INTERNET OF THINGS PROJECT

IOT PROJECT ON MINI WEATHER STATION

YEAR: 2020-21

DONE BY: R. DHARSHINII

REFERRED FROM:

IOT PROJECT ON MINI WEATHER STATION

WHAT IS A WEATHER STATION?

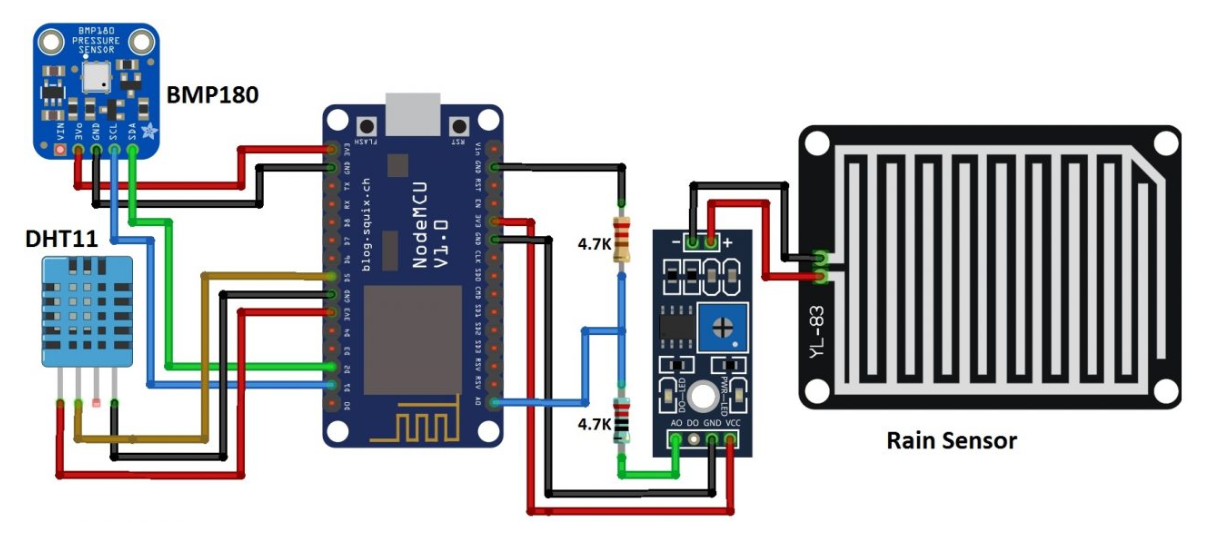
A **weather station** is a device that collects data related to the weather & environment using different sensors. There are two types of weather stations, one which is having own sensors and the second type of weather station is where we pull data from the weather station servers. In this tutorial, we will go for the first one, i.e. we will design our own **weather station**.

Weather station sensors may include a **thermometer** to take temperature readings, a **barometer** to measure the atmospheric pressure, **Hygrometer** to measure humidity, **rain sensor** to measure rainfall, an **anemometer** to measure wind speed, and more. **Weather stations** are also called weather centers, personal weather stations, professional weather stations, home weather stations, weather forecaster, and forecasters.

REQUIREMENTS:

* NODEMCU ESP8266
* BMP180 BAROMETRIC PRESSURE SENSOR
* DHT11 TEMPERATURE SENSOR
* FC-37 RAIN SENSOR MODULE
* RESISTOR (4.7 K)
* JUMPER WIRES
* BREADBOARD

CIRCUIT DIAGRAM:



PROGRAM CODE:

INDEX.H CODE

**const char MAIN\_page[] PROGMEM = R"=====(**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>www.how2electronics.com</title>**

**</head>**

**<style>**

**@import url(https://fonts.googleapis.com/css?family=Montserrat);**

**@import url(https://fonts.googleapis.com/css?family=Advent+Pro:400,200);**

**\*{margin: 0;padding: 0;}**

**body{**

**background:#544947;**

**font-family:Montserrat,Arial,sans-serif;**

**}**

**h2{**

**font-size:14px;**

**}**

**.widget{**

**box-shadow:0 40px 10px 5px rgba(0,0,0,0.4);**

**margin:100px auto;**

**height: 330px;**

**position: relative;**

**width: 500px;**

**}**

**.upper{**

**border-radius:5px 5px 0 0;**

**background:#f5f5f5;**

**height:200px;**

**padding:20px;**

**}**

**.date{**

**font-size:40px;**

**}**

**.year{**

**font-size:30px;**

**color:#c1c1c1;**

**}**

**.place{**

**color:#222;**

**font-size:40px;**

**}**

**.lower{**

**background:#00A8A9;**

**border-radius:0 0 5px 5px;**

**font-family:'Advent Pro';**

**font-weight:200;**

**height:130px;**

**width:100%;**

**}**

**.clock{**

**background:#00A8A9;**

**border-radius:100%;**

**box-shadow:0 0 0 15px #f5f5f5,0 10px 10px 5px rgba(0,0,0,0.3);**

**height:150px;**

**position:absolute;**

**right:25px;**

**top:-35px;**

**width:150px;**

**}**

**.hour{**

**background:#f5f5f5;**

**height:50px;**

**left:50%;**

**position: absolute;**

**top:25px;**

**width:4px;**

**}**

**.min{**

**background:#f5f5f5;**

**height:65px;**

**left:50%;**

**position: absolute;**

**top:10px;**

**transform:rotate(100deg);**

**width:4px;**

**}**

**.min,.hour{**

**border-radius:5px;**

**transform-origin:bottom center;**

**transition:all .5s linear;**

**}**

**.infos{**

**list-style:none;**

**}**

**.info{**

**color:#fff;**

**float:left;**

**height:100%;**

**padding-top:10px;**

**text-align:center;**

**width:25%;**

**}**

**.info span{**

**display: inline-block;**

**font-size:40px;**

**margin-top:20px;**

**}**

**.weather p {**

**font-size:20px;padding:10px 0;**

**}**

**.anim{animation:fade .8s linear;}**

**@keyframes fade{**

**0%{opacity:0;}**

**100%{opacity:1;}**

**}**

**a{**

**text-align: center;**

**text-decoration: none;**

**color: white;**

**font-size: 15px;**

**font-weight: 500;**

**}**

**</style>**

**<body>**

**<div class="widget">**

**<div class="clock">**

**<div class="min" id="min"></div>**

**<div class="hour" id="hour"></div>**

**</div>**

**<div class="upper">**

**<div class="date" id="date">21 March</div>**

**<div class="year">Temperature</div>**

**<div class="place update" id="temperature">23 &deg;C</div>**

**</div>**

**<div style="text-align: center;"><a href="https://www.how2electronics.com" style="align:center">www.how2electronics.com</a></div>**

**<div class="lower">**

**<ul class="infos">**

**<li class="info temp">**

**<h2 class="title">TEMPERATURE</h2>**

**<span class='update' id="temp">21 &deg;C</span>**

**</li>**

**<li class="info weather">**

**<h2 class="title">PRESSURE</h2>**

**<span class="update" id="pressure">0 mb</span>**

**</li>**

**<li class="info wind">**

**<h2 class="title">RAIN</h2>**

**<span class='update' id="rain">0%</span>**

**</li>**

**<li class="info humidity">**

**<h2 class="title">HUMIDITY</h2>**

**<span class='update' id="humidity">23%</span>**

**</li>**

**</ul>**

**</div>**

**</div>**

**<script>**

**setInterval(drawClock, 2000);**

**function drawClock(){**

**var now = new Date();**

**var hour = now.getHours();**

**var minute = now.getMinutes();**

**var second = now.getSeconds();**

**//Date**

**var options = {year: 'numeric', month: 'long', day: 'numeric' };**

**var today  = new Date();**

**document.getElementById("date").innerHTML = today.toLocaleDateString("en-US", options);**

**//hour**

**var hourAngle = (360\*(hour/12))+((360/12)\*(minute/60));**

**var minAngle = 360\*(minute/60);**

**document.getElementById("hour").style.transform = "rotate("+(hourAngle)+"deg)";**

**//minute**

**document.getElementById("min").style.transform = "rotate("+(minAngle)+"deg)";**

**//Get Humidity Temperature and Rain Data**

**var xhttp = new XMLHttpRequest();**

**xhttp.onreadystatechange = function() {**

**if (this.readyState == 4 && this.status == 200) {**

**var txt = this.responseText;**

**var obj = JSON.parse(txt); //Ref: https://www.w3schools.com/js/js\_json\_parse.asp**

**document.getElementById("rain").innerHTML = obj.Rain + "%";**

**document.getElementById("temperature").innerHTML = Math.round(obj.Temperature) + "&deg;C";**

**document.getElementById("temp").innerHTML = Math.round(obj.Temperature) + "&deg;C";**

**document.getElementById("humidity").innerHTML = Math.round(obj.Humidity) + "%";**

**document.getElementById("pressure").innerHTML = Math.round(obj.Pressuremb) + " mb";**

**}**

**};**

**xhttp.open("GET", "readADC", true); //Handle readADC server on ESP8266**

**xhttp.send();**

**}**

**</script>**

**</body>**

**</html>**

**)=====";**

**WEATHER STATION MAIN CODE**

**#include <ESP8266WiFi.h>**

**#include <WiFiClient.h>**

**#include <ESP8266WebServer.h>**

**#include <SFE\_BMP180.h>**

**#include <Wire.h>**

**#include "index.h" //Our HTML webpage contents with javascripts**

**#include "DHTesp.h" //DHT11 Library for ESP**

**#define LED 2 //On board LED**

**#define DHTpin 14 //D5 of NodeMCU is GPIO14**

**SFE\_BMP180 pressure;**

**#define ALTITUDE 1655.0 // Altitude in meters**

**DHTesp dht;**

**//SSID and Password of your WiFi router**

**const char\* ssid = "Alexahome";**

**const char\* password = "hngzhowxiantan";**

**ESP8266WebServer server(80); //Server on port 80**

**void handleRoot() {**

**String s = MAIN\_page; //Read HTML contents**

**server.send(200, "text/html", s); //Send web page**

**}**

**float humidity, temperature;**

**void handleADC() {**

**char status;**

**double T,P,p0,a;**

**double Tdeg, Tfar, phg, pmb;**

**status = pressure.startTemperature();**

**if (status != 0)**

**{**

**// Wait for the measurement to complete:**

**delay(status);**

**status = pressure.getTemperature(T);**

**if (status != 0)**

**{**

**// Print out the measurement:**

**Serial.print("temperature: ");**

**Serial.print(T,2);**

**Tdeg = T;**

**Serial.print(" deg C, ");**

**Tfar = (9.0/5.0)\*T+32.0;**

**Serial.print((9.0/5.0)\*T+32.0,2);**

**Serial.println(" deg F");**

**status = pressure.startPressure(3);**

**if (status != 0)**

**{**

**// Wait for the measurement to complete:**

**delay(status);**

**status = pressure.getPressure(P,T);**

**if (status != 0)**

**{**

**// Print out the measurement:**

**Serial.print("absolute pressure: ");**

**Serial.print(P,2);**

**pmb = P;**

**Serial.print(" mb, ");**

**phg = P\*0.0295333727;**

**Serial.print(P\*0.0295333727,2);**

**Serial.println(" inHg");**

**p0 = pressure.sealevel(P,ALTITUDE); // we're at 1655 meters (Boulder, CO)**

**Serial.print("relative (sea-level) pressure: ");**

**Serial.print(p0,2);**

**Serial.print(" mb, ");**

**Serial.print(p0\*0.0295333727,2);**

**Serial.println(" inHg");**

**a = pressure.altitude(P,p0);**

**Serial.print("computed altitude: ");**

**Serial.print(a,0);**

**Serial.print(" meters, ");**

**Serial.print(a\*3.28084,0);**

**Serial.println(" feet");**

**}**

**else Serial.println("error retrieving pressure measurement\n");**

**}**

**else Serial.println("error starting pressure measurement\n");**

**}**

**else Serial.println("error retrieving temperature measurement\n");**

**}**

**else Serial.println("error starting temperature measurement\n");**

**int rain = analogRead(A0);**

**//Create JSON data**

**String data = "{\"Rain\":\""+String(rain)+"\",\"Pressuremb\":\""+String(pmb)+"\",\"Pressurehg\":\""+String(phg)+"\", \"Temperature\":\""+ String(temperature) +"\", \"Humidity\":\""+ String(humidity) +"\"}";**

**digitalWrite(LED,!digitalRead(LED)); //Toggle LED on data request ajax**

**server.send(200, "text/plane", data); //Send ADC value, temperature and humidity JSON to client ajax request**

**delay(dht.getMinimumSamplingPeriod());**

**humidity = dht.getHumidity();**

**temperature = dht.getTemperature();**

**Serial.print("H:");**

**Serial.println(humidity);**

**Serial.print("T:");**

**Serial.println(temperature); //dht.toFahrenheit(temperature));**

**Serial.print("R:");**

**Serial.println(rain);**

**}**

**void setup()**

**{**

**Serial.begin(115200);**

**Serial.println();**

**// dht11 Sensor**

**dht.setup(DHTpin, DHTesp::DHT11); //for DHT11 Connect DHT sensor to GPIO 17**

**pinMode(LED,OUTPUT);**

**//BMP180 Sensor**

**if (pressure.begin())**

**Serial.println("BMP180 init success");**

**else**

**{**

**Serial.println("BMP180 init fail\n\n");**

**while(1); // Pause forever.**

**}**

**WiFi.begin(ssid, password); //Connect to your WiFi router**

**Serial.println("");**

**// Wait for connection**

**while (WiFi.status() != WL\_CONNECTED) {**

**delay(500);**

**Serial.print(".");**

**}**

**//If connection successful show IP address in serial monitor**

**Serial.println("");**

**Serial.print("Connected to ");**

**Serial.println(ssid);**

**Serial.print("IP address: ");**

**Serial.println(WiFi.localIP()); //IP address assigned to your ESP**

**server.on("/", handleRoot); //Which routine to handle at root location. This is display page**

**server.on("/readADC", handleADC); //This page is called by java Script AJAX**

**server.begin(); //Start server**

**Serial.println("HTTP server started");**

**}**

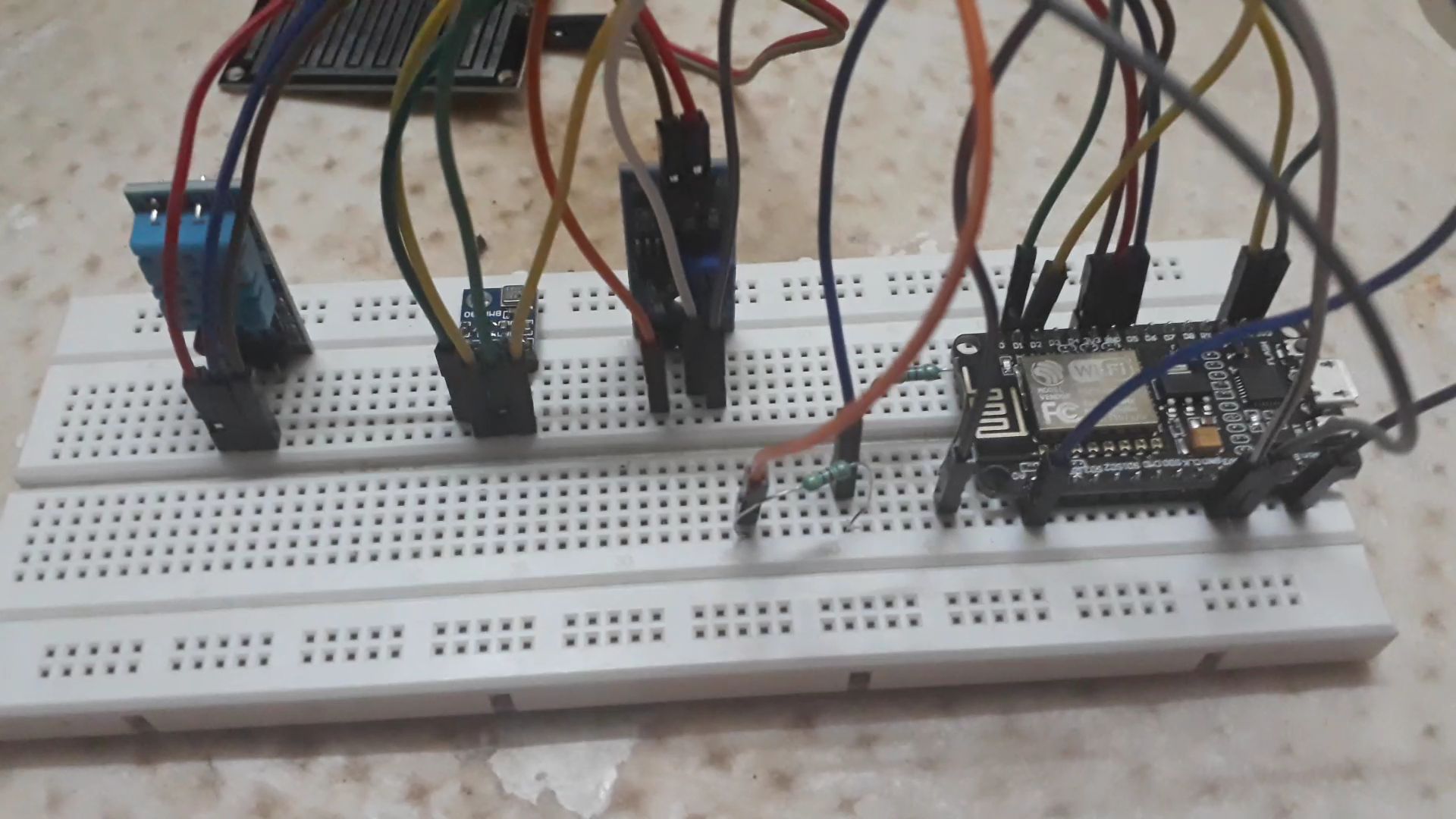
**void loop()**

**{**

**server.handleClient(); //Handle client requests**

**}**

**REAL TIME WORKING:**



**RESULT:**

