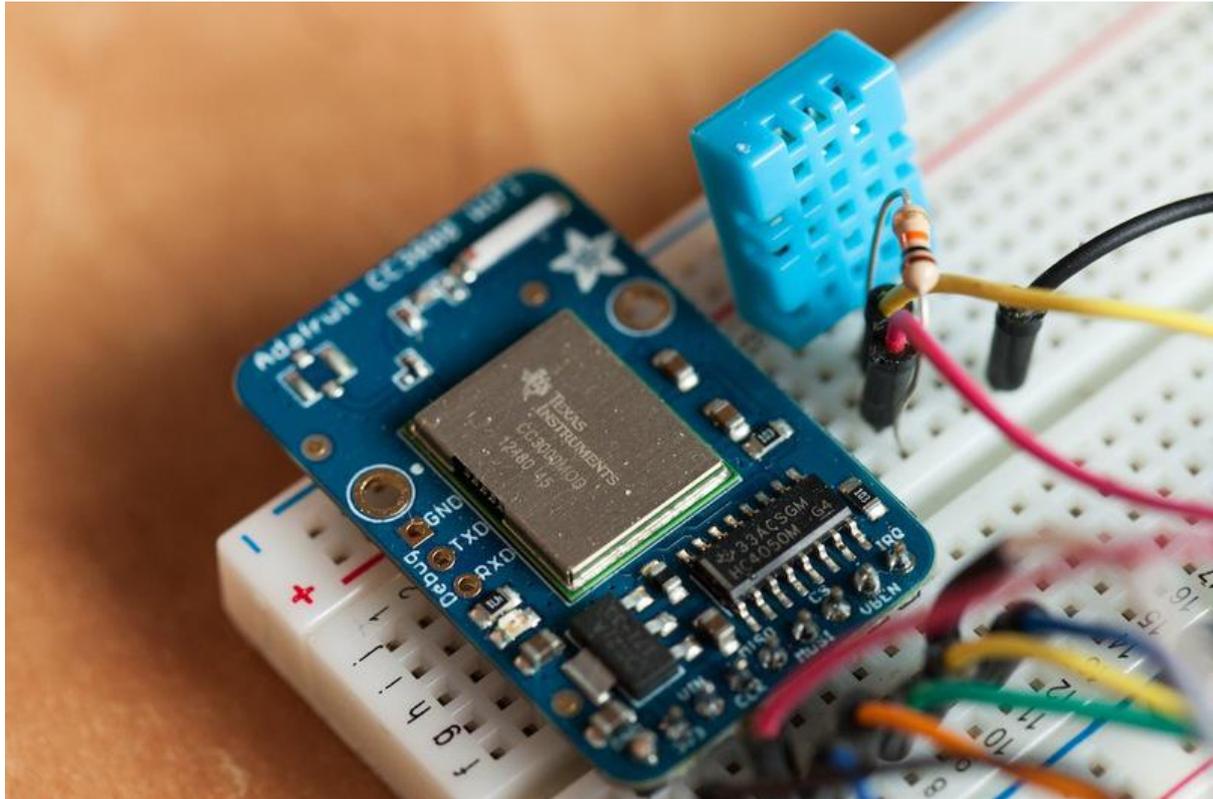




WiFi Weather Station

Created by Marc-Olivier Schwartz



<https://learn.adafruit.com/wifi-weather-station-arduino-cc3000>

Last updated on 2023-08-29 02:23:33 PM EDT

Table of Contents

Introduction	3
Connections	3
<ul style="list-style-type: none">• CC3000 Breakout Board• DHT11 sensor	
Testing the hardware	7
<ul style="list-style-type: none">• CC3000 WiFi chip• DHT11 sensor	
Arduino sketch	10
Building the web interface	11
Using the weather station	12

Introduction

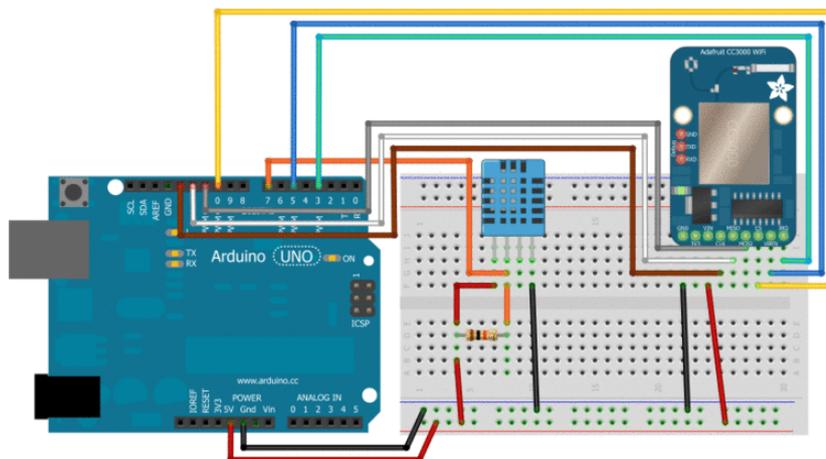
WiFi Weather Station

Temperature: 22 C Humidity: 39 % **Online**

As open-source hardware users and makers, we love playing with new chips, boards and tools. And there is one chip which is quite popular these days: the CC3000 WiFi chip from TI. This all-in-one module has low-power WiFi and a microcontroller-friendly interface.

It has been featured in many articles around the web, and the Adafruit's CC3000 breakout board really makes it easy to integrate this module in any Arduino project. In this tutorial, you will learn how to use this chip for home automation purposes. And in particular, we are going to see how to use this chip to build a simple WiFi-connected weather station. The station will measure two variables: temperature and humidity. But we won't display the information on an LCD screen, like a weather station you could buy in a store. No, instead, we will transmit the data wirelessly via WiFi to your computer and display it there on a nice interface. Excited? Let's get started!

This tutorial is for people who want to use the CC3000 to send data to a basic server script. It's best followed by those who are comfortable setting up a webserver on their computer and running a Node.js server script

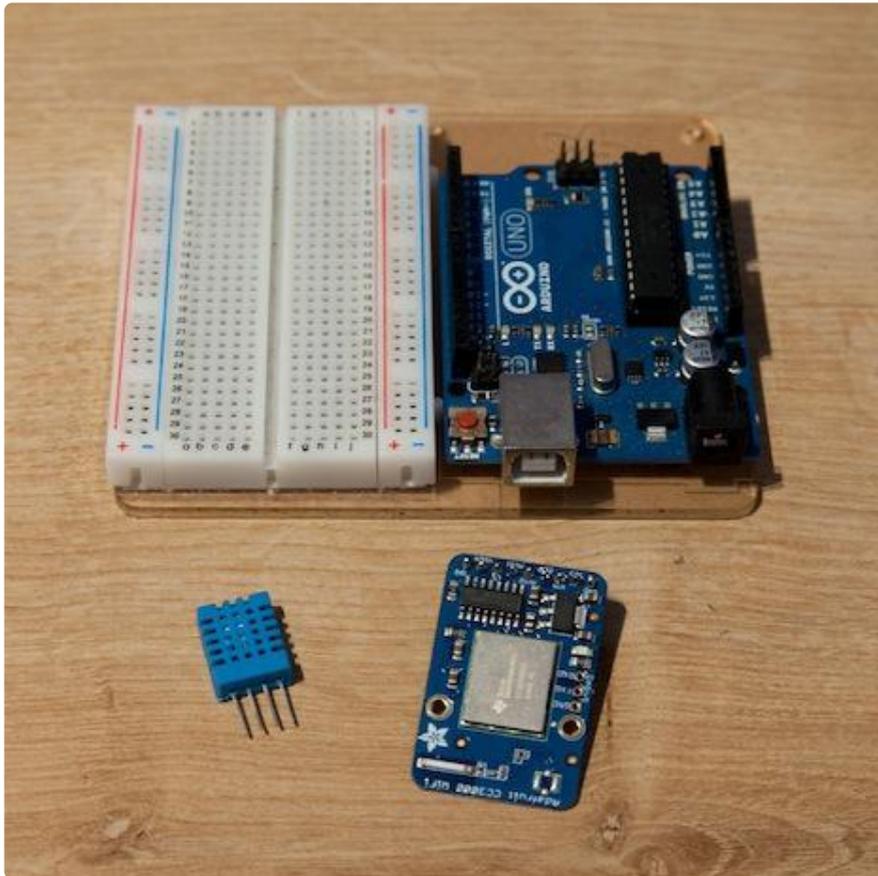


Connections

The whole project is based on the Arduino platform, so of course you will need an Arduino board. I really recommend using the Arduino Uno board for this project, as it

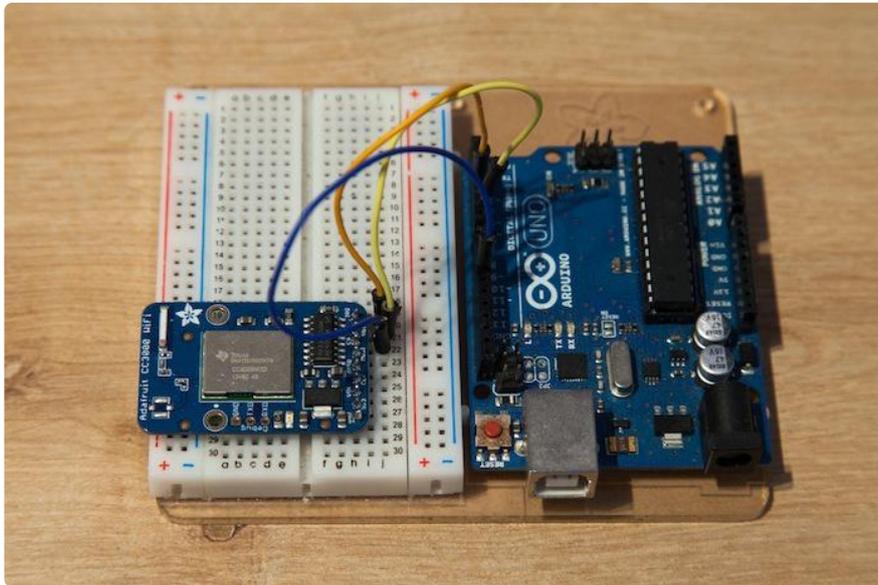
is the only board that is currently compatible with the CC3000 library at the time this tutorial was written.

Then, you need the [Adafruit CC3000 breakout board](http://adafru.it/1469) (<http://adafru.it/1469>) to make the WiFi communication, and the DHT11 temperature & humidity sensor (you can also use the DHT22 or AM2302 which are almost identical to wire up but higher quality). You also need a 10K Ohm resistor to be used with the DHT sensor. Finally, you need a breadboard and some jumper wires to make the connections between the different parts.

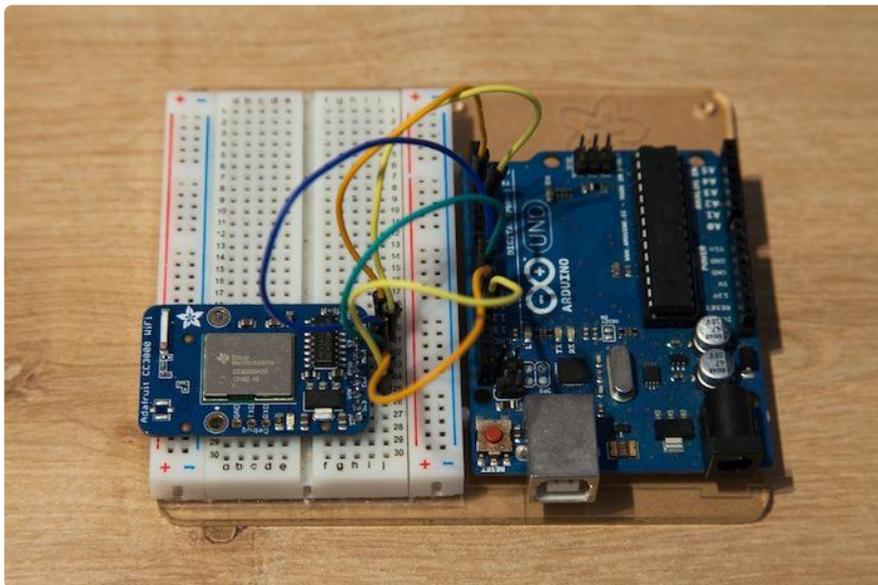


CC3000 Breakout Board

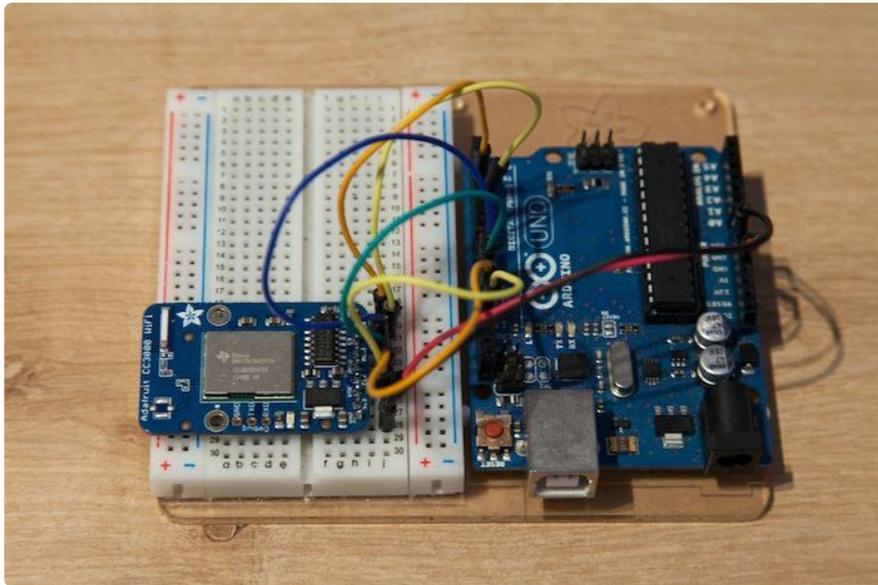
The hardware configuration of the CC3000 breakout board is relatively easy. Connect the IRQ pin of the CC3000 board to pin number 3 of the Arduino board, VBAT to pin 5, and CS to pin 10.



Then, you need to connect the SPI pins of the board to the corresponding pins on the Arduino board: MOSI, MISO, and CLK go to pins 11,12, and 13, respectively.

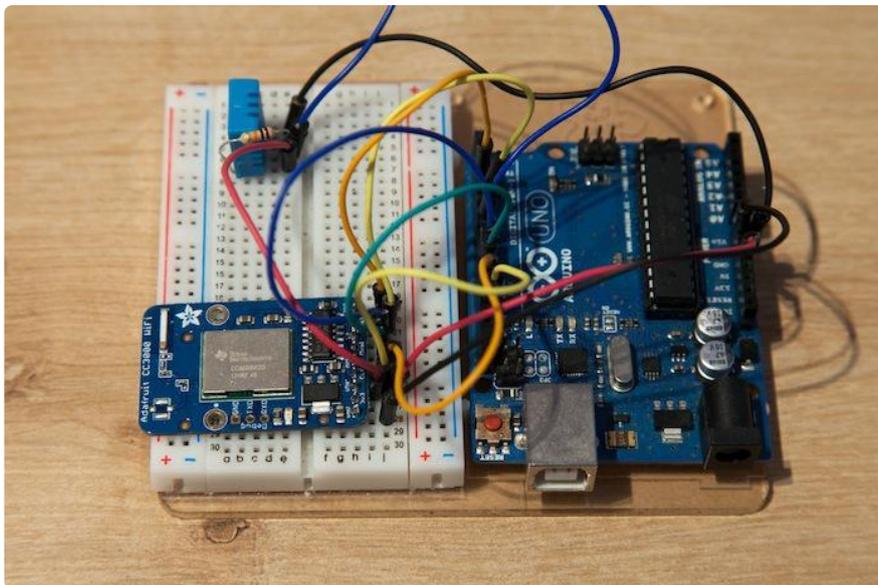


Finally, you have to take care of the power supply: Vin goes to the Arduino 5V, and GND to GND.

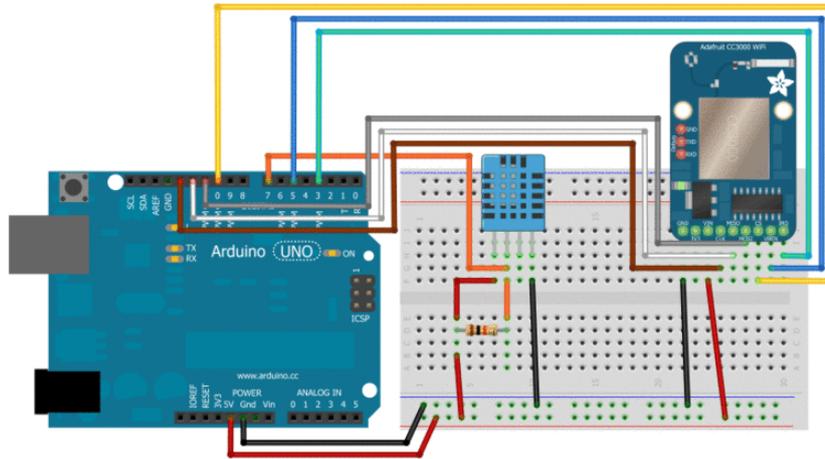


DHT11 sensor

The DHT sensor is much easier to connect: just plug the pin number 1 to the Arduino's 5V, pin number 4 to GND, and pin number 2 to Arduino pin 7. Finally, put the 10K resistor between the sensor pins number 1 and 2.



The following picture summarizes the hardware connections:



Testing the hardware

CC3000 WiFi chip

To see if your WiFi chip is correctly wired and operational, I recommend to use the test sketches that come with the Adafruit library that [you need to get from GitHub \(\)](#). I used for example the one called WebClient. Just open it from the library's folder, and save it to a new file (you need to be able to modify the sketch to enter your WiFi network name & password).

Then, modify the sketch with the correct data for you WiFi network, and upload to the board. You can then open your serial monitor, and if everything was wired correctly (and your Internet connection is working!) you should see your Arduino connecting to the web, then connecting to the Adafruit's website, and printing some information:

Hello, CC3000!

Free RAM: 1141 bytes

Initializing...

Started AP/SSID scan

Connecting to yourNetwork...Waiting to connect...Connected!

Request DHCP

IP Addr: 192.168.0.2

Netmask: 255.255.255.0

Gateway: 192.168.0.254

DHCPsrv: 192.168.0.254

DNSserv: 192.168.0.254

www.adafruit.com -> 207.58.139.247

Connect to 207.58.139.247:80

HTTP/1.1 200 OK

Date: Tue, 17 Sep 2013 09:23:39 GMT

Server: Apache

Access-Control-Allow-Origin: http://learn.adafruit.com

Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept,
Accept-Encoding, Authorization, Referer, User-Agent

Access-Control-Allow-Methods: GET, POST, OPTIONS

Access-Control-Allow-Credentials: true

Access-Control-Max-Age: 1728000

Last-Modified: Thu, 27 Jun 2013 14:13:27 GMT

Accept-Ranges: bytes

Content-Length: 74

Connection: close

Content-Type: text/html

This is a test of the CC3000 module!

If you can read this, its working :)

Disconnecting

DHT11 sensor

To use the DHT11 sensor, you need to [download the corresponding library \(\)](#) first and put it into your /libraries folder inside your Arduino folder. You can use the following sketch to test the DHT11 sensor, update it for DHT22 or AM2302 if you end up using one of those instead

```
// Include required libraries
#include <SPI.h>;
#include <string.h>;
#include "DHT.h"

// DHT11 sensor pins
#define DHTPIN 7
#define DHTTYPE DHT11

// DHT instance
DHT dht(DHTPIN, DHTTYPE);

void setup(void)
{
    // Initialize DHT sensor
    dht.begin();
}
```

```
Serial.begin(115200);  
}  
void loop(void)  
{  
    // Measure the humidity & temperature  
    float h = dht.readHumidity();  
    float t = dht.readTemperature();  
  
    // Transform to String  
    String temp = String((int) t);  
    String hum = String((int) h);  
  
    Serial.print("Temperature: ");  
    Serial.println(temp);  
    Serial.print("Humidity: ");  
    Serial.println(hum);  
    Serial.println("");  
}
```

Upload this sketch to the Arduino board, open the serial monitor and select 115200 baud, and this is what you should see:

Temperature: 21
Humidity: 23

Temperature: 21
Humidity: 23

Temperature: 21
Humidity: 22

Temperature: 21
Humidity: 23

Temperature: 21
Humidity: 23

Temperature: 21
Humidity: 23

Temperature: 21
Humidity: 23

If you can't communicate with the sensor, or the data looks really wrong, check your wiring and code until it works. It has to work before we continue!

Arduino sketch

We can now dive into the home automation part of this project. The goal is to get the sensor's data, send it via WiFi to a server (running on your computer), and display the information. The code for each part is quite long, so I will only discuss the important parts. To get the complete code, simply go to the [GitHub repository of the project](#) ().

To make this part work, you will also need to install the [CC3000 MDNS library](#) () and the [aREST library](#) ().

First, the Arduino sketch. You need to import the right libraries:

```
#include <Adafruit_CC3000.h>;
#include <SPI.h>;
#include <CC3000_MDNS.h>;
#include <aREST.h>;
#include "DHT.h"
```

Then, you need to define inside the code what is specific to your configuration, like your WiFi name & password:

```
#define WLAN_SSID "yourNetwork"
#define WLAN_PASS "yourPassword"
#define WLAN_SECURITY WLAN_SEC_WPA2
```

We can then create the CC3000 instance:

```
Adafruit_CC3000 cc3000 = Adafruit_CC3000(ADAFRUIT_CC3000_CS, ADAFRUIT_CC3000_IRQ,
    ADAFRUIT_CC3000_VBAT,
    SPI_CLOCK_DIV2);
```

We also need to create an instance of the aREST library, which we will use to access the data on the device:

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

We will also create a server running on our board:

```
Adafruit_CC3000_Server restServer(80);
```

And let's not forget to create the instance for the DHT sensor:

```
DHT dht(DHTPIN, DHTTYPE);
```

In the setup() part of the sketch, we need to initialize the DHT sensor:

```
dht.begin();
```

And to connect the CC3000 to the network:

```
cc3000.connectToAP(WLAN_SSID, WLAN_PASS, WLAN_SECURITY);
```

In the loop() part, we need to get data from the sensor, and then accept incoming connections to our server. The first part is simple:

```
temperature = (uint8_t)dht.readTemperature();  
humidity = (uint8_t)dht.readHumidity();
```

At this point you can already test the project. Upload the sketch to your Arduino board (make sure you changed the WiFi name & password!) and then open the Serial monitor. You should see the IP address of the board displayed.

Then, go a web browser, and type:

```
192.168.1.104/temperature
```

You should be greeted with the following message:

```
{"temperature": 24, "id": "1", "name": "weather_station", "connected": true}
```

If you can see that, it mean you project is working! If you want to learn more about the aREST framework that we are using here, you can visit the corresponding GitHub repository:

[https://github.com/marcoschwartz/aREST \(\)](https://github.com/marcoschwartz/aREST)

Building the web interface

It's now time to write the server part. Here, a combination of Node.js, Jade and Javascript is used. The whole code won't be detailed here, but once you download the code for this project, you will need to change one line in a file.

First, download all the files from the [GitHub repository of the project \(\)](#), and open the interface folder. In this folder, you will find a file called app.js. Open this file, and locate this line:

```
rest.addDevice('http', '192.168.1.103');
```

You simply need to enter the IP address of your CC3000 chip that you got before.

At this point, also make sure that Node.js is installed on your computer.

Using the weather station

It's now time to test the project! Make sure that the code is uploaded to the Arduino board, and go to the folder of the interface using your favorite Terminal software. Then, type:

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

That will install the required modules for Node.JS. Then, type:

```
node app.js
```

You should be greeted by a message:

```
Listening on port 3000
```

After that, go to this URL in your favorite web browser:

```
http://localhost:3000
```

Wait a bit so that the board can send the first set of data, and this is what you should see:

WiFi Weather Station

Temperature: 22 C Humidity: 39 % **Online**

If you can see the data being displayed, congratulations, your WiFi weather station is online! Of course, if you want to dive a bit into the Node.js code, you can add more measurements to this dashboard, from sensors on the same project or even from different boards in your home. Have fun!

Chances are you are running the webserver on a home or business computer that is behind a cable-modem or DSL-modem which has a firewall to prevent outsiders from getting to your machine. That means you can only use this setup inside your home. If you want to see data from your home network while at work, an external service is required which is not covered in this tutorial