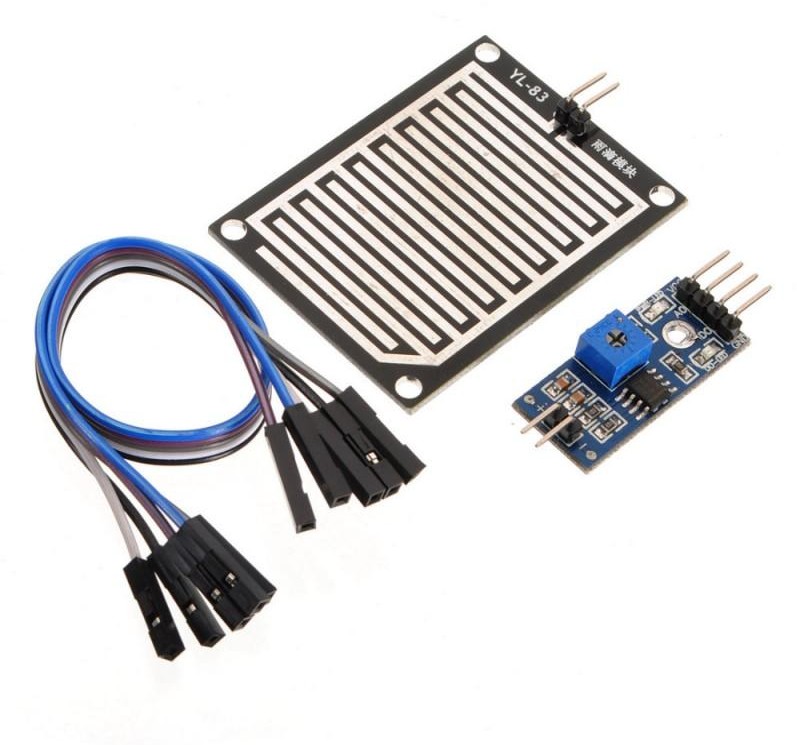
Maintenance free

**▪**

**▪**

* Fast and accurate precipitation detection (ON/OFF)
* Rain intensity measurement with processing unit

Features/Benefits



*Vaisala YL-83 Rain Detector*

Rain and snow are quickly and accurately detected with the YL-83 Rain Detector. The YL-83 operates via droplet detection rather than by signal level threshold.

A special delay circuitry allows about two-minute interval between raindrops before assuming an OFF (no rain) position. This enables the sensor to accurately distinguish between rain cessation and light rain.

The YL-83 also features an analog Rain Signal for estimating rain intensity. Since this signal is proportional to the percentage of moist or wet area on the sensor plate, rain intensity has a direct impact on the amplitude and variation of this analog signal.

The YL-83 sensor is positioned at a 30° angle. This design, together with the internal heating element, ensures that the surface dries quickly, an essential factor in calculating intensity. The same heating element also protects the surface from fog and condensed moisture, and is activated at

low temperatures in order to melt snow, thus allowing snow detection. Sensor performance is not affected by reasonable amounts of dirt and dust due to droplet detection.

It is intended to be used in areas

with only rain or wet/moist snow precipitation.



# Technical Data

Sensor

Capacitive principle, thick layer sensor

RainCap™ with a thin glass shield. Integrated heater element.

Output

Rain ON/OFF

Open collector, active low signal corresponds to rain Maximum voltage

Maximum current

15 V

50 mA

Sensitivity of Rain Detection

Minimum wet area 0.05 cm²

OFF-delay (active) < 5 min

Physical

Sensor plate

Analog output 1...3 V (wet...dry)

Frequency output 1500...6000 Hz,

non-calibrated

Input

Control to switch heater OFF

Sensing area Angle

7.2 cm²

30°

Open circuit input enables the heater. Connection to GND disables the heater.

Housing material Polypropylene

Windshield and support bracket Aluminum Moisture shield Polyurethane

Contact rating min. 15 V, 2 mA

Dimensions

With wind shield Without wind shield

(h × w × l) 110 × 80 × 175 mm

90 × 46 × 157 mm

Ground Wiring

Separate ground wires for signal and heater

Weight 500 g

Cable length 4 m

Electrical

Supply voltage 12 VDC ± 10 %

Supply current

Temperature Range

Operating -15...+55 °C (+5...+131 °F)

Storage -40...+65 °C (-40...+149 °F)

Mounting

Typical less than Maximum Heater OFF

Sensor plate

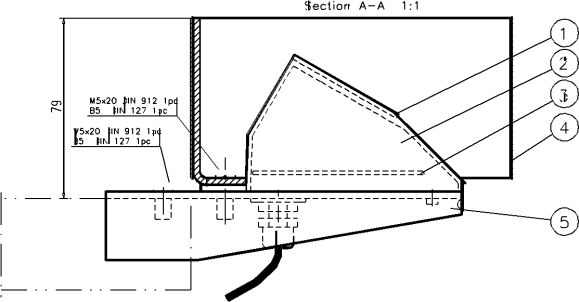
150 mA

260 mA

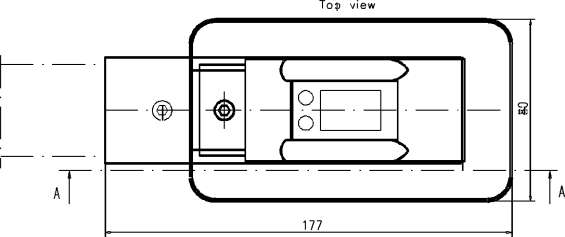
25 mA

By one screw (M5 x 20 mm) to sensor arm

Heating power 0.5 2.3 W



1. *Sensor, RainCapTM*

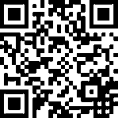


1. *Polyurethane moisture shield*
2. *Component assembly*
3. *Wind shield*
4. *Mounting plate*

Please contact us at [www.vaisala.com/requestinfo](http://www.vaisala.com/requestinfo)



Ref. B010018EN-B ©Vaisala 2015

This material is subject to copyright protection, withall copyrights retained by Vaisala and its individual partners. All rights reserved. Any logos and/or product names are trademarks of Vaisala or its

individual partners. The reproduction, transfer, distribution or

storage of information contained in this brochure

in any form without the prior written consent of Vaisala is strictly

[www.vaisala.com](http://www.vaisala.com/)

prohibited. All specifications — technical included — are subject to change without notice.

Raindrop sensor is basically a board on which nickel is coated in the form of lines. It works on the principal of resistance. When there is no rain drop on board. Resistance is high so we get high voltage according to V=IR. When rain drop present it reduces the resistance because water is conductor of electricity and presence of water connects nickel lines in parallel so reduced resistance and reduced voltage drop across it.

**6.2 Installation of libraries:**

You need to install the **DHTLib** library. It has all the functions needed to get the humidity and temperature readings from the sensor. It’s easy to install.

Open up the **Arduino IDE,** then go to **Sketch** > **Include Library** > **Manage Libraries** > **Search DHTLib**

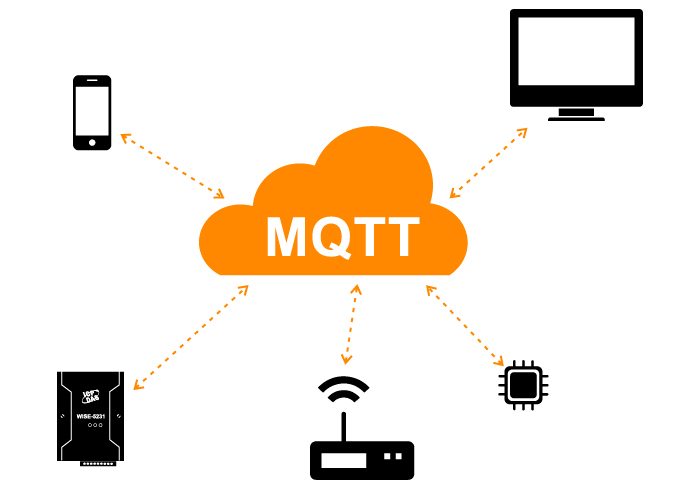
After it’s installed, upload this program to the Node MCU and check output in the serial monitor.

After connecting the devices, the code is written and send to the cloud through MQTT protocol.

There the code is uploaded into Arduino and it shows live data of temperature and humidity of a particular location for every 5 minutes.

**6.3 MQTT PROTOCOL:**

MQTT means Message Queuing Telemetry Transport. It is small size, light weight, low power usage, minimized data packets and ease of implementation .it is used for “machine-to-machine” connection.



**Fig 6.8: MQTT Protocol**

**Features:**

MQTT has unique features you can hardly find in other protocols, like:

It’s a lightweight protocol. So, it’s easy to implement in software and fast in data transmission.

It’s based on a messaging technique.

Minimized data packets. Hence, low network usage.

Low power usage. As a result, it saves the connected device’s battery.

**Working:**

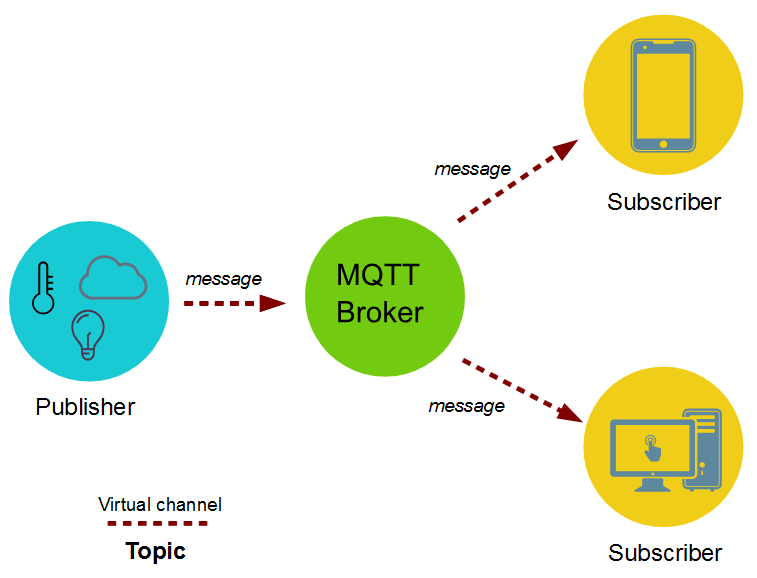
 MQTT is based on clients and a server. The server is the one who is responsible for handling the client’s requests of receiving or sending data between each other. MQTT server is called a broker and the clients are simply the connected devices.

When a device (a client) wants to send data to the broker, we call this operation a “publish”.

When a device (a client) wants to receive data from the broker, we call this operation a “subscribe”.

Publish, is the process a device does to send its message to the broker.

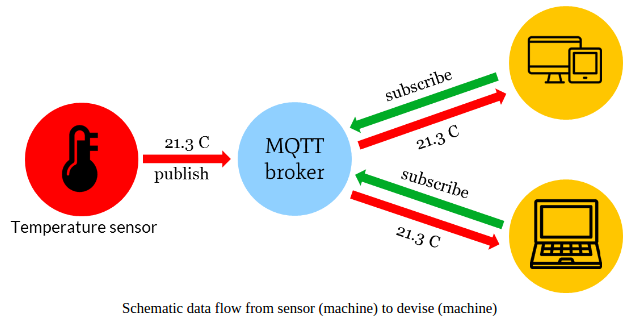
Subscribe, where a device does to retrieve a message from the broker.

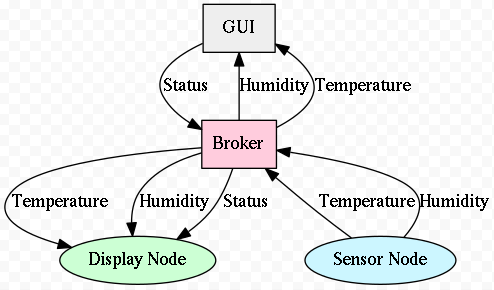


**Fig 6.9: MQTT Broker**

**6.3.1 Sending sensor data to client through MQTT protocol:**

The data such as temperature and humidity values from publisher is send to the **mqtt broker**. The broker role here is to take the message “temperature value” and deliver through a message to the subscriber phone or application.

**Fig 6.10:** **MQTT Schematic Data Flow**

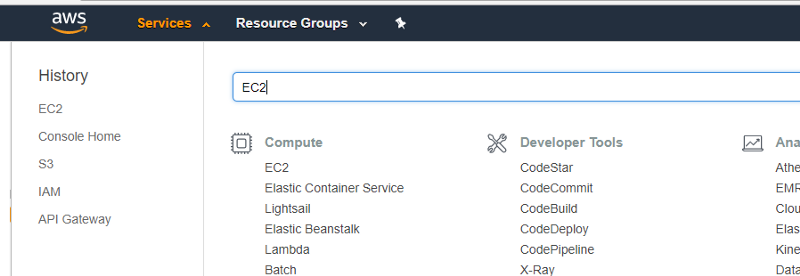


**Fig 6.11:** **MQTT Display INFO**

6.3.2 Configure MQTT

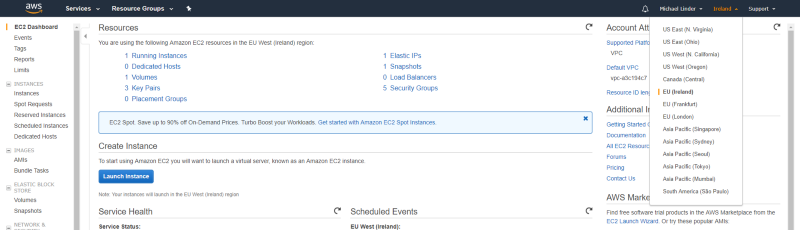
First of all, you need to have an AWS account. If you don’t have one, you can create a new account on AWS ([https://aws.amazon.com](https://aws.amazon.com/)) and get one-year computing, storage and several other services for free using the so called “Free Tier”.

Logon to the AWS Console and then select EC2 in services Section



Select EC2

2. Select the preferred AWS region and then launch a new instance



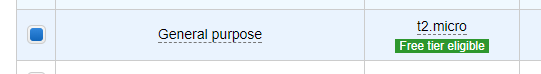
Select AWS region and launch

3. Select Ubuntu Server

https://cdn-images-1.medium.com/max/800/1*njCIH6d4m93BEb5xARdtNg.png

4. Select an instance type

In order to use the free tier contingent, it’s recommended to use t2.micro for testing purposes.



t2.micro selection

5. Configure security group

In this example following ports will be used:

**Step 1: Install Mosquitto**

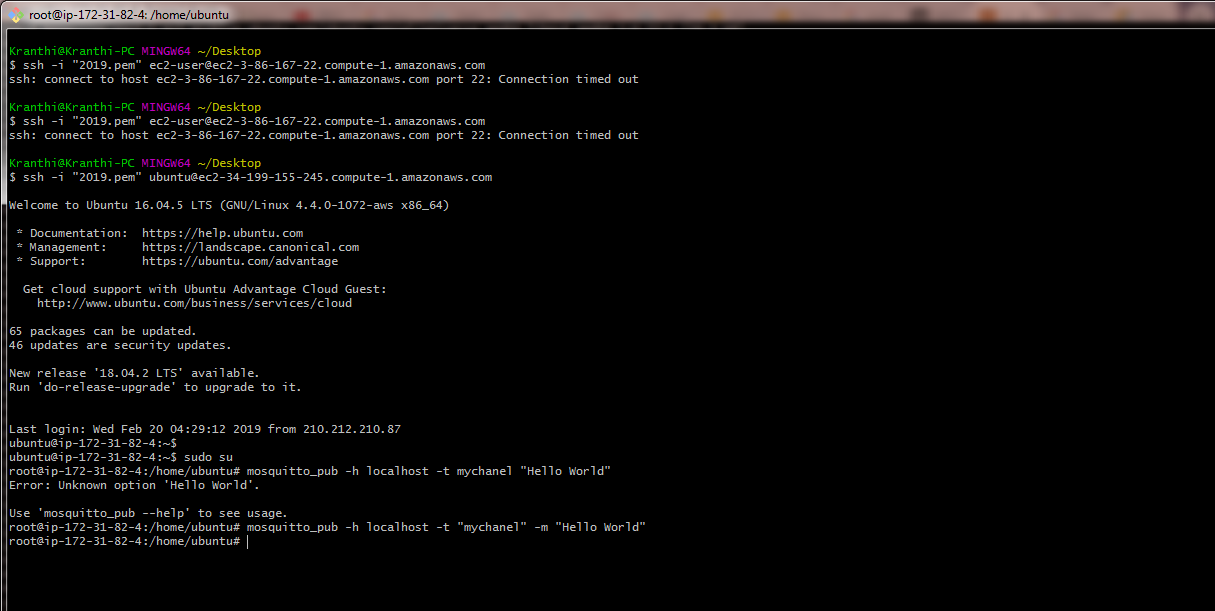
Log into the AWS Ubuntu 16/18 machine.

$ sudo apt-get update

Install

$ sudo apt-get install mosquitto mosquitto-clients

The command above installs both the mosquitto broker and the publish / subscribe clients. The mosquitto broker is now installed and active. You can listen to declare any channel to subscribe and publish to test it



open the duplicate session for this Ubuntu, type the same command for publish

