**Project 26:**

# Iot Weather Station Using Firebase and MIT App Inventor

## Introduction: -

Hello all during this Global Outbreak of COVID-19, the world is swiftly moving towards technology Dependence. Controlling things remotely and Working from Home is the new normal. Learning IOT or say Internet of things is the absolute necessity of time and making a Weather station would be the best way to learn Internet of Things. so, in this blog post, we will learn to make a Weather station that will monitor Temperature, Humidity and Rainfall too, we will integrate all of this on an Android APP.

## COMPONENTS: -

1. DHT 11
2. RAIN WATER SENSOR
3. WEMOS

## 1 Electronics

In this project, we need to create a hardware interface between our Control Point and Internet, which basically means we need 2 Node Points (or say 2 devices), one at Sensor side and another at Control side. The Medium would be Internet.

*MicroController:*

For our build, we will need a Microcontroller that can read our Sensor Data, Send it to Internet and also react to any Inputs received from the Internet. Most Capable candidate for this Task

Will be a *NodeMCU* which has ESP8266 Wi-Fi Enabled board along with USB Interface and Complementary Electronics and the best part is that we can Program it with *Arduino IDE*. Refer to this blog to go through the basics of node MCU.

*Sensors:*

For Temperature and Humidity, we will use a *DH11 Sensor* to sense Temperature and Humidity, we have also covered the basics of DHT sensor in this Post.

For Rain, we will use a generic *Rain Sensor*, which uses Conductive Property of water to change resistance and indicate if it's raining or not, and also how intense is the rain.

It comes as a Set with an operational amplifier to provide more accurate Calibrations which give us Digital or Analog Corresponding Outputs, it has a Potentiometer that can be used to Adjust the Sensitivity of sensor, which we will calibrate by pouring few drops of water on sensor and adjusting the pot until we see output LED glow.

## *Connections:*

For now, connection are very simple,

**DHT11**

Vcc - 3.3v of Node MCU

GND - GND of Node MCU

Data - D5 of Node MCU

**Rain Sensor**

Vcc - 3.3v of Node MCU

GND - GND of Node MCU

Data - D2 of Node MCU

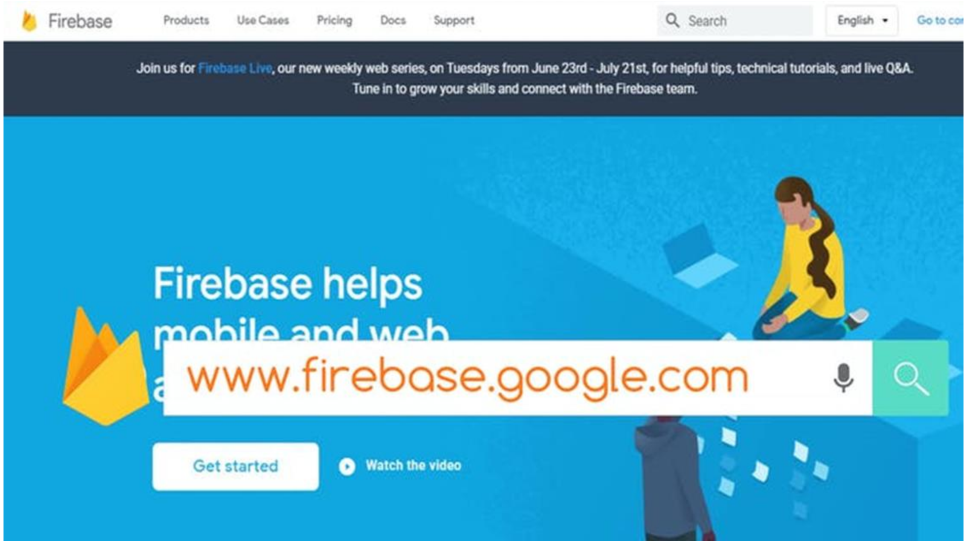
Thanks UTSOURCE.net to offer electronic components for my projects

## Step 2: Software Overview

To Control or Monitor our Sensor in real time, we will need 2 nodes, one node is Microcontroller that connects to Internet via Wireless ACCESS POINT, but we need a second node or point to store data or Process it, for which we have quite a lot of option like Adafruit IO, Blynk, IFTTT, MQTT, Firebase, etc.

The Initial goal of the project was to push the data on Website server ofminov.in,with Firebase and a Firebase WordPress Integration plugin, but couldn’t get it working like expected, so switched to Firebase and MIT- App inventor to make an Android App that would display our data, and it turned out to be Awesome, We will learn that in the further steps.

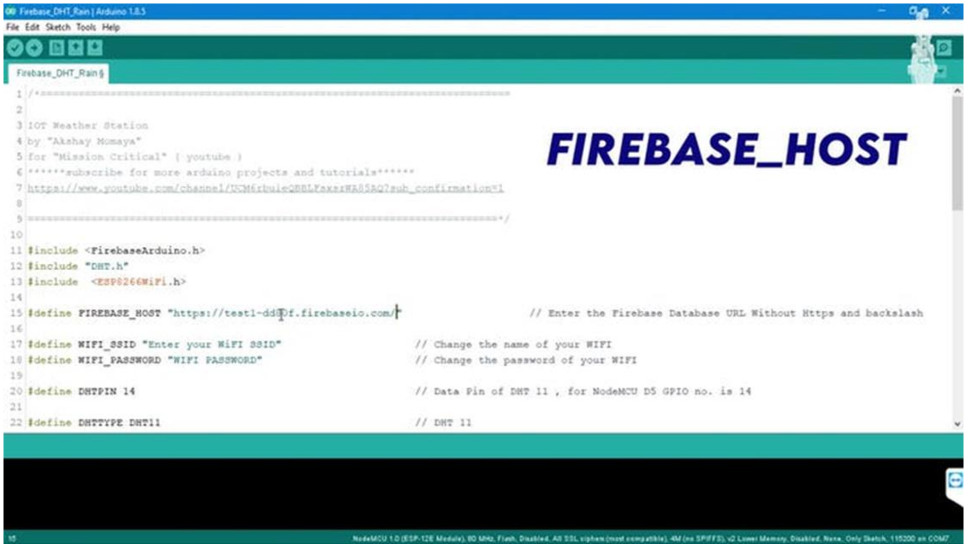
## Step 3: Firebase Database



Join us for Firebase Live, our new weekly web series, on Tuesdays from June 23rd - July 21st, for helpful tips, technical tutorials, and live Q&A. Tune in to grow your skills and connect with the Firebase team.

First go to firebase.google.com, use your Google Accounts and get started, start a new Project by naming and completing the follow-up steps till you see the Dashboard, now go to database option on left side, remember we have Real-time data to deal with, so create a Real-time Database, now it asks if we need authentication or not, since we will make an App and we want it to be publicly available, so choose test mode Database, which will keep Authentication off, it is less secure but works for this application.

## Step 4: Node MCU Code


Firebase_DHT_Rain | Arduino 1.8.5
File Edit Sketch Tools Help
Firebase_DHT_Rain §
FIREBASE_HOST

```
/*=========================================================================

IOT Weather Station
by "Akshay Momaya"
for "Mission Critical" ( youtube )
******subscribe for more arduino projects and tutorials******
https://www.youtube.com/channel/UCM6rbuieQBBLFaxezWA85AQ?sub_confirmation=1

=========================================================================*/

#include <FirebaseArduino.h>
#include "DHT.h"
#include  <ESP8266WiFi.h>

#define FIREBASE_HOST "https://test1-dd0f.firebaseio.com/"                // Enter the Firebase Database URL Without Https and backslash

#define WIFI_SSID "Enter your WiFI SSID"                       // Change the name of your WIFI
#define WIFI_PASSWORD "WIFI PASSWORD"                          // Change the password of your WIFI

#define DHTPIN 14                                              // Data Pin of DHT 11 , for NodeMCU D5 GPIO no. is 14

#define DHTTYPE DHT11                                          // DHT 11
```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, All SSL ciphers (most compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM7

Open the code and Libraries attached in this step and Paste the URL copied in previous step in firebase_host definition line without the https and /(backslash). Make sure you remove it or else the data won’t be directed correctly.

```
#define FIREBASE_HOST "Paste URL here"  // Enter the
Firebase Database URL Without Https and backslash
```

Other than that we need internet connectivity, for which we have to enter our Wi-Fi SSID and Password in definitions.

```
#define WIFI_SSID "Enter your WiFI SSID"   // Change the
name of your WIFI
#define WIFI_PASSWORD "WIFI PASSWORD"      // Change the
password of your WIFI
```

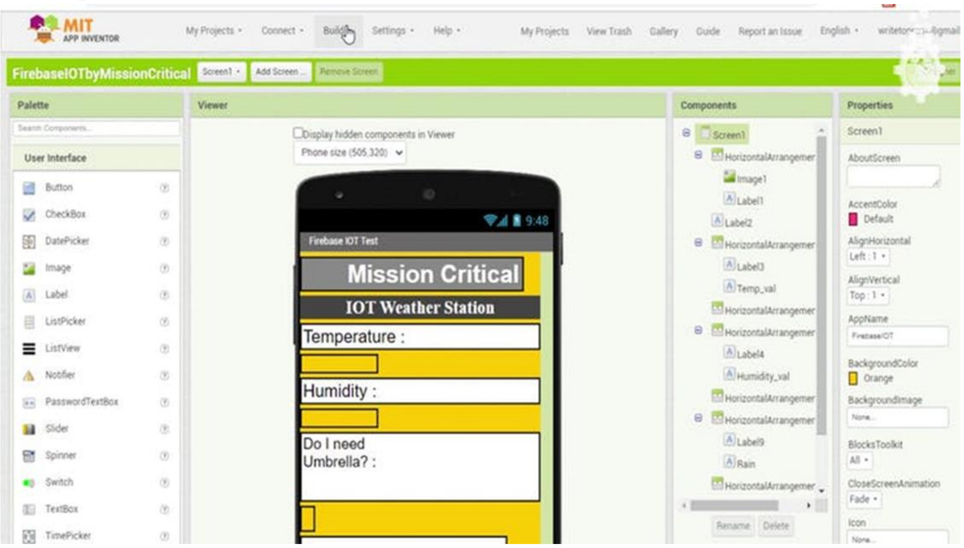
rest of the code is very simple and self-explanatory, where in we are basically pushing all the data from sensor on Real-time Database of firebase.

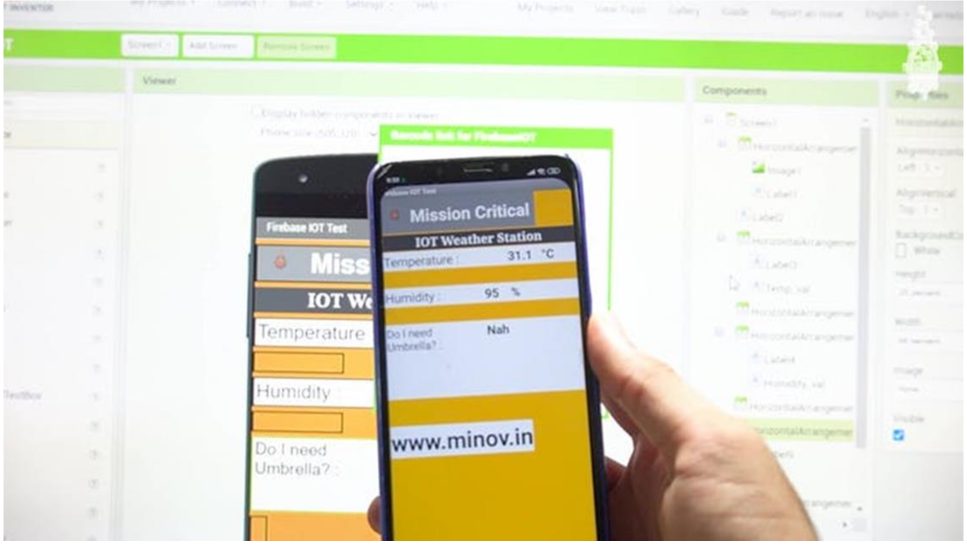
```
void loop()
{
float h = dht.readHumidity();
float t = dht.readTemperature();         // Reading
temperature as Celsius (the default)
int r = digitalRead(4);                  // Reading the Rain
Sensor Data

Firebase.setFloat ("Temperature",t);
Serial.println(t);
Firebase.setFloat ("Humidity",h);
Serial.println(h);
if (r == 1)
{
Firebase.setString("Rain", "Yeah");
Serial.println("its raining");
}
else
{
Firebase.setString("Rain", "Nah");
Serial.println("no rains today");
}
delay(200);
}
```

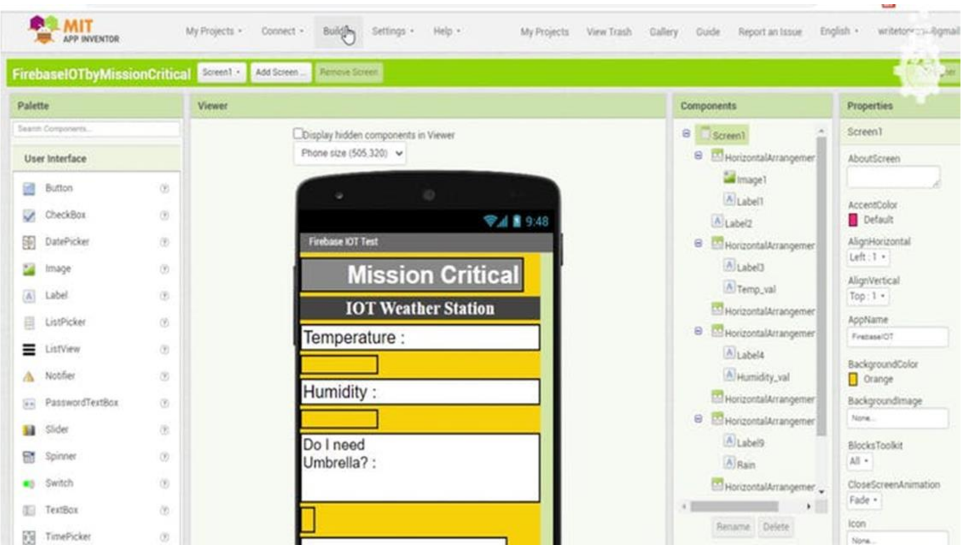
find the code here!

and Libraries here!

## Step 5: Mobile APP

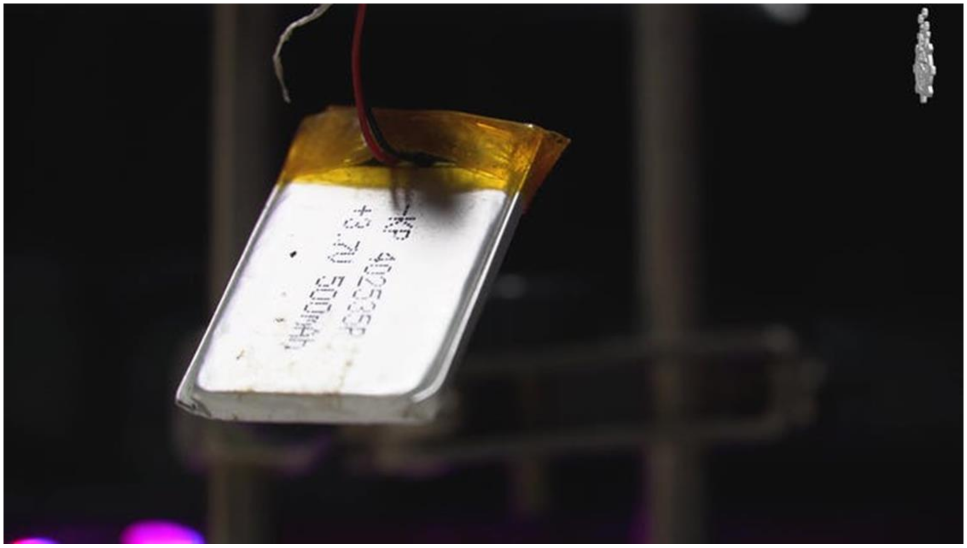






We used MIT APP DEVELOPER, so that you can easily reproduce this project for yourself, here is the.aia file which you can import into MIT App inventor and easily customize for your project and application.

## Step 6: Power Electronics



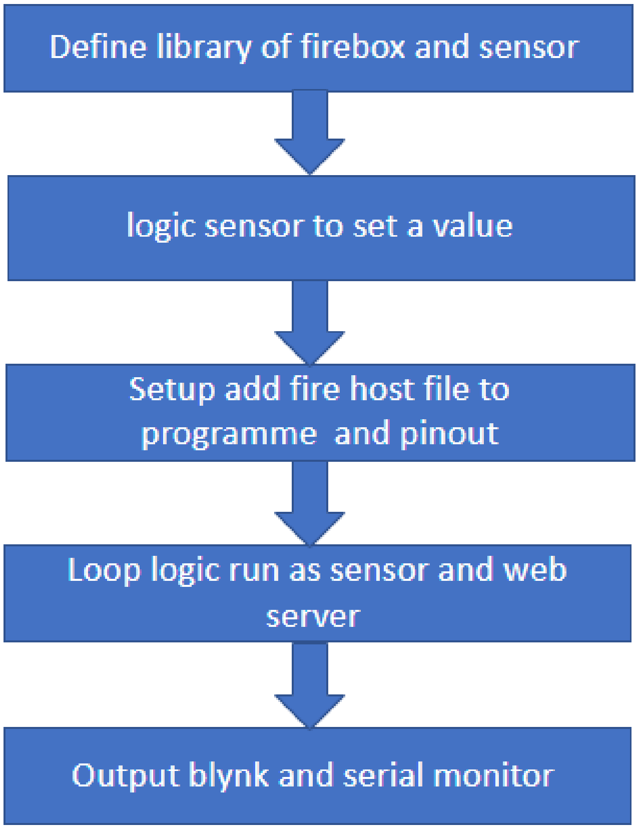
We need to mount this system in Balcony or some place with access to Environment, we can’t power this system with USB wire, so it makes sense to use a Battery, best option would be a Lithium Polymer Battery but my Balcony is home to lots of birds and squirrels, I can’t afford accidental fire here. So safer option would be a Lithium Ion battery, which adds up a little weight but offers higher Capacity and Safety as well. I have this Lithium Ion Cell with 2000 mAh Capacity and we will add a TP4056 Li ion charging module along with it.

After looking at Node MCU’s datasheet, we get to know that NODE MCU uses up to 170 milliamps. At 2000 mAh capacity and current draw of 170 milliamps, we get use time of around 11 hours, which is not much, but luckily we have Deep Sleep Mode in our NodeMCU, which basically puts our System in Deep Sleep to reduce Power Consumption. Simply add Esp.deepsleep (time in microseconds) at the end of our loop.

```
Esp.deepsleep (3e+8);
```

Upload the code and then connect D0 to Reset pin,do not connect D0 and Reset before uploading the code else you will get uploading error. During the Sleep mode, our system uses 8 milliamps. Which turns down our overall current consumption of system and it brings battery life to 10 days ideally. We can remove the LEDs on Node MCU, DHT11 and Rain sensor to further reduce power consumption, but honestly 10 days is fine with me and I am happy with this system.

## FLOW CHART: -


Define library of firebox and sensor
logic sensor to set a value
Setup add fire host file to programme  and pinout
Loop logic run as sensor and web server
Output blynk and serial monitor

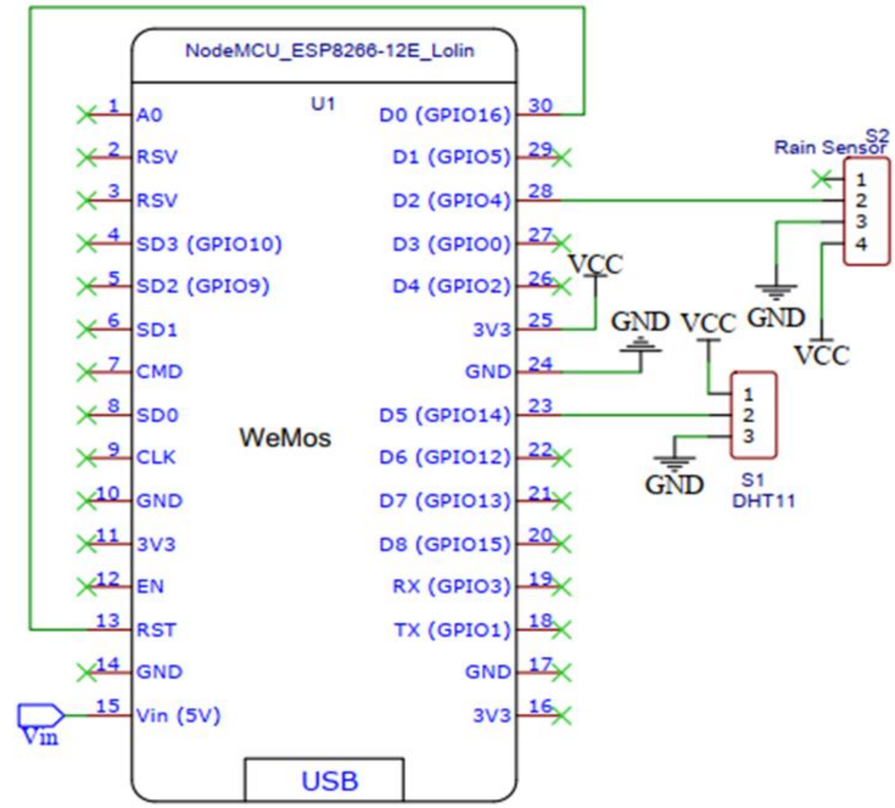
## PROGRAMMING: -

```
void loop() { float h = dht.readHumidity(); float t =
dht.readTemperature();         // Reading temperature
as Celsius (the default) int r = digitalRead(4);
// Reading the Rain Sensor Data   Firebase.setFloat
("Temperature",t); Serial.println(t);
Firebase.setFloat ("Humidity",h); Serial.println(h);
if (r == 1)  {   Firebase.setString("Rain", "Yeah");
Serial.println("its raining");  }   else   {
Firebase.setString("Rain", "Nah");
Serial.println("no rains today");  } delay(200);}<br>
```

## HARDWARE CONNECTION:-

1. **Connect DHT11 to rain sensor and wemos**
2. **Connect pin VCC to VCC**
3. **Connect pin GND to GND**
4. **Connect DHT 11 Data pin D5**
5. **Connect rain sensor data pin D2**

## DIRCUIT DAIGRAM: -


NodeMCU_ESP8266-12E_Lolin
U1
WeMos
1 A0
2 RSV
3 RSV
4 SD3 (GPIO10)
5 SD2 (GPIO9)
6 SD1
7 CMD
8 SD0
9 CLK
10 GND
11 3V3
12 EN
13 RST
14 GND
15 Vin (5V)
30 D0 (GPIO16)
29 D1 (GPIO5)
28 D2 (GPIO4)
27 D3 (GPIO0)
26 D4 (GPIO2)
25 3V3
24 GND
23 D5 (GPIO14)
22 D6 (GPIO12)
21 D7 (GPIO13)
20 D8 (GPIO15)
19 RX (GPIO3)
18 TX (GPIO1)
17 GND
16 3V3
USB
Vin
VCC
GND
VCC
S2
Rain Sensor
1
2
3
4
GND
VCC
1
2
3
GND
S1
DHT11