

Connect the ESP8266 WiFi Chip to your Raspberry Pi

November 2nd 2015 / by Marco Schwartz

In this project, we are going to make a typical home automation project: a WiFi weather measurement station. We will connect a DHT11 sensor to an ESP8266 board, and access the data via WiFi. However, here, we are actually going to grab that data from a Raspberry Pi, and make the Pi display the data on a simple graphical interface.

To do so, we will run a simple web server on the ESP8266 chip. The Raspberry Pi will then access this data via WiFi, and display it graphically. The nice thing is you can apply this in several other projects, to make the Raspberry Pi the 'hub' of your home, with several ESP8266-based devices connected to it. Let's start!

Hardware & Software Requirements

For this project, you will of course need an ESP8266 chip. Here, I used the Adafruit ESP8266 board.

You will also need a fully configured Raspberry Pi board, with the Raspbian operating system installed on it.

You will also need a temperature sensor. I used a DHT11 sensor, which is cheap, very easy to use & that will allow us to measure the ambient temperature & humidity.

This is a list of all the extra components that will be used in this chapter:

- [Raspberry Pi board](#)
- [Adafruit ESP8266 breakout board](#)
- [USB 5V FTDI converter](#)
- [DHT11 sensor](#)
- [Breadboard](#)
- [Jumper wires](#)

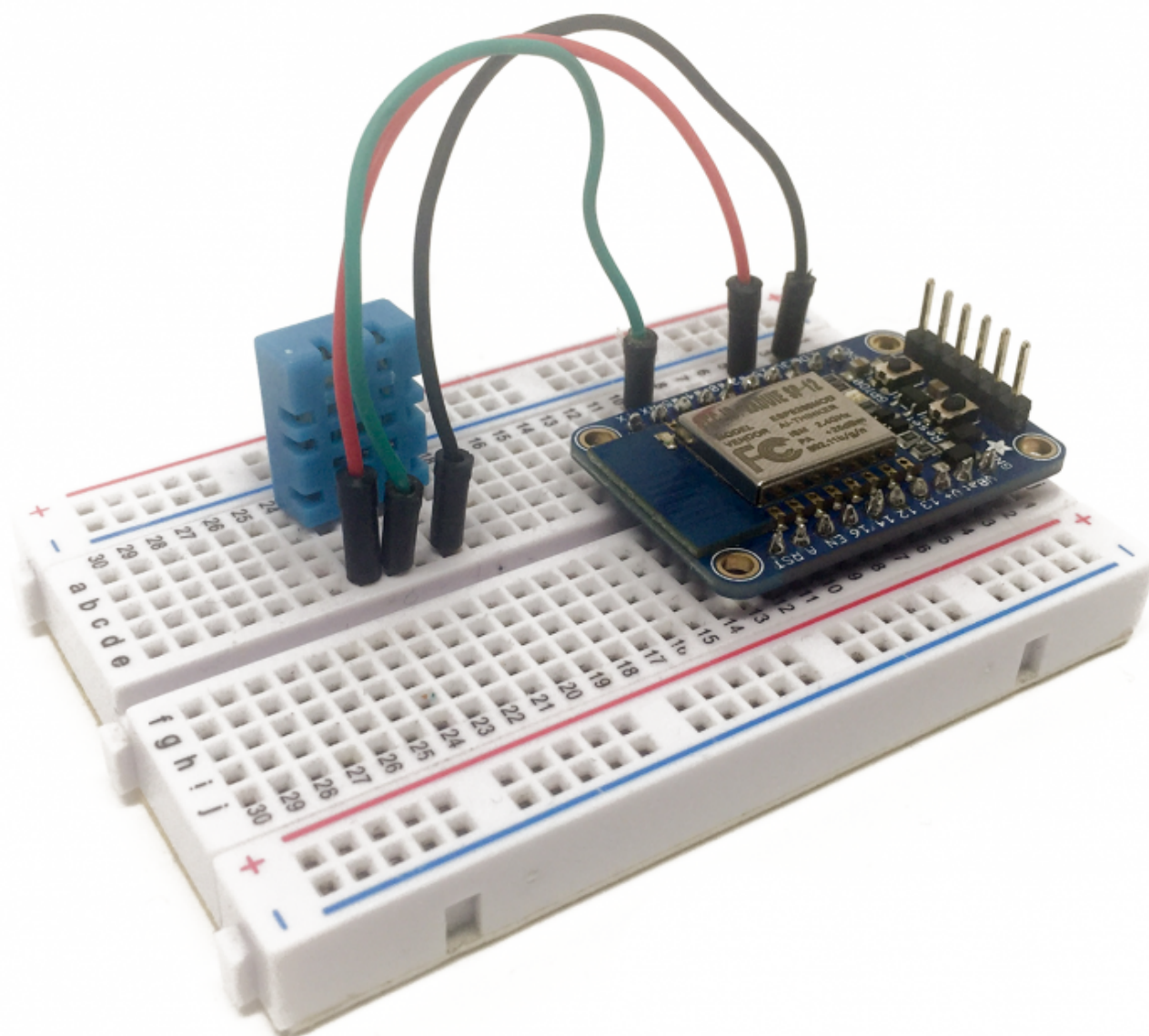
You will need to have the latest version of the Arduino IDE installed. You will also need the DHT library. You can install it from the Arduino library manager, that you can access from **Sketch>Include Libraries>Manage Libraries** inside the Arduino IDE.

On your Raspberry Pi, you will also need to have Node.js installed. To do so, follow the instructions from:

<https://learn.adafruit.com/node-embedded-development/installing-node-dot-js>

Hardware Configuration

Once this is done, simply insert the DHT11 sensor on the breadboard. Then, connect the left pin of the sensor to the 3.3V pin of the ESP8266, the right pin to GND (blue power rail), and the pin next to VCC to the GPIO5 pin on your ESP8266 board. This is the final result, not showing the USB-to-Serial FTDI converter:



Make sure that you connected everything according to the instructions above, or you won't be able to continue.

Testing the Sensor

We are now going to test the sensor. Again, remember that we are using the Arduino IDE, so we can code just like we would do using an Arduino board. Here, we will simply print the value of the temperature inside the Serial monitor of the Arduino IDE.

This is the complete code for this part:

Someone from Ukraine

Recently downloaded the free eBook

10 days ago

```
// Libraries
#include "DHT.h"

// Pin
#define DHTPIN 5

// Use DHT11 sensor
#define DHTTYPE DHT11

// Initialize DHT sensor
DHT dht(DHTPIN, DHTTYPE, 15);

void setup() {

// Start Serial
Serial.begin(115200);

// Init DHT
dht.begin();
}

void loop() {

// Reading temperature and humidity
float h = dht.readHumidity();
// Read temperature as Celsius
float t = dht.readTemperature();

// Display data
Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(t);
Serial.println(" *C ");

// Wait a few seconds between measurements.
delay(2000);

}
```

Let's see the details of the code. You can see that all the measurement part is contained inside the loop() function, which makes the code inside it repeat every 2 seconds.

Then, we read data from the DHT11 sensor, print the value of the temperature & humidity on the Serial port.

Note that the complete code can also be found inside the GitHub repository of the book:

<https://github.com/openhomeautomation/connect-esp8266-rpi>

You can now paste this code in the Arduino IDE. Then, go in **Tools>Boards**, and select the Adafruit ESP8266 board from the list.

After that, we need to put the board in bootloader mode, so we can program it. To do so, refer to the first chapter of this part of the book.

Now, upload the code to the board, and open the Serial monitor when this is done. Also set the Serial monitor speed to 115200.

You should immediately see the temperature & humidity readings inside the Serial monitor. My sensor was reading around 24 degrees Celsius when I tested it, which is a realistic value.

Accessing the Sensor via WiFi

At this point, we are sure that the sensor is working and that data can be read by the ESP8266 chip. Now, we are going to build the sketch that will connect to your WiFi network, and then make the measurements accessible by the Raspberry Pi.

As this sketch is quite long, I will only detail the most important parts here. You can of course find the complete code for this project inside the GitHub repository of the project.

It starts by including the required libraries for this project:

```
#include "ESP8266WiFi.h"
#include <aREST.h>
#include "DHT.h"
```

Then, you need to set up your own WiFi network name & password in the code:

```
const char* ssid = "your_wifi_network_name";
const char* password = "your_wifi_network_password";
```

We also declare the aREST API, that will help us control the board remotely from the Raspberry Pi:

```
aREST rest = aREST();
```

Now, we also declare two variables that will contain the measurements made by the sensor:

```
int temperature;
int humidity;
```

After that, we create a web server on port 80:

```
WiFiServer server(80);
```

Then, inside the setup() function of the sketch, we connect the ESP8266 to the WiFi network:

```
WiFi.begin(ssid, password);

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

Then, we expose the two measurement variables to the aREST API, so they can be access from the outside:

After that, we set the name & ID of the board:

```
rest.set_id("1");  
rest.set_name("sensor_module");
```

Then, we start the server, and print the IP address on the Serial port:

```
// Start the server  
server.begin();  
Serial.println("Server started");  
  
// Print the IP address  
Serial.println(WiFi.localIP());
```

Inside the `loop()` function of the sketch, we check if a client is connected to the ESP8266, and handle the request:

```
WiFiClient client = server.available();  
if (!client) {  
  return;  
}  
while(!client.available()){  
  delay(1);  
}  
rest.handle(client);
```

Then, we read data from the sensor:

```
temperature = dht.readTemperature();  
humidity = dht.readHumidity();
```

It's now time to upload the sketch the board. You can grab the complete code from:

<https://github.com/openhomeautomation/connect-esp8266-rpi>

Follow the instructions we saw in the previous section to upload the code to the board. Then, open the Serial monitor to get the IP address of the board, you will need it soon.

Connecting the Board to Your Raspberry Pi

We are now going to see how to connect the board to your Raspberry Pi, so it can display the data in a nice graphical interface.

The only thing that you need to change to run this interface is the IP address of your ESP8266 board inside the `app.js` file:

```
rest.addDevice('http', '192.168.0.100');
```

You can get all the code from:

<https://github.com/openhomeautomation/connect-esp8266-rpi>

Place all the files into a folder on your Pi, navigate to this folder with a terminal, and type:

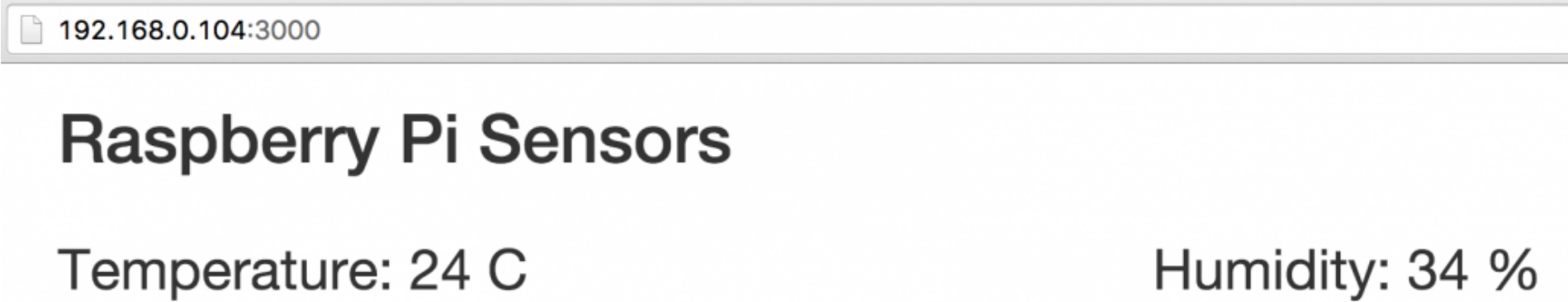
```
sudo npm install express jade arest
```

Be patient, this step can take a while. After that, type:

This will start the application on your Raspberry Pi. Then, navigate to the URL of your Raspberry Pi on port 3000, for example:

```
http://192.168.0.104:3000
```

You should immediately see the interface showing the temperature & humidity readings made by the ESP8266 board:



How to Go Further

Let’s summarise what we achieved in this project. We built a WiFi measurement station based on the ESP8266 WiFi chip. We used a sensor to measure the local temperature & humidity. Then, we displayed all the measurements on a web page that was served by the Raspberry Pi.

There are many things you can do to improve this project. You can for example deploy more of those sensors boards in your home, and display all the measurements on a single page on your Pi.

Do you want to learn more?

If yes, join over 2,000 people who receive exclusive information about open-source home automation & the IoT. You will also receive my free eBook "Build a Simple Thermostat with the Raspberry Pi". Simply click on the button below!

Get Started!

Leave a Comment

Your name ...

Your email ...

Join the discussion...

Post Comment

Amit Bhorania

2 years ago

Nice work and Good explanation..... :-)

Reply

Marco Schwartz

↩ Amit Bhorania 2 years ago

Thank you!

Craig

2 years ago

Do you have a diagram showing the connection between with the USB-to-Serial FTDI converter? Thank you in advance Marco.

Reply

santa2015 2 years ago

I get strange readings from the DHT 11 (connected to the ESP8266), like "Temperature 2147483647 C Humidity: 2147483647 %", which I guess are the maximum values for a Java application. Any idea what might be the problem? Might a capacitor between VCC and GND, and a pull-up resistor between VCC and data might be the solution? When the mote (esp8266+dht) was placed in a room with average temperature > 26 degrees everything was working ok, but when the mote was moved to a colder room (~15 C degrees), I get these strange values.

Reply

Marco Schwartz ↩ santa2015 2 years ago

Strange readings indeed. On the ESP8266 I never needed to use a resistor with the DHT11, but that's something you might try.

Lars 2 years ago

Sorry, I am out of ideas to fix this problem: /home/lars/esp8266_temp/connect-esp8266-rpi-master/interface/app_160104.js:13:rest.addDevice('http','192.168.1.72'); ^TypeError: undefined is not a function at Object.<anonymous> (/home/lars/esp8266_temp/connect-esp8266-rpi-master/interface/app_160104.js:13:6) at Module._compile (module.js:460:26) at Object.Module._extensions..js (module.js:478:10) at Module.load (module.js:355:32) at Function.Module._load (module.js:310:12) at Function.Module.runMain (module.js:501:10) at startup (node.js:129:16) at node.js:814:3

Reply

Marco Schwartz ↩ Lars 2 years ago

Hi Lars, did you correctly installed the Node.js aREST module on the Pi ?

Julian Balles 2 years ago

Hey I'm getting this problem: user "root" does not have permission to access the dev dir "/root/.node-gyp/0.12.6" then the pi "attempts to reinstall using temporary dev dir "/folder where i have the project" Many Errors are following after some time. "node-pre-gyp install --fallback-to-build" fails the system "problem with serial port package" i have node.js installed. any help ? THX !

Reply

Julian Balles ↩ Julian Balles 2 years ago

fixxed it. had to run this befor "node app.js": sudo npm install --unsafe-perm serialport now everything works fine

Svenhook 2 years ago

Hey Marco, this is a great write-up. Thank you. I was wondering if there is a way to write to other ESP8266 boards from the Pi server. For example, I have another ESP8266 set up on the same network and I want to turn on the LEDs on the board. How would I go about doing that? Thanks! Steve

Reply

Marco Schwartz ↩ Svenhook 2 years ago

Hello Steve! I actually wrote a book on that :) <http://www.amazon.com/Home-...> Let me know by email at contact@openhomeautomation.net if you need the book in PDF or ePub

Karthik Bandi 2 years ago

Hai Marco, the project is very nice and really like to do, but can you explain how to connect the wifi module to laptop using ftdi converter, and can you email me the book you have written in pdf format to karthikbandi1991@gmail.com

Reply

Jango Flaisher 2 years ago

hi theri just try your great tutorial and my dht alwasys shows "nan" as a result (temp and HUM) every 2 sec it shows an empty result, can you guess why? of course i double checked

Marco Schwartz ↩ Jango Flaisher 2 years ago

Hey! I'd check the resistor between pin 1 & 2, if 4.7k Ohm doesn't work try 10k

Karthik Bandi 2 years ago

Hail cannot find Adafruit ESP8266 board under Tools>Boards, I even searched in Board manager. Can any one help the project is very nice and really like to do, but can you explain how to connect the wifi module to laptop using usb to ttl converter, and can you email me the book you have written in pdf format to karthikbandi1991@gmail.com

Reply

Marco Schwartz ↩ Karthik Bandi 2 years ago

Hi Karthik, you need to install the board first using the Board manager. Follow the instructions from: <https://github.com/esp8266/...>

Greg Walters 2 years ago

Marco, Great article! Question. My network uses DHCP to assign IP addresses, which means it is possible that an ESP8266 could get a different IP address if it was off line for a period of time. Therefore, I would like to actually enable the Raspberry Pi to "discover" the ESP8266 devices on my network, allowing me to add new sensors without hard coding the IP addresses of every sensor. Do you have any suggestions? Do you cover this in your book? Thanks!

Reply

Marco Schwartz ↩ Greg Walters 2 years ago

Hi Greg, this is indeed something I've been thinking about, and a very good question. What you could do is MDNS to give names to each of your ESP8266 device, that won't change if the IP address of the boards change. I am also exploring the actual 'discovery' of the devices, using nmap & Node.js: <https://github.com/jas-/nod...> When I have some results I'll for sure include that into the aREST framework :)

divya chintarpalli 2 years ago

hey, I want to build a wireless sensor network using arduino , raspberry pi and wifi module . DHT11 is connected to arduino and i want to send my sensor data from arduino to raspberry pi using wifi module, any idea how can i do that? Thanks!

Reply

Marco Schwartz ↩ divya chintarpalli 2 years ago

Hey! I think you can easily adapt this tutorial with an Arduino board, for example a MKR1000 :)

Andrew 2 years ago

Hey Marco, great read and you explained everything very good. Just to mention that in order for the project to work you have to unsafe install the serial port files as well as use pug instead of jade version. Is there any chance i get a hold of the pdf you mentioned in a previous comment ? Thank again for all your effort.

Reply

Marco Schwartz ↩ Andrew 2 years ago

Thanks! About the PDF, contact me directly by email: contact@openhomeautomation.net

Yogesh ↩ Andrew a year ago

Hi Andrew, I too get an error while running the "sudo npm install express jade arest" command (I have changed jade to pug): Failed at the serialport@4.0.7 install script 'node-pre-gyp install --fallback-to-build'. Could you please elaborate on the "you have to unsafe install the serial port files" that you mention in above comment?

Luis German Ruiz Garcia 2 years ago

Art Holder 2 years ago

Hi Marcol being a novice I went through your course and got the DTH 11 working however the next day when I wanted to build the interface I could not get any readings from the DTH 11 no temperature or humidity readings only returned nan any ideas what went wrong appreciate any help to get this working again I am stumped.

Reply

Art Holder 2 years ago

Found the problem jumper wire faulty thanks

Reply

Ranganath Dn 2 years ago

Hey,We have a ready product of EPS8266 based WiFi relay which works with Raspberry Pi to control any appliance wireless. Check the following link for more details
:https://www.crazypi.com/ras...

Reply

Dean Greenhough a year ago

There appears to be an issue with Jade and changing to pug, running app.js gives me an error of cannot find module mime-db. Can anyone help with this please, am i not ptting everything in the correct place?

Reply

Dean Greenhough a year ago

Hi MarcoCan you kindly update this page as I am unable to get this to work due to a Jade/pug issue. I did purchase your builder pack for both raspl and esp8266, but still with all the extra useful information I am unable to get this to work. Your kind and helpful input would be gratefully aprecited

Reply

Marco Schwartz ↩ Dean Greenhough a year ago

Hi Dean, can you send a message to me directly at
marcolivier.schwartz@gmail.com?

ROHIT CHAKRABORTY a year ago

how to do the same thing without making esp connect to wifi / internet?is there some other way to send data to pi without connecting to internet? as I have a lot of esp to connect to one pi , I cant make all esp to connect to a router.my pi can get internet access.

Reply

clinton panjaitan a year ago

hei sir.. why i got this problem? https://uploads.disquscdn.c...

Reply

Søren Løkkegaard 9 months ago

Would i be possible to switch it around, so the ESP is the client instead of web server.so it would be possible to put it to sleep, and send data at a fixed interval.This way it could be battery operated.

Reply