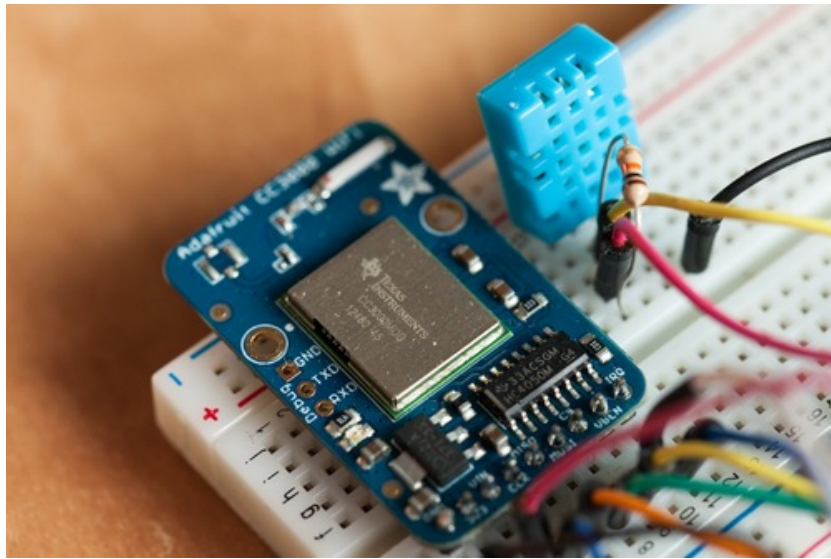


WiFi Weather Station

Created by Marc-Olivier Schwartz

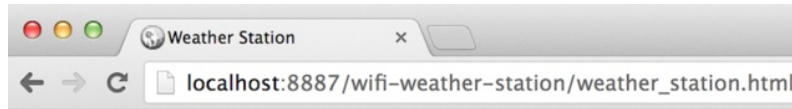


Last updated on 2013-10-14 12:00:30 PM EDT

Guide Contents

Guide Contents	2
Introduction	3
Connections	4
CC3000 Breakout Board	4
DHT11 sensor	6
Testing the hardware	8
CC3000 WiFi chip	8
DHT11 sensor	9
Arduino sketch	11
Building the web interface	14
Using the weather station	16

Introduction



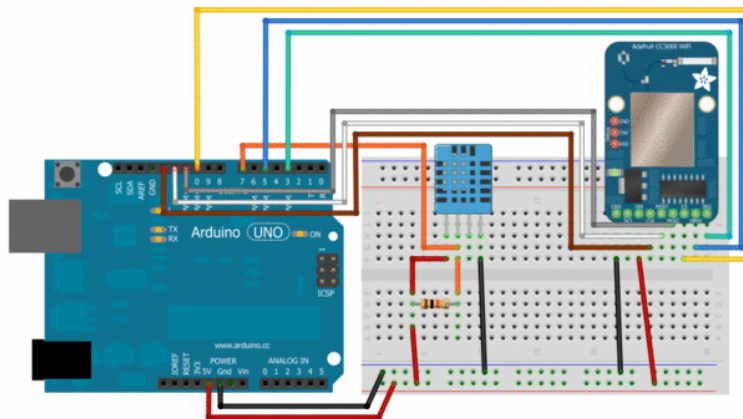
Temperature: 21

Humidity: 22

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](http://adafru.it/cHf) (<http://adafru.it/cHf>). 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 PHP server script

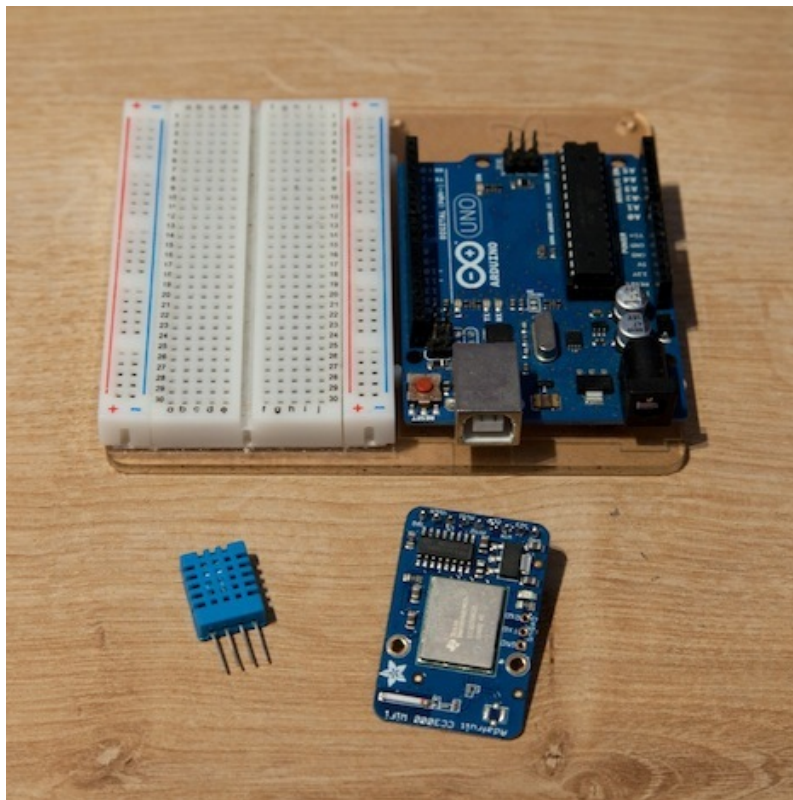


Connections

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.

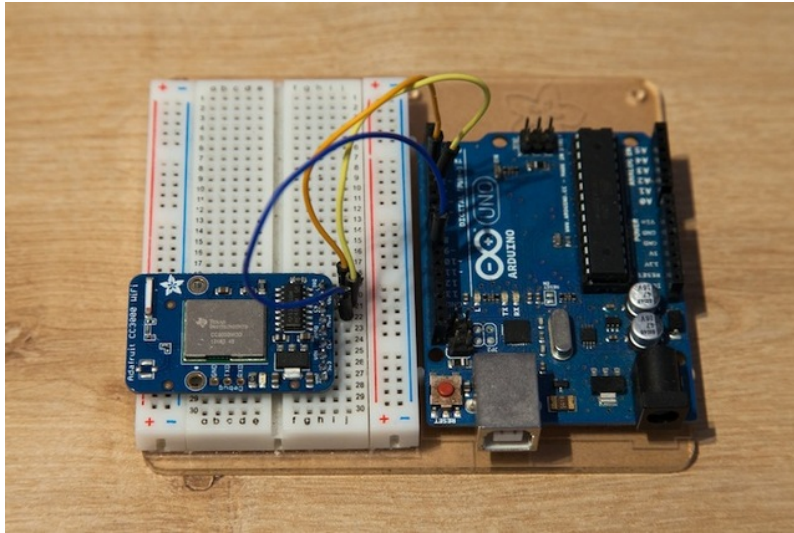
Then, you need the [Adafruit CC3000 breakout board](http://adafru.it/1469) (<http://adafru.it/1469>) to make the WiFi communication, and the DHT11 temperature & humidity sensor (you can also use the DHT22 or AM2302 which are almost identical to wire up but higher quality). You also need a 10K Ohm resistor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

Finally, you need a breadboard and some jumper wires to make the connections between the different parts.

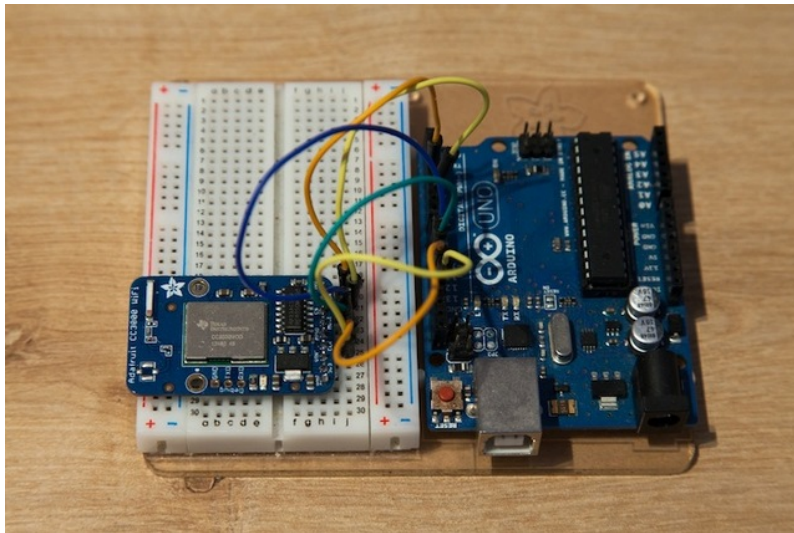


CC3000 Breakout Board

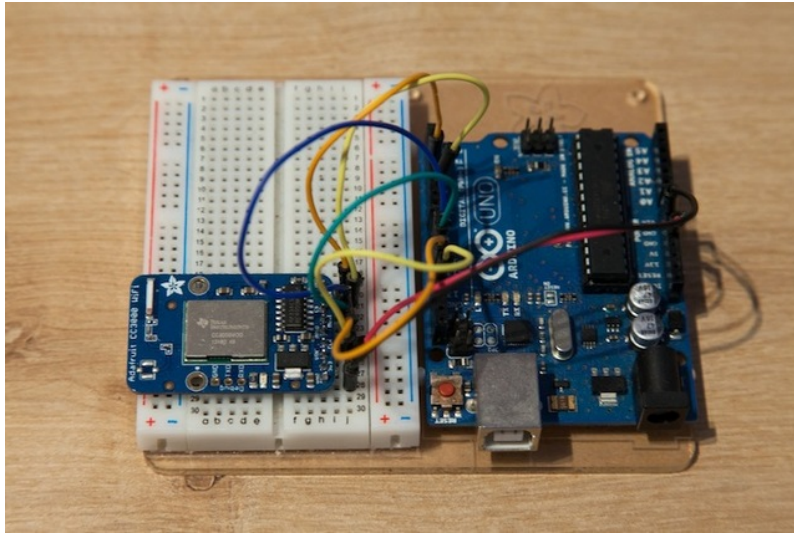
The hardware configuration of the CC3000 breakout board is relatively easy. Connect the IRQ pin of the CC3000 board to pin number 3 of the Arduino board, VBAT to pin 5, and CS to pin 10.



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.

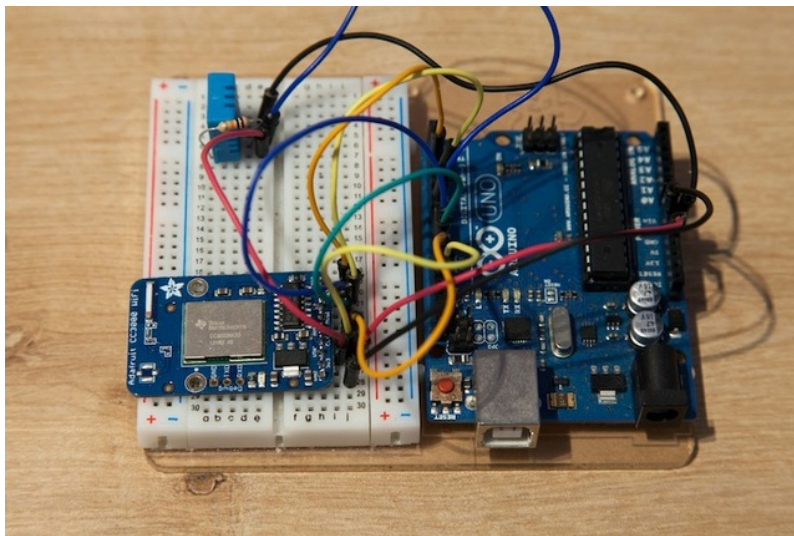


Finally, you have to take care of the power supply: Vin goes to the Arduino 5V, and GND to GND.

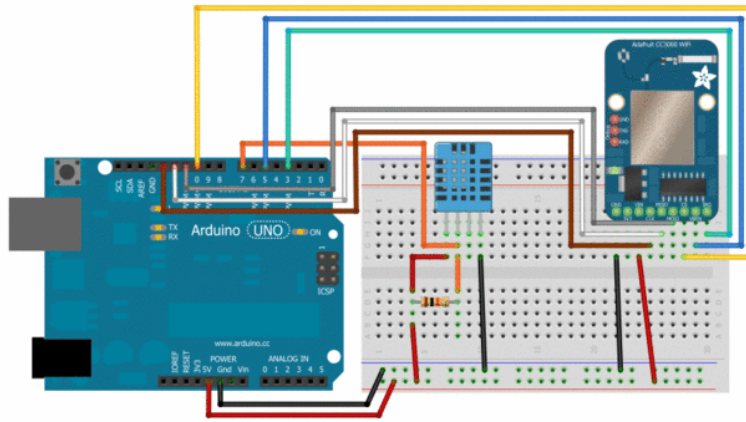


DHT11 sensor

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 \(http://adafru.it/cFn\)](http://adafru.it/cFn). 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 \(http://adafru.it/aJX\)](http://adafru.it/aJX) 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 \(http://adafru.it/cji\)](http://adafru.it/cji).

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

```
#include <Adafruit_CC3000.h>
#include <ccspi.h>
#include <SPI.h>
#include <string.h>
#include "utility/debug.h"
#include "DHT.h"
#include <stdlib.h>
```

Then, you need to define inside the code what is specific to your configuration: WiFi name & password, IP address of your computer, and port of your server (usually 80):

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

uint32_t ip = cc3000.IP2U32(192,168,1,2);
int port = 80;
```

We can then create the CC3000 instance:

```
Adafruit_CC3000 cc3000 = Adafruit_CC3000(ADAFRUIT_CC3000_CS, ADAFRUIT_CC3000_IRQ, ADAFRUIT_CC3000_SPI_CLOCK_DIV2);
```

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 to send it to the server by doing a GET request. The first part is really simple:

```
float h = dht.readHumidity();  
float t = dht.readTemperature();
```

You just need to take care of converting these floats to Strings, because you want to insert this information into an HTTP request. In this case, we just convert the floats to integers, and then transform them to strings:

```
String temperature = String((int) t);  
String humidity = String((int) h);
```

Finally, the request is sent using the send_request() function:

```
String request = "GET " + repository + "sensor.php?temp=" + temperature + "&hum=" + humidity  
send_request(request);
```

This function basically connects to the server, sends the request, and close the connection. It starts by connecting to the server with:

```
Adafruit_CC3000_Client client = cc3000.connectTCP(ip, port);
```

And then send the request with this line:

```
client.println(request);  
client.println(F(""));
```

Finally, it reads out the answer (which we don't use in this project, but you could perfectly get data back from the server):

```
while (client.available()) {  
  // Read answer  
  char c = client.read();  
}
```

We then wait 10 seconds between each temperature & humidity update. Once again, to get the complete code, just go to the [GitHub repository of the project \(http://adafru.it/cji\)](http://adafru.it/cji).

Building the web interface

It's now time to write the server part. Here, a combination of HTML, JavaScript, and PHP is used. The PHP part will receive the data from the board at each GET request, and save it into files:

```
<?php
// Store data
if ($_GET["temp"] && $_GET["hum"]) {

    $myFile = "temp_data.txt";
    $fh = fopen($myFile, 'w');
    fwrite($fh, $_GET["temp"]);
    fclose($fh);

    $myFile = "hum_data.txt";
    $fh = fopen($myFile, 'w');
    fwrite($fh, $_GET["hum"]);
    fclose($fh);
}
?>
```

Of course, you could use more elaborate ways to save the data like XML or SQL but let's keep things simple. Then, two other PHP scripts actually display the data on the main HTML page. This is for example the one for the temperature:

```
<?php
$myFile = "temp_data.txt";
$fh = fopen($myFile, 'r');
$line = fgets($fh);
fclose($fh);
echo $line;
?>
```

Finally, the main page just organizes all this information in two separate blocks. This is for example one of them:

```
<div class="data">
<div class="dataTitle">Temperature: </div>
<div id="temperatureDisplay">Waiting for data ...</div>
</div>
```

There is also a simple JavaScript code to update the display every second, just where we need it:

```
setInterval(function()
```

```
{  
  $("#temperatureDisplay").load('temperature_display.php');  
  $("#humidityDisplay").load('humidity_display.php');  
}, 1000);
```

It is based on jQuery, which is also included in the [GitHub repository of the project \(http://adafru.it/cjli\)](http://adafru.it/cjli). There is also some CSS in the page so it doesn't look too ugly, you can look at the details in the repository.

To use this code, you will need to have a web server running. I recommend [EasyPHP \(http://adafru.it/cjj\)](http://adafru.it/cjj) if you are using Windows, [MAMP \(http://adafru.it/cjk\)](http://adafru.it/cjk) if you are under OS X, and simply running Apache if you are running Linux. Please make sure your web server is running before going further, and put all the files of this project into a folder of your web server.

Using the weather station

That's all for the software part, so it is time to test it out! Make sure that your web server is running. Upload the sketch to the Arduino board, and open the serial monitor to check that everything is working correctly. This is what you should see:

Started AP/SSID scan

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

Request DHCP

Temperature: 21

Humidity: 22

Starting connection to server...

Connect to 192.168.0.1:8887

Connected & Data sent

Closing connection

Temperature: 21

Humidity: 22

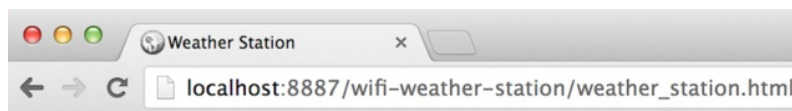
Starting connection to server...

Connect to 192.168.0.1:8887

Connected & Data sent

Closing connection

Now, you can also open the html page inside your favorite browser. Wait a bit so that the board can send the first set of data, and this is what you should see:



Temperature: 21

Humidity: 22

This page is now displaying the data from the DHT11 sensor, and is updating automatically every second.

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

