August 3, 2019 By EG Projects

In the era of internet of things its now our need to bring control of every thing over internet. Monitoring and controlling peripherals, devices, sensors and nodes remotely saves time and resources. Many of us have gardens in front or back of our houses. Suitable amount of water and normal temperature is crucial for plants to remain green, stand still and grow. Its our utmost desire to water our plants in garden when needed. In this project i am going to solve this problem. I am going to monitor the garden temperature and soil moisture level. If the water content in soil is less than needed by the plants an autonomous switch will start the water pump for watering the plants. User can monitor and control the garden now by his cell phone, desktop and laptop.

Project circuit

I am going to measure the temperature and humidity levels, soil water moisture content and detecting if their is rain in garden. A pump will be driven if water is required by the plants. Sensors which i am using in the project are

- DHT-11 Temperature and humidity sensor.
- Arduino soil and moisture level sensor.
- Nodemcu esp8266-12e WiFi module.

DHT-11 is a one wire digital temperature and humidity sensor. It can measure temperature from 0 degree Celsius to 50 degree Celsius with 2% margin of error. It can measure humidity levels between 20 to 80% with 5% margin of error. It requires 3 to 5 volts for its operation. I have an another and simple tutorial on interfacing DHT-11 with nodemcu wifi module you can learn more from that tutorial. Click the button below to learn more

Nodemcu WiFi with DHT-11 temperature sensor

Soil/water/rain sensor which i am using in the project is popular water moisture sensor among diy circuit makers. It has two legs coated with nickel or copper. Normally the resistance between the legs is high and it decreases when we insert the legs in ground. The moisture in soil made a path between legs and starts conducting power. The strength of conduction depends on the moisture level or water content in soil. More water means more conduction. The output of the soil moisture sensor is analog and digital. Analog output depicts the ratio of water in soil. Where as digital output shows that the moisture water content reached or increased the desired level. Rain/water sensor works on 3.3 to 5 volts. I also have a simple tutorial on interfacing the soil moisture sensor with nodemcu and checking water level over WiFi. Click the below button to learn more.

Nodemcu interfaced with soil moisture sensor

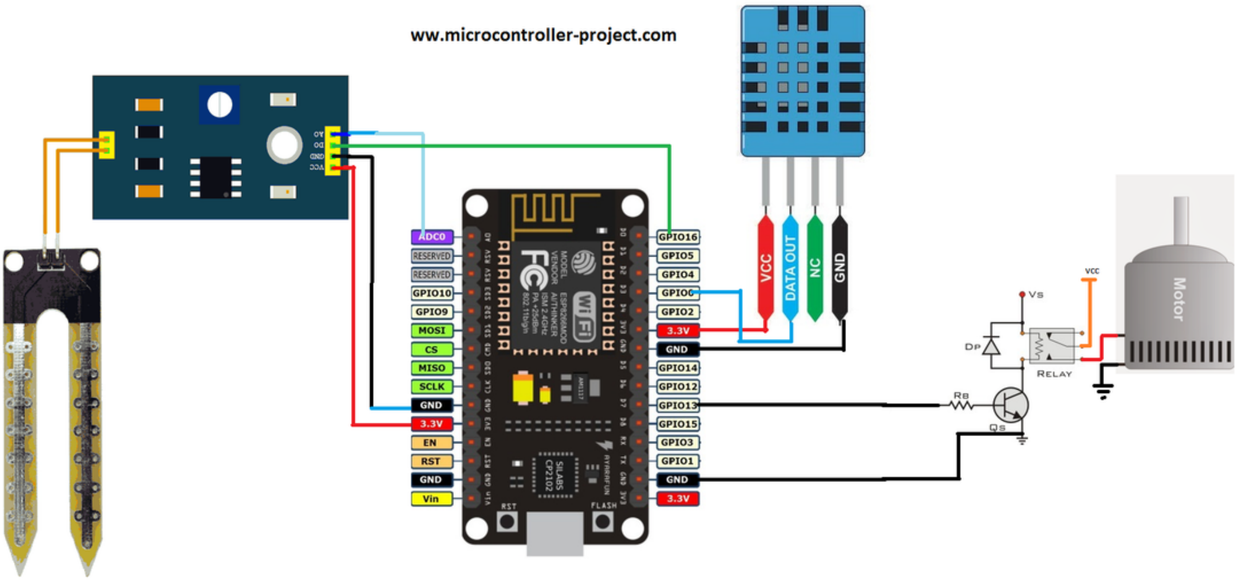
Upon checking the temperature and soil moisture content reading if they are found to be low it means plants needs water. I am going to drive a water pump/motor if soil moisture content is found to be low. For this purpose i interfaced a relay with nodemcu to drive heavy load like motor, pump or water valve. The advantage of relay is it can drive direct as well alternating power loads. You can connect direct current or alternating current motor/pump with relay. Another advantage of relay is it isolates the load from the main circuit and any damage to relay does not impact the main controller.

Nodemcu works on 3.3 volts TTL logic. Its I/O pins also provides 3.3 volts as output. Relays normally needs 5 volts to 12 volts for proper activation of coils. 3.3 volts are not enough to activate the relay coils, alternatively drive the motor/pump. In order to tackle this i connected a NPN transistor with relay and controlled it base with the output of nodemcu digital pin. Now nodemcu 3.3 volt digital pin can drive a 12 volt relay. When digital pin is high motor starts and when it is made low motor will stop. I also have an another tutorial on interfacing relays with nodemcu and controlling heavy loads. Click the below button to learn more.

Interfacing Relay with nodemcu and controlling loads over WiFi

Project Circuit

Soil moisture sensor analog output is connected to nodemcu analog channel 0. Digital output of the soil/water/rain sensor is connected with GPIO-16 or D0 pin of nodemcu. Soil moisture sensor is powered with nodemcu 3.3 volt output. DHT-11 temperature an humidity sensor data output is connected to GPIO-0 or D3 of nodemcu esp-8266-12e WiFi module. DHT-11 is also powered with the 3.3 volt output of nodemcu. Note the pin out of dht-11 their is one void pin. DHT-11 comes in two different pin outs. One has four pins and the other has 3 pins. I used the one with four pins. If you are going to interface the three pin sensor than please first check the right pin out before connecting it with the nodemcu. Both the 3 leg and 4 pin of relay. Connect the com(common) pin of relay with the anode of motor. If pump is working on alternating 110 or 220 volts connect one lead of pump directly to one end of the power rail and NC contact of relay to another line of rail and comm of relay to second lead of pump.



Nodemcu smart garden weather station circuit diagram

Coming to the code portion. First necessary libraries are included for sensors ope-rationalization. Then control pins are defined. Next enter the ssid and password of the WiFi to which you want your nodemcu to be connected to. This is the most important step. Do not forget to do this before uploading the code to nodemcu. Enter the SSID and Password in the block quotes.

const char* ssid = "Your SSID";

const char* password = "Your Wifi Password";

In the setup function nodemcu is requesting your WiFi router for an IP allotment. It then starts its server after IP allotment by the router. Serial communication baud rate is also initialized in the setup loop. Baud rate is set to 9600 bps.

```

/*****************************************************
 * Written by   : Usman Ali Butt                    *
 * Dated        : 4 july 2018                       *
 * Property off : www.microcontroller-projecct.com   *
 ****************************************************/
 #include <ESP8266WiFi.h>
 #include "DHT.h"        // DHT11 temperature and humidity sensor Predefined library

#define DHTTYPE DHT11   // DHT 11
#define dht_dpin 0      //GPIO-0 D3 pin of nodemcu

int Raw       = A0;      //Analog channel A0 as used to measure temperature
int threshold = 16;      //Nodemcu digital pin water sensor read-GPIO16---D0 of NodeMCU
int Solenoid = 13;       // GPIO13---D7 of NodeMCU--Motor connection

const char* ssid = "Your SSID";
const char* password = "Your Wifi Password";


DHT dht(dht_dpin, DHTTYPE);
WiFiServer server(80);

```

```
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);     //Begin WiFi

 while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address on serial monitor
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");    //URL IP to be typed in mobile/desktop browser
  Serial.print(WiFi.localIP());
  Serial.println("/");
}



void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }
  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

 float h =0.0;  //Humidity level
 float t =0.0;  //Temperature in celcius
 float f =0.0;  //Temperature in fahrenheit
 float percentage = 0.0; // Calculating percentage of moisture
 float reading    = 0.0; //Analog channel moisture read

  // Match the request
  int value = LOW;
  if (request.indexOf("/Up=ON") != -1)  {
     h = dht.readHumidity();    //Read humidity level
     t = dht.readTemperature(); //Read temperature in celcius
     f = (h * 1.8) + 32;        //Temperature converted to Fahrenheit
     reading = analogRead(Raw); //Analog pin reading output voltage by water moisture rain sensor
     percentage = (reading/1024) * 100;   //Converting the raw value in percentage

    if (reading<=110){  // If less mositure in soil start the motor otherwise stop
    digitalWrite(Solenoid, HIGH);
    value = HIGH;
    }
    else {
    digitalWrite(Solenoid, LOW);
  if (request.indexOf("/Solenoid=OFF") != -1)  {  //Motor OFF
    digitalWrite(Solenoid, LOW);
    value = LOW;
  }

  // Return the response
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println(""); //  do not forget this one
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");
  client.println("<h1 align=center>Smart Garden - Weather Station</h1><br><br>");
  client.print("Temperature in Celsius =");
  client.println(t);
  client.println("<br>");
  client.print("Temperature in Fahrenheit =");
  client.println(f);
  client.println("<br>");
  client.print("Humidity =");
  client.println(h);
  client.print(" %");
  client.println("<br>");
  client.println();
  client.print("Moisture Level Percentage =");
  client.print(percentage);
  client.print("%");

  if(digitalRead(threshold)==HIGH){ // Read digital output of soil sensor
  client.println("Threshold Reached = Rain detected / Moisture exceeded / Water detected");
  }

  client.println("<br><br>");
  if(value == HIGH)
    client.println("Motor/Pump Operational");
   else
    client.print("Motor/Pump at Halt");

  client.println("<br><br>");
  client.println("<a href=\"/Up=ON\"\"><button>Update = Temperature  Humidity Moisture Values</button></a><br />");
  client.println("<a href=\"/Solenoid=ON\"\"><button>Motor Pump On </button></a>");
  client.println("<a href=\"/Solenoid=OFF\"\"><button>Motor Pump Off </button></a><br />");
  client.println("</html>");
  delay(1);
  Serial.println("Client disonnected");
  Serial.println("");
}
```

weather-station-with-nodemcu.ino hosted with ❤ by **GitHub** **view raw**

After successful initialization of pins and server start up. The main logic comes in loop function. In loop function i am checking for client requests. If any requests arrives temperature, humidity, soil moisture status and other variables are calculated. Before closing the request the client is send back a web page by the server containing the instant values of variables. Client can be desktop, laptop, mobile or a notebook. Nodemcu is working as a server, it is serving a web page. It responds to client when ever the request is made.

One thing to take care of. Both the client and server must be on the same network or in other words they must be connected to same WiFi. If they are not on same network their will be no communication between the client and server. So please be sure both are no the same network.

The headings and buttons above in code are due to the HTML code in the arduino ide. They are visible above because you are viewing it in a browser.

Just make the circuit and upload the code in nodemcu. After uploading the code enter the arduino serial monitor from arduino ide. You will see nodemcu rebooting. First connecting to your WiFi router. Then starting the server. After server start up it will show a web address. This web address is the address of the web page on which user can see the smart

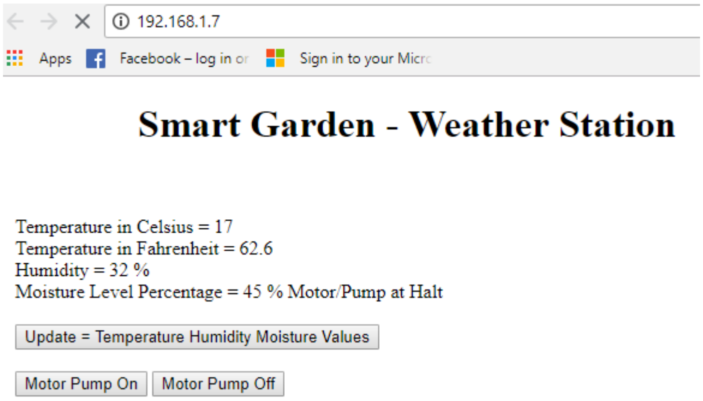
COM1

```
:

Connecting to Your SSID
```

After entering the above address in you browser a web page will be loaded. The format of the page will be some thing like given below. It displays the garden temperature in Celsius and Fahrenheit scale. Humidity in percentage and moisture level also in percentage. It also shows the status of the motor/pump. If the pump is operational page will display "Motor/Pump Operational" status. If water pump is not working page will display "Motor/Pump at Halt" message.

Web page also contains three buttons. Pressing the update button will update the readings displayed on page. Two other buttons turn off and on motor/pump manually. You can turn motor on and off through the web page.



Smart Garden - Weather Station

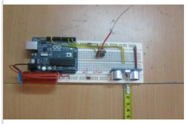
Temperature in Celsius = 17
Temperature in Fahrenheit = 62.6
Humidity = 32 %
Moisture Level Percentage = 45 % Motor/Pump at Halt

nodemcu smart garden monitoring over WiFi esp8266-12e

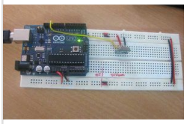
Download the project code. Folder contains the arduino project .ino file. If you have any queries and questions write them below in the comments section.

Code/Files

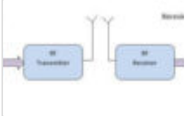
Filed Under: ESP8266, Microcontroller Projects



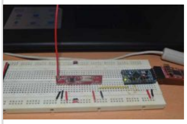
Wireless Distance Measurement using Ultrasonic Sensor (Part 18/23)



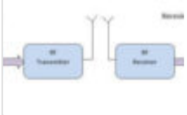
Wireless communication between PC & Arduino for Message Display (Part 20/23)



Adaptive Headlights for Cars (Part 21/23)



RF Data Communication between two Personal Computers (Part 22/23)



Wireless Humidity and Temperature Monitoring System (Part 23/23)

RECENT ARTICLES

- Toshiba releases high-peak output current photocouplers
- Allegro brings high-end LED lighting technology to mainstream vehicles
- Micron and MediaTek first to validate LPDDR5X 5G chipset for smartphone
- Digi-Key and Truphone to host “eSIM Technology for IoT” webinar
- Microchip expands its gallium-nitride RF power portfolio

MOST POPULAR

5G 555 timer circuit 8051 Arduino atmega16 automotive avr connector dc motor display Electronic Part Electronic Parts Fujitsu gsm ic infineontechnologies integratedcircuit Intel invention IoT ir lcd ldr led maximintegratedproducts microchip Microchip Technology microchiptechnology microcontroller motor powermanagement Raspberry Pi remote renesaselectronics renesaselectronicscorporation Research robot samsung sensor software STMicroelectronics switch Technology vishayintertechnology wireless

ANALOG IC TIPS
CONNECTOR TIPS
DESIGNFAST
EDABOARD FORUMS
EE WORLD ONLINE
ELECTRO-TECH-ONLINE FORUMS
MICROCONTROLLER TIPS
POWER ELECTRONIC TIPS
SENSOR TIPS
TEST AND MEASUREMENT TIPS
5G TECHNOLOGY WORLD
ABOUT US
CONTACT US
ADVERTISE