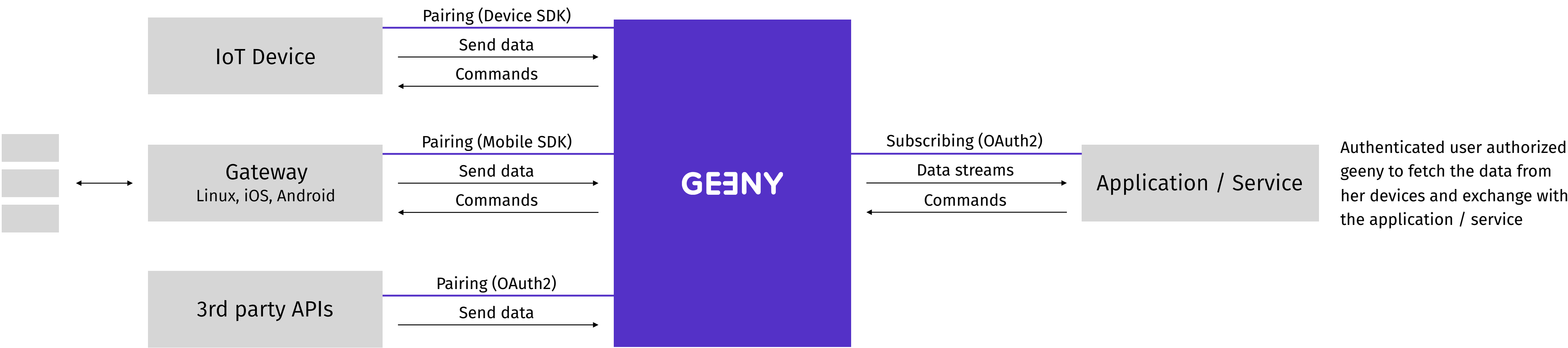


# **Intro to Making** • How to connect the NodeMCU to Geeny

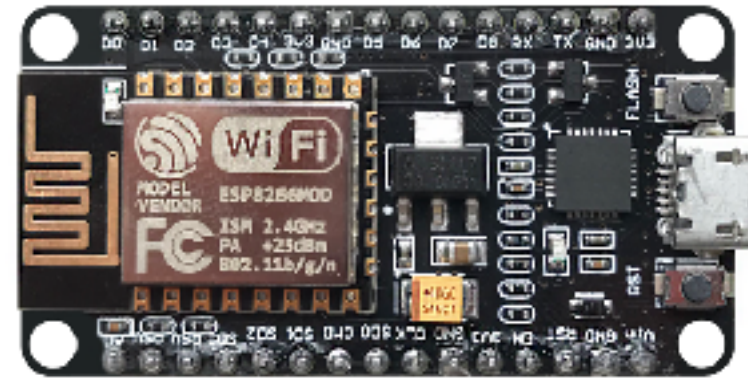
8.3.2018

# What is Geeny?

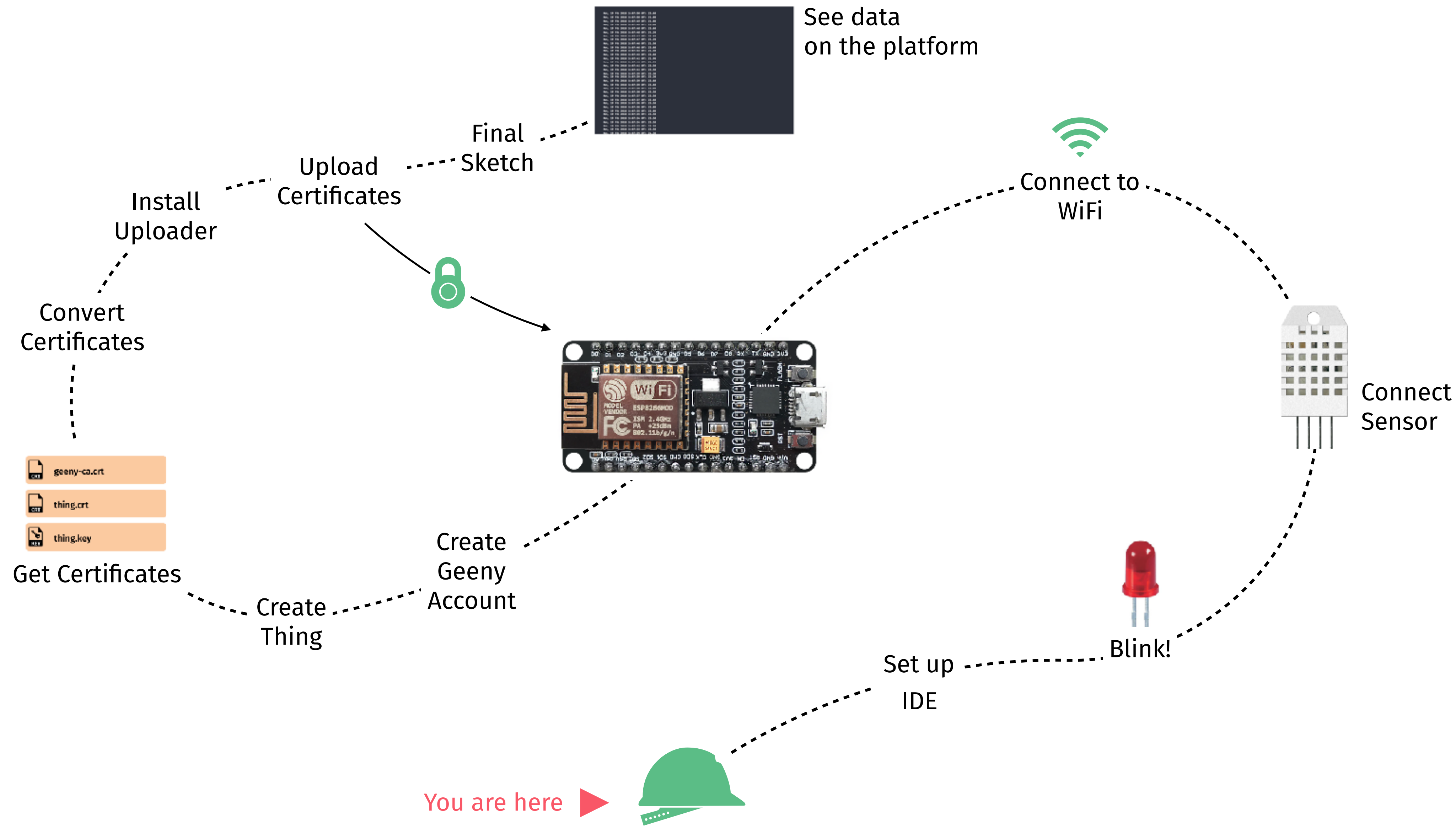
Geeny is the consumer IoT platform by Telefónica NEXT



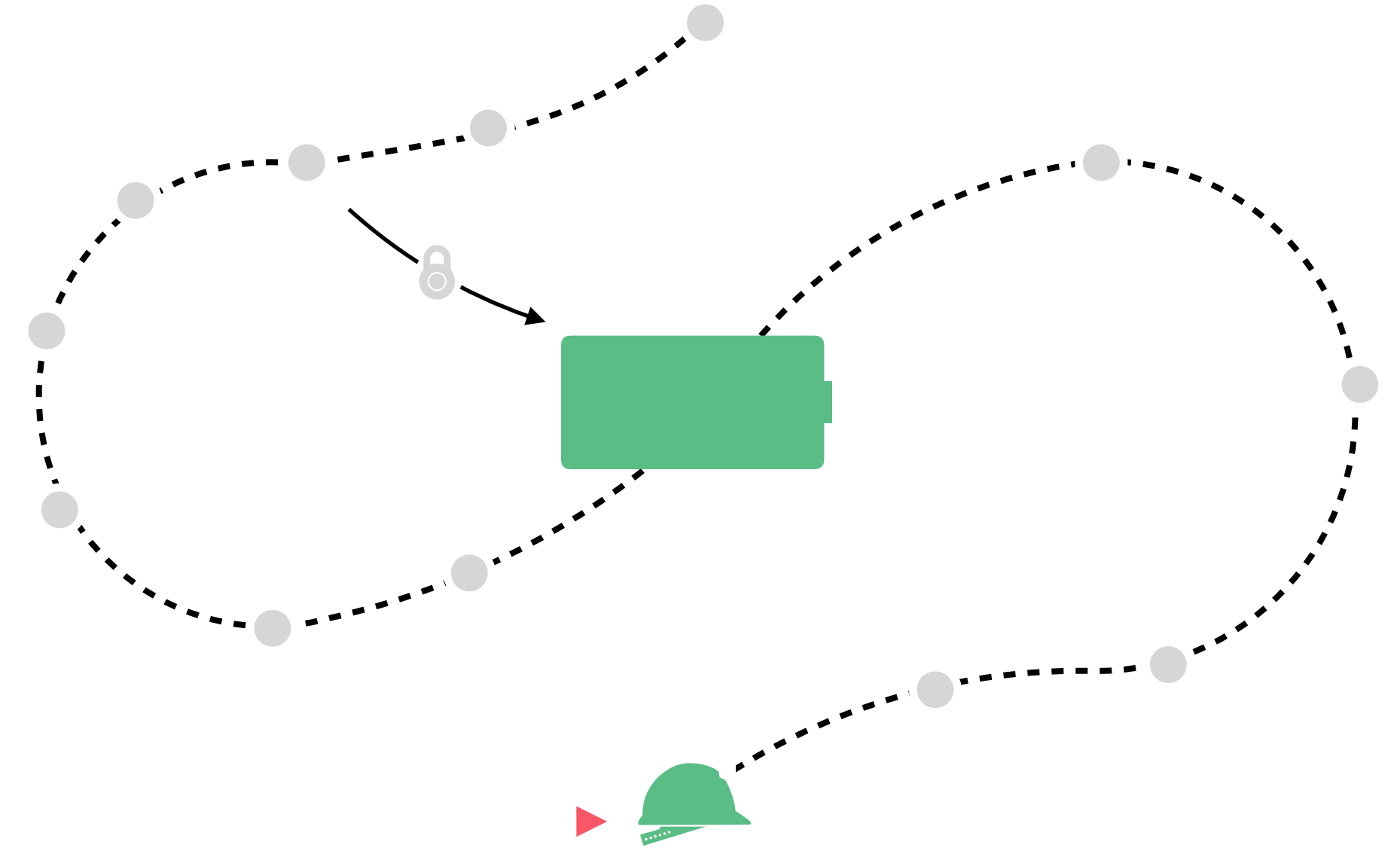
# The Journey



# The Journey



## First Steps



## First Steps

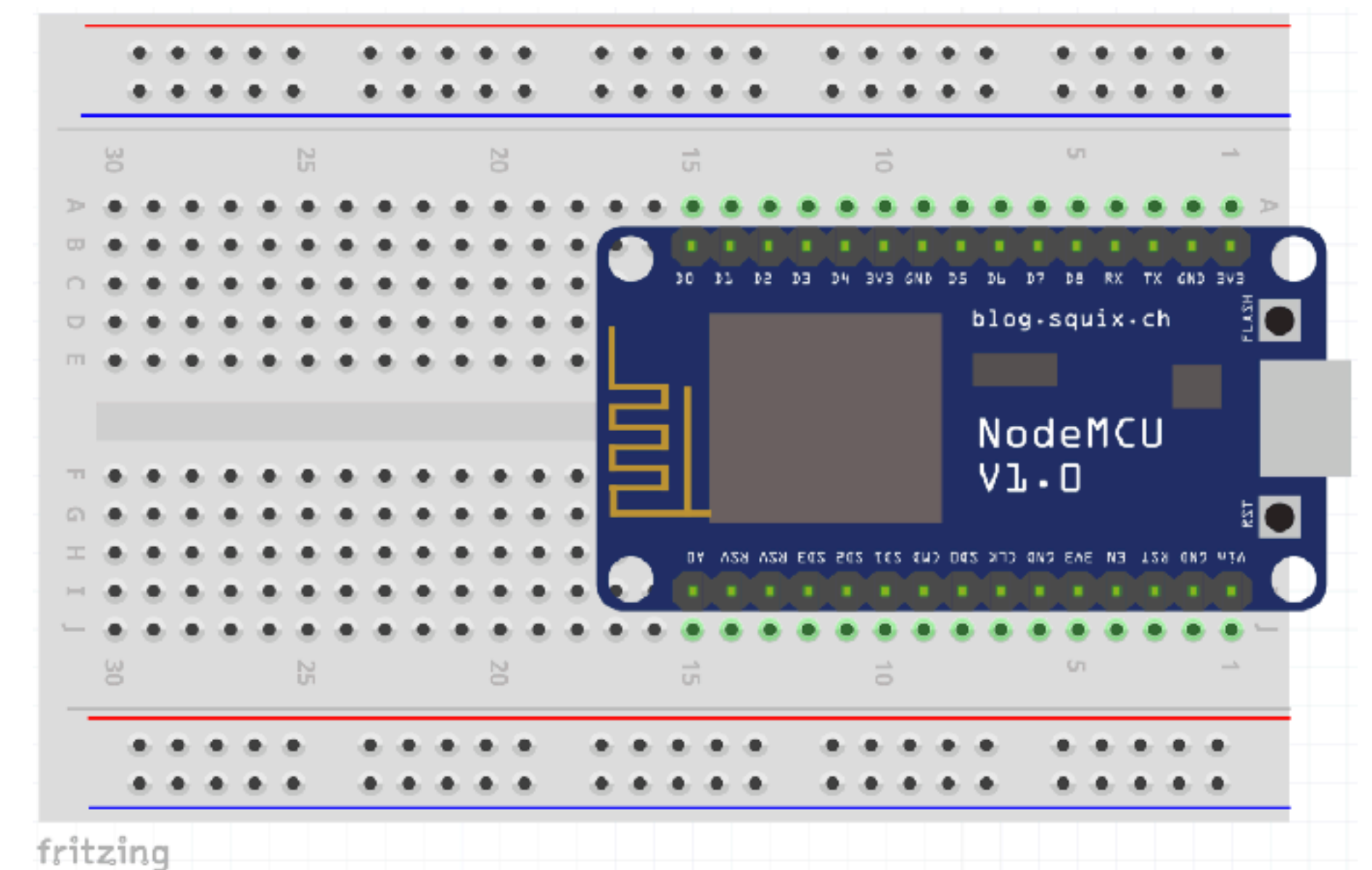
### 1. Set up the IDE

1. Install Arduino IDE from [www.arduino.cc](http://www.arduino.cc)
2. Go to *Files>Preference* (Mac: *Arduino>Preferences*) in the Arduino IDE
3. Copy the below link in the *Additional boards Manager* field:  
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)
4. Go to *Tools>Boards>Board Manager*
5. Search for [esp8266](#) (by *esp8266 community*) and install it
6. Select from *Tools>Board* the [NodeMCU 1.0 \(ESP12E module\)](#)
7. Maybe: Install USB driver from  
<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>
8. Copy the content of the Arduino Sketchfolder in this repository into your Arduino Sketchfolder

## First Steps

### 2. Blink()

1. Open *Files>Examples>Basics>Blink*
2. Go to *Tools>Port* and select (usually) the last entry
3. Upload the code to the NodeMCU:  
*Sketch>Upload* or click the upload button in the menu

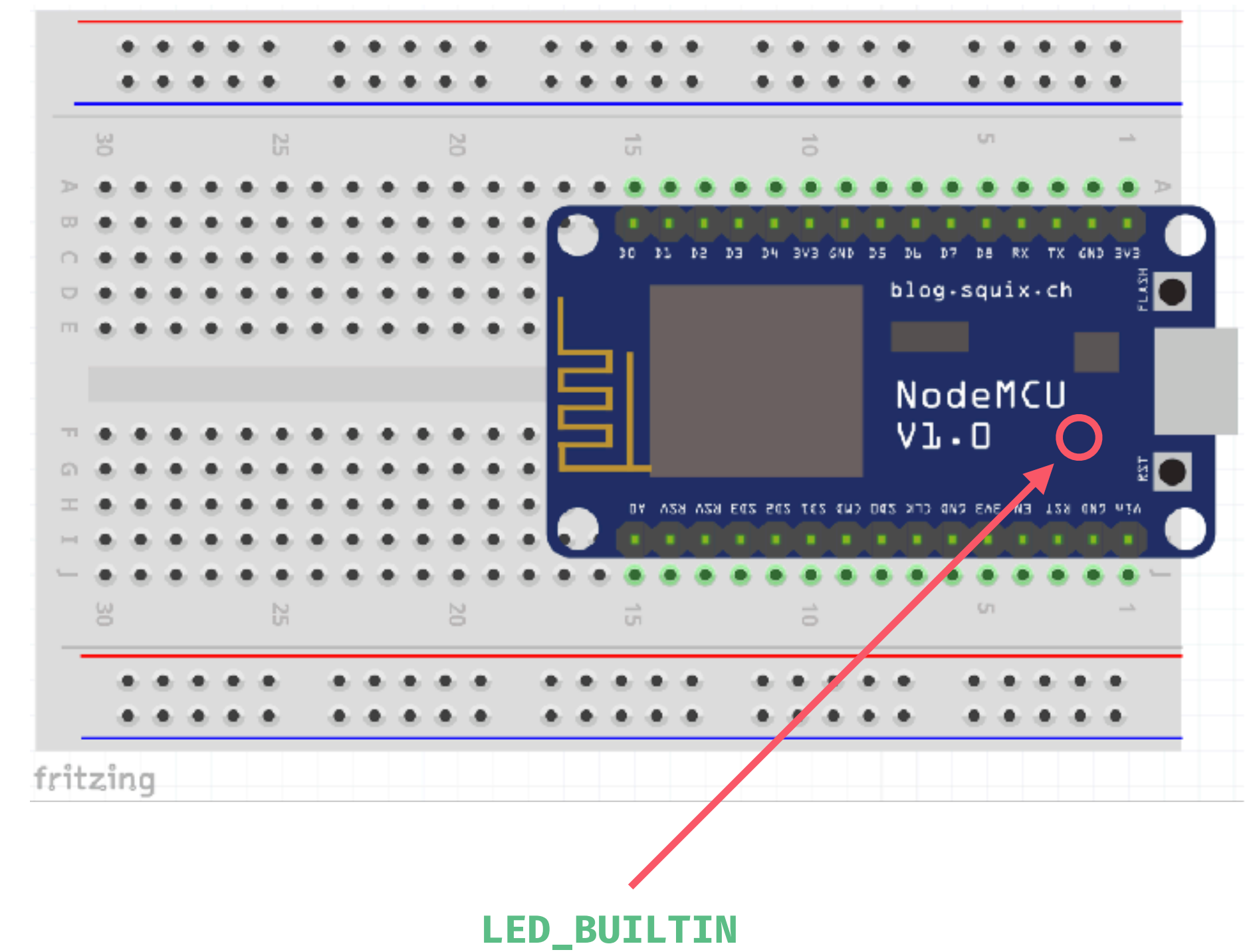




## First Steps

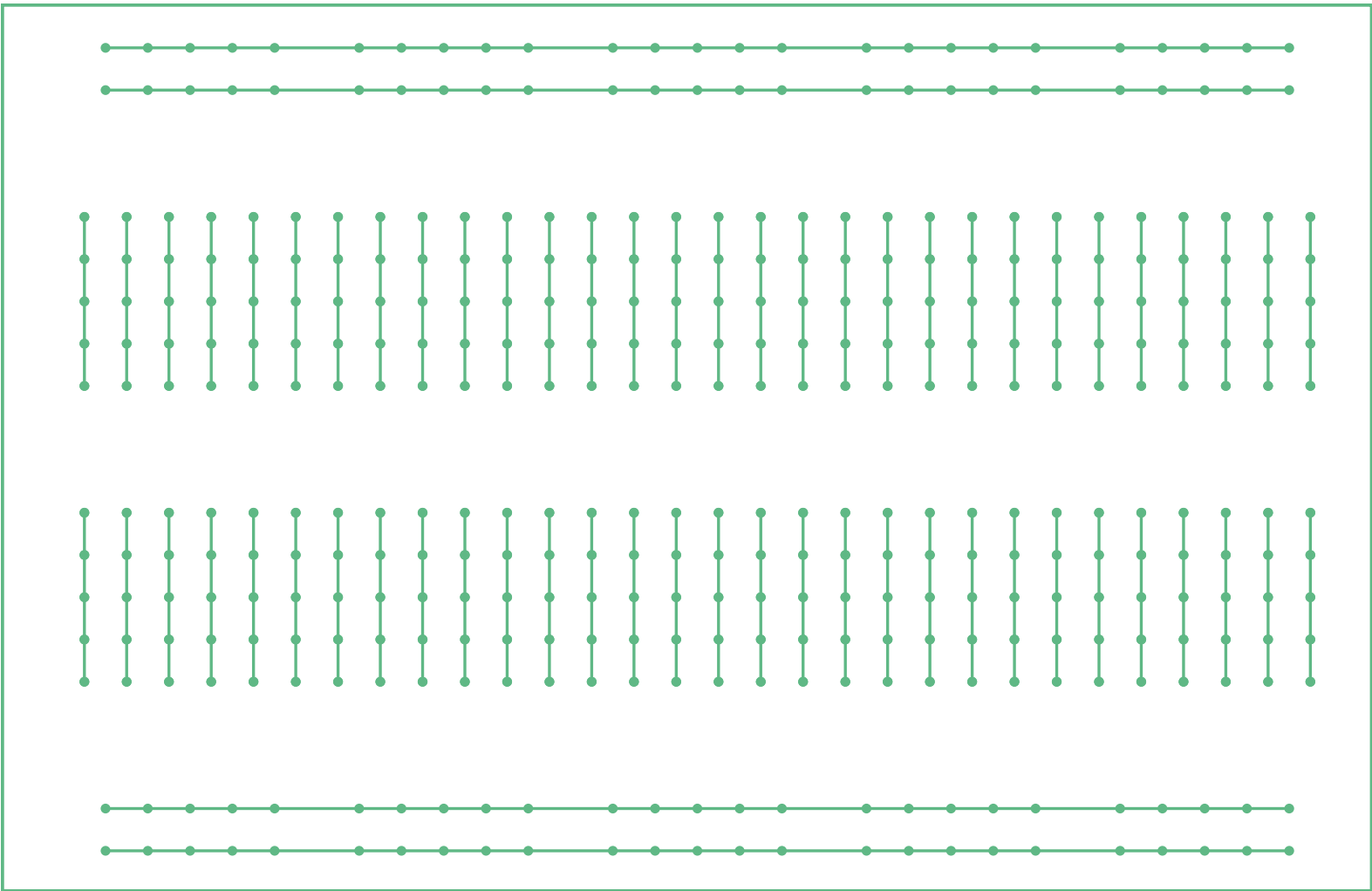
### 2. Blink()

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

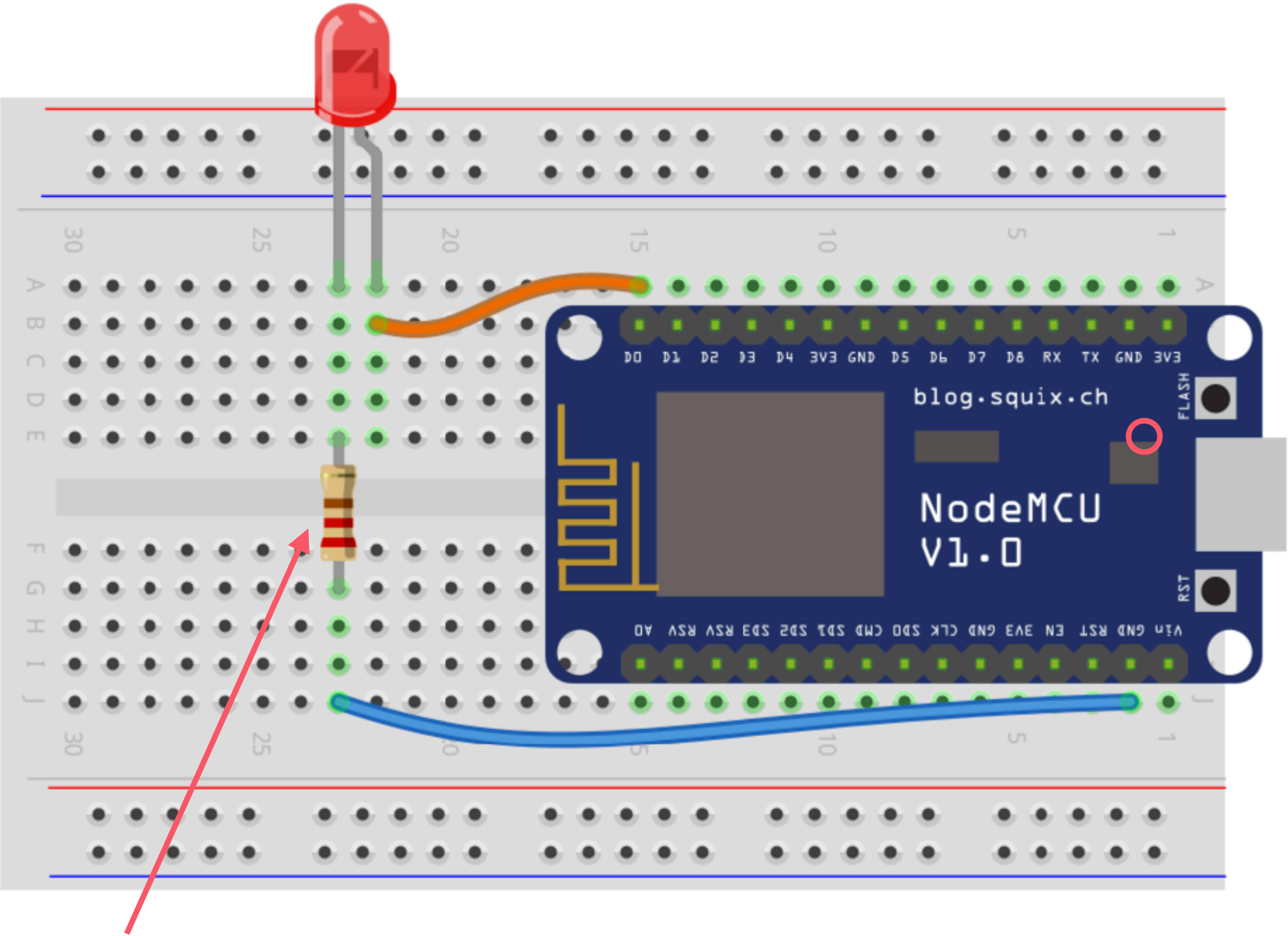




# First Steps



Interconnections of the Breadboard



220  $\Omega$  resistor



4 band resistor (red, red, brown) or



5 band resistor (red, red, black, black)

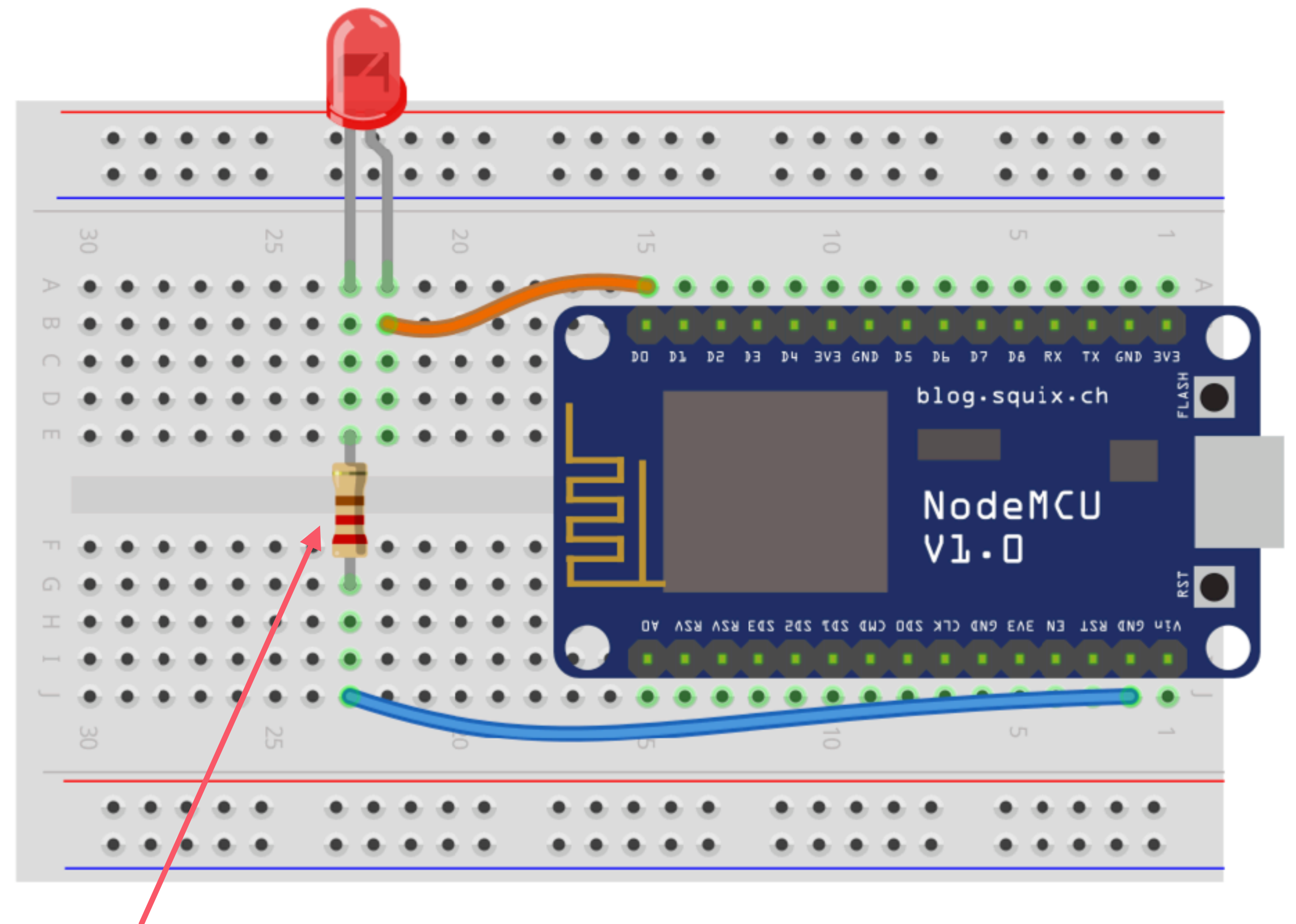
## First Steps

### 2. Blink()

```
int ledPin = D0;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```



220  $\Omega$  resistor

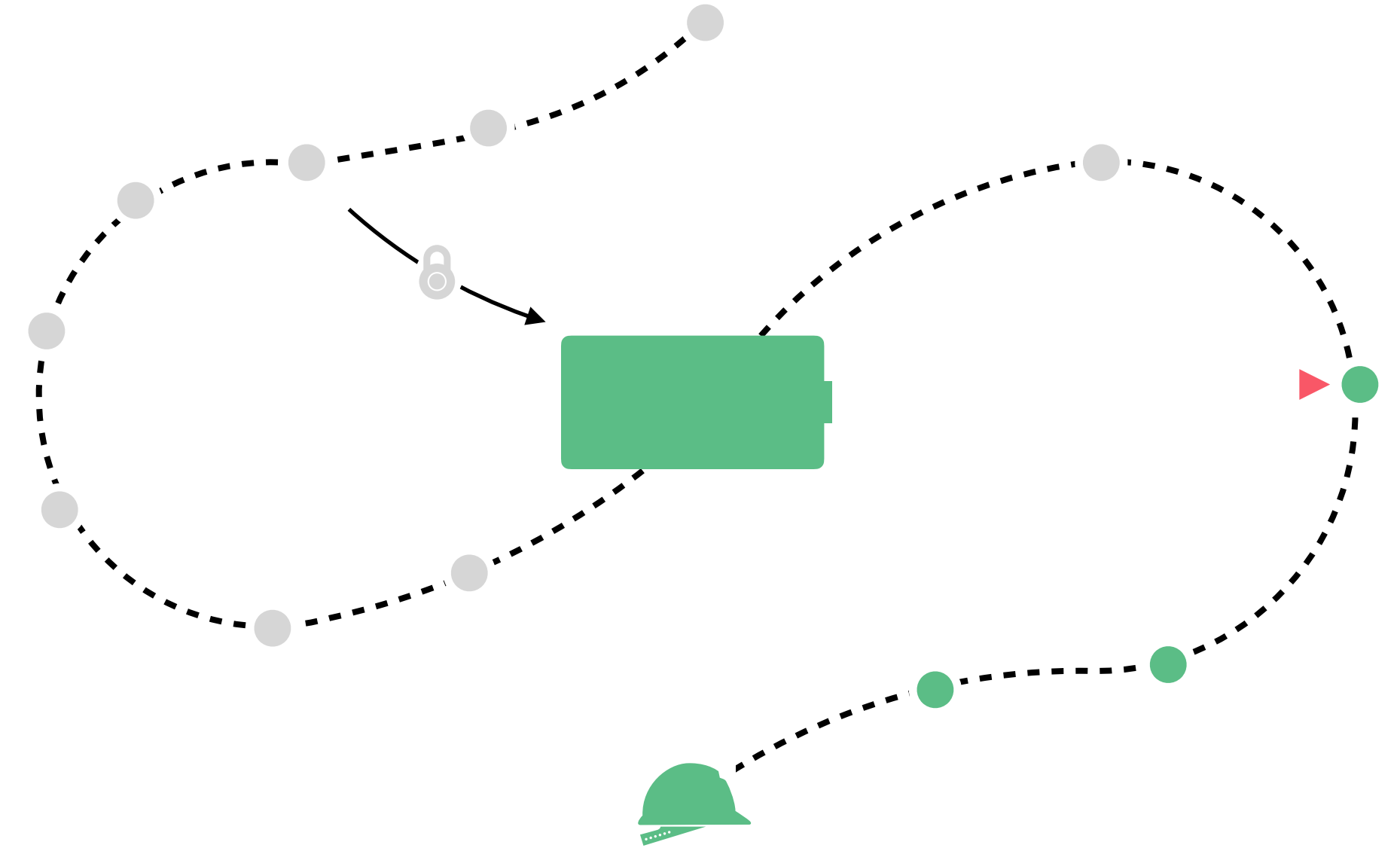


4 band resistor (red, red, brown) or



5 band resistor (red, red, black, black)

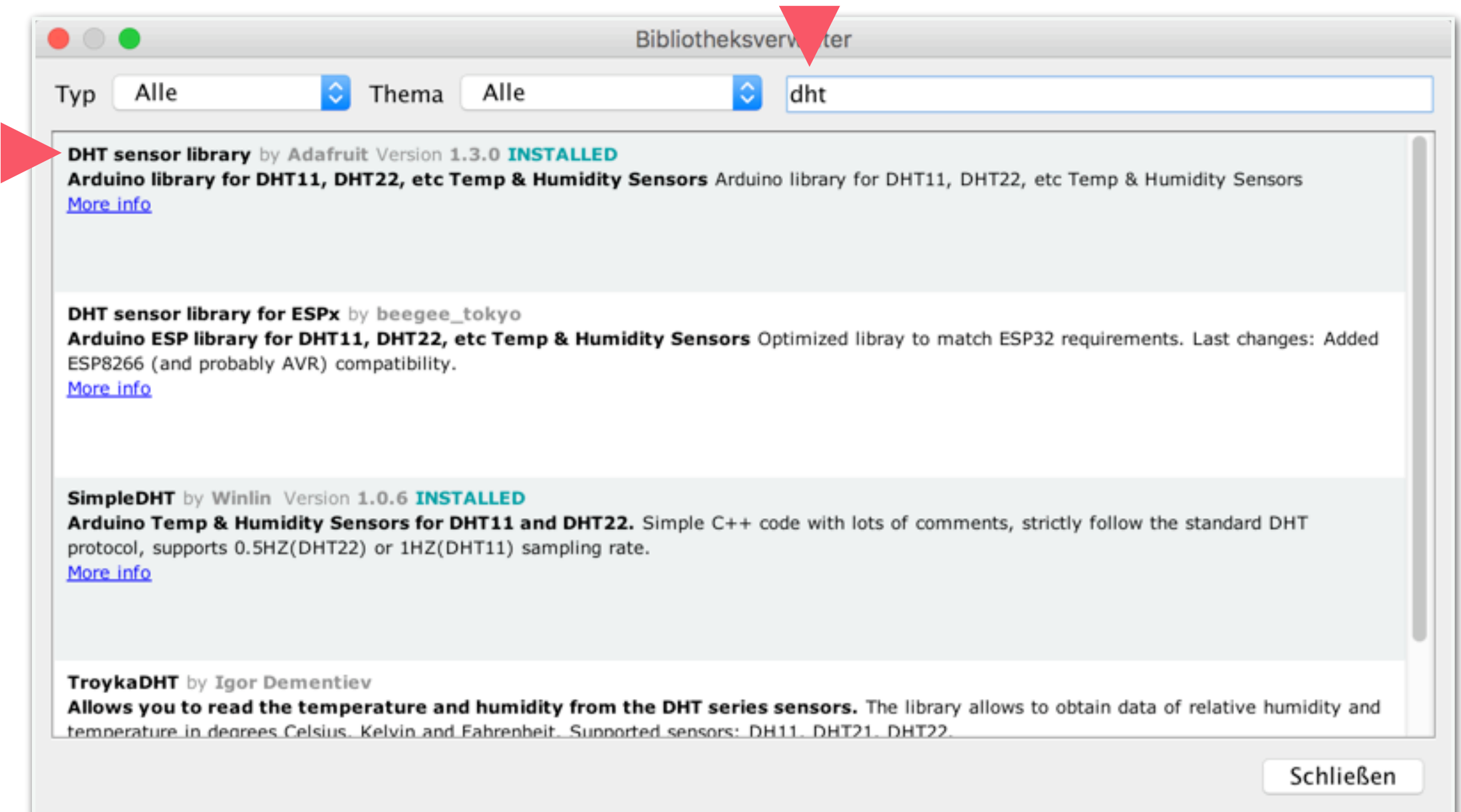
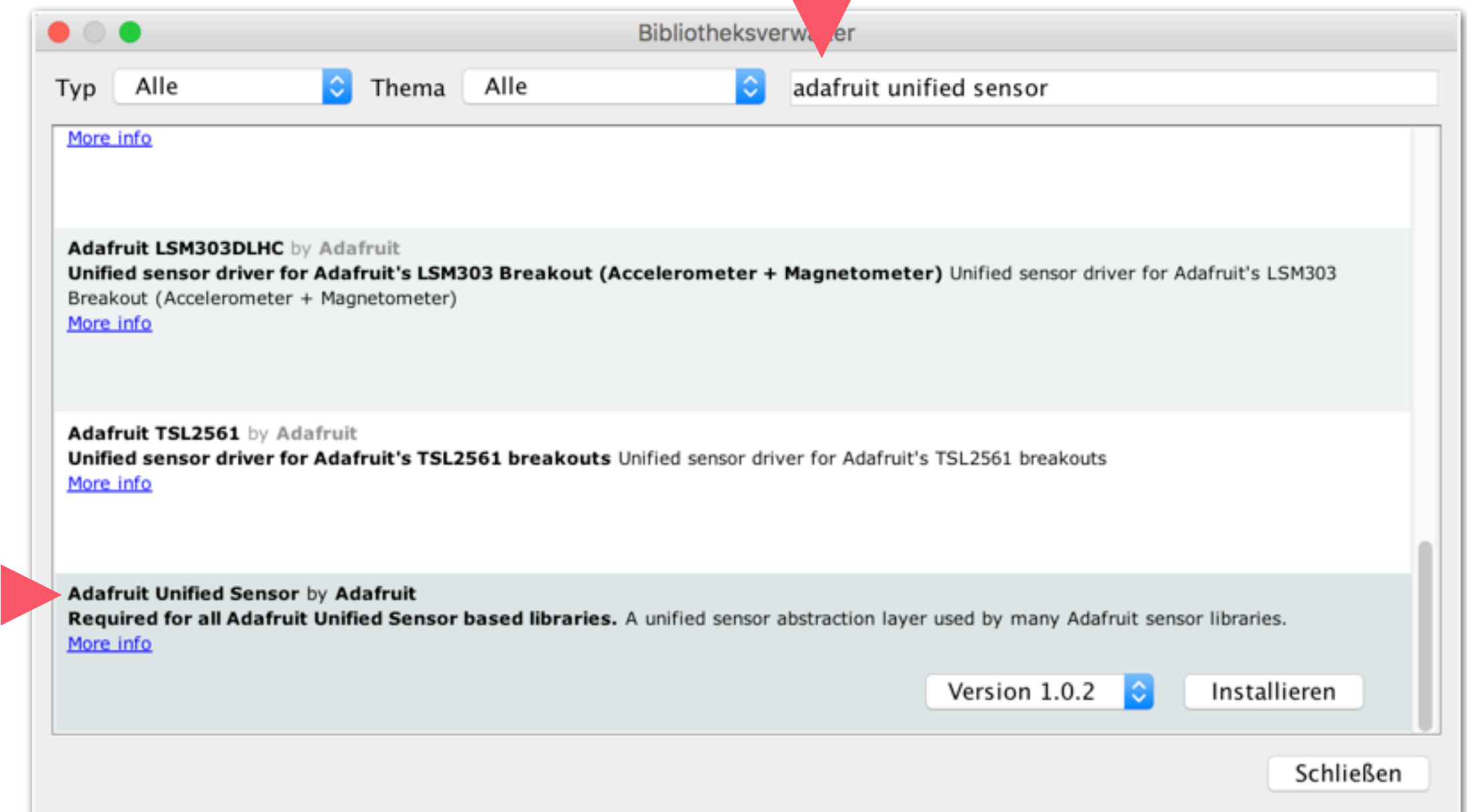
## DHT22 Temperature and Humidity



## DHT22 Temperature and Humidity

### 1. Install library

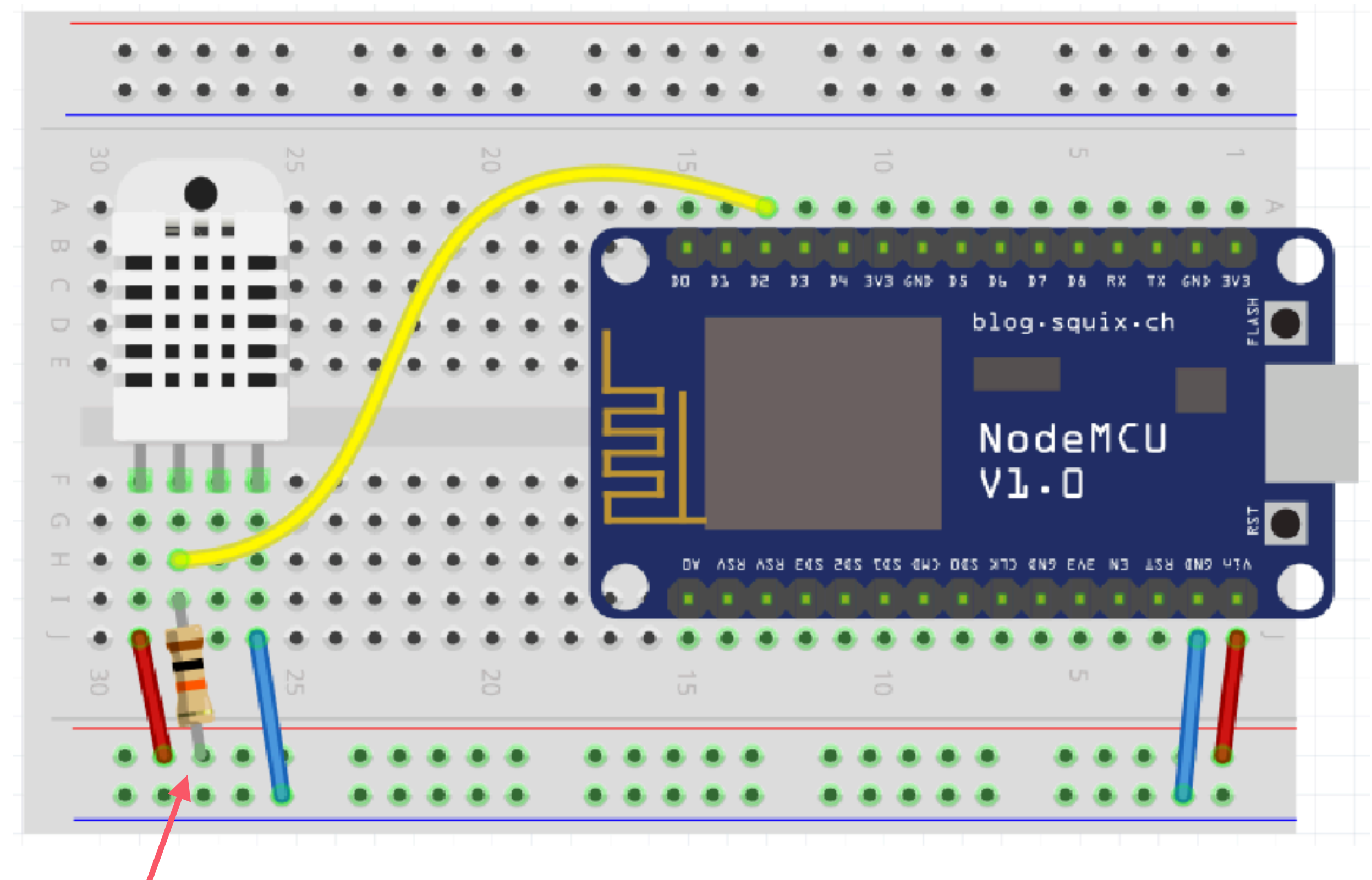
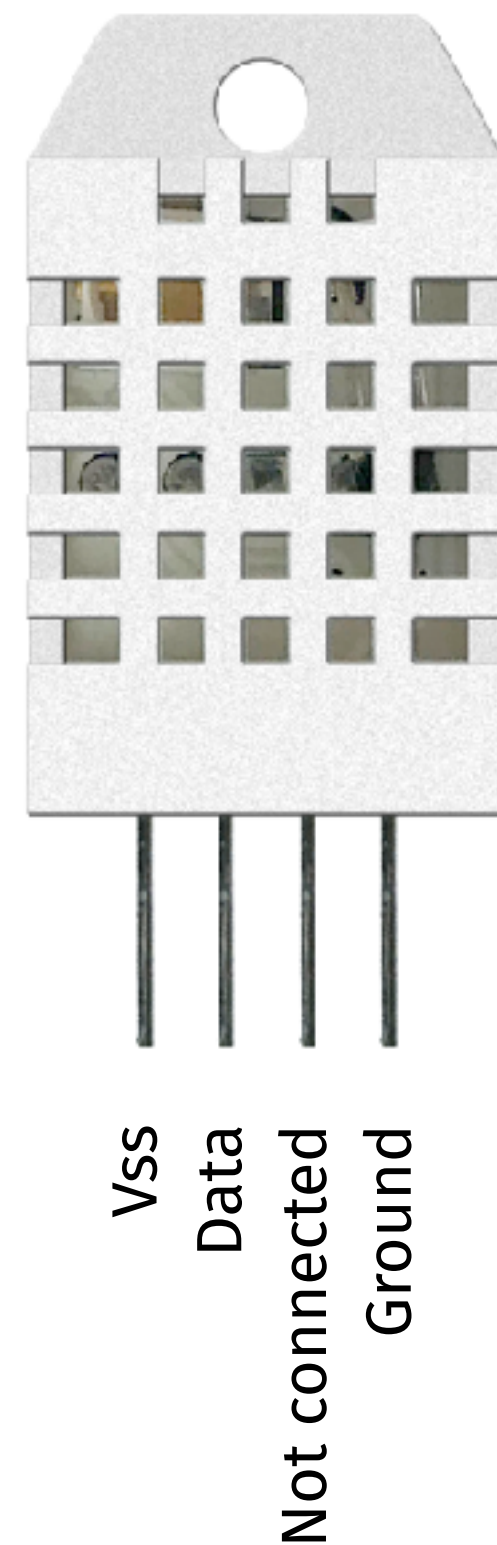
1. In the Arduino IDE go to *Sketch>Include Library>Manage libraries ...* and search for **adafruit unified sensor**
2. Install the **Adafruit Unified Sensor by Adafruit** library
3. Once again but this time search for **dht**
4. Install the **DHT sensor library by Adafruit** library





## DHT22 Temperature and Humidity

## 2. Build the circuit



10 k $\Omega$  resistor



4 band resistor (brown, black, orange) or



5 band resistor (brown, black, black, red)

## DHT22 Temperature and Humidity

### 3. The sketch (2\_DHT22.ino)

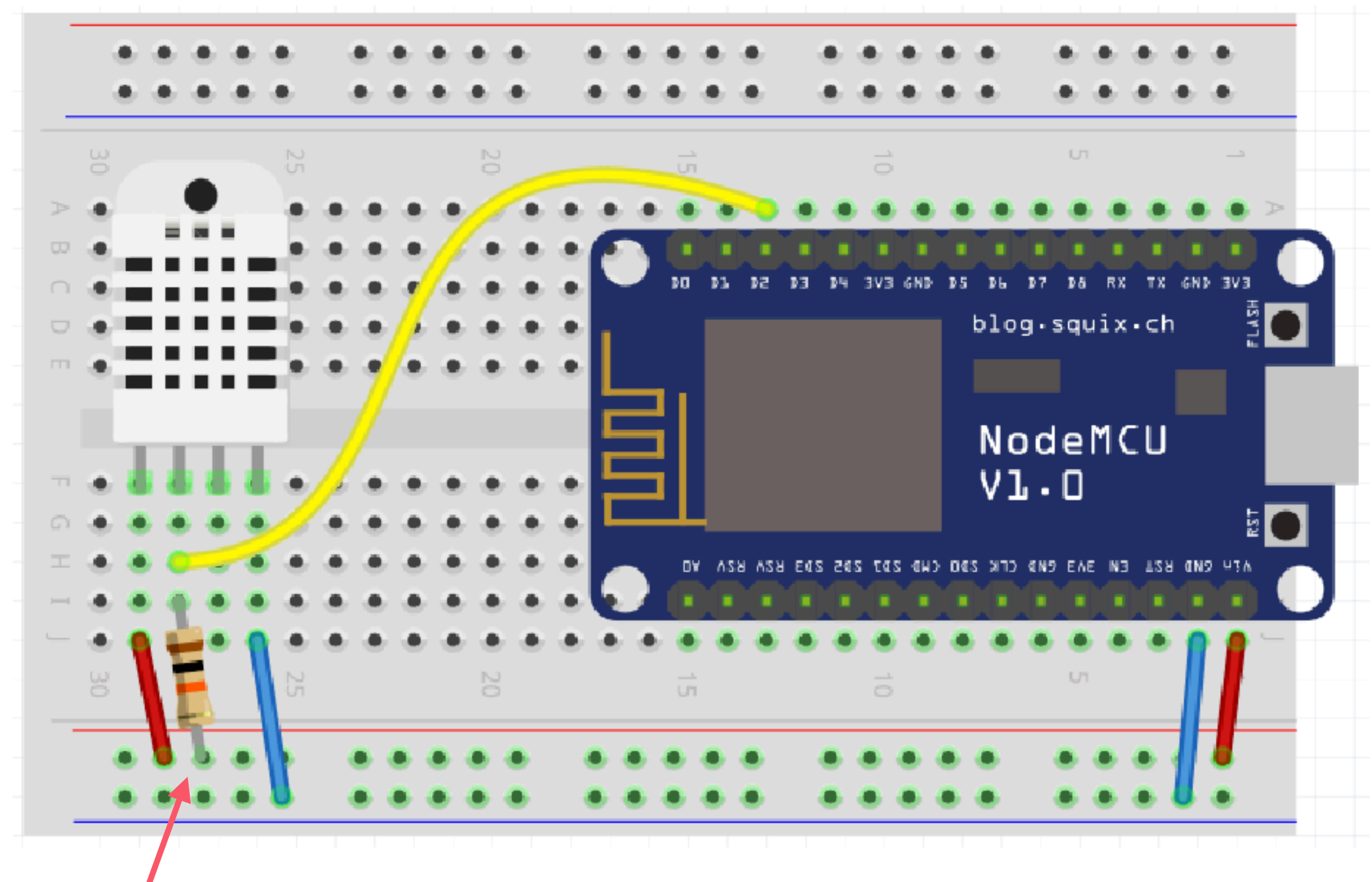
```
#include <ESP8266WiFi.h>
#include <DHT.h>

#define DHTPIN D2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

unsigned long lastMillis = 0;

void setup() {
  Serial.begin(115200);
  Serial.println("Startup");
}

void loop() {
  if (millis() - lastMillis > 1000) {
    lastMillis = millis();
    sendSensorData();
  }
}
```



10 kΩ resistor



4 band resistor (brown, black, orange) or

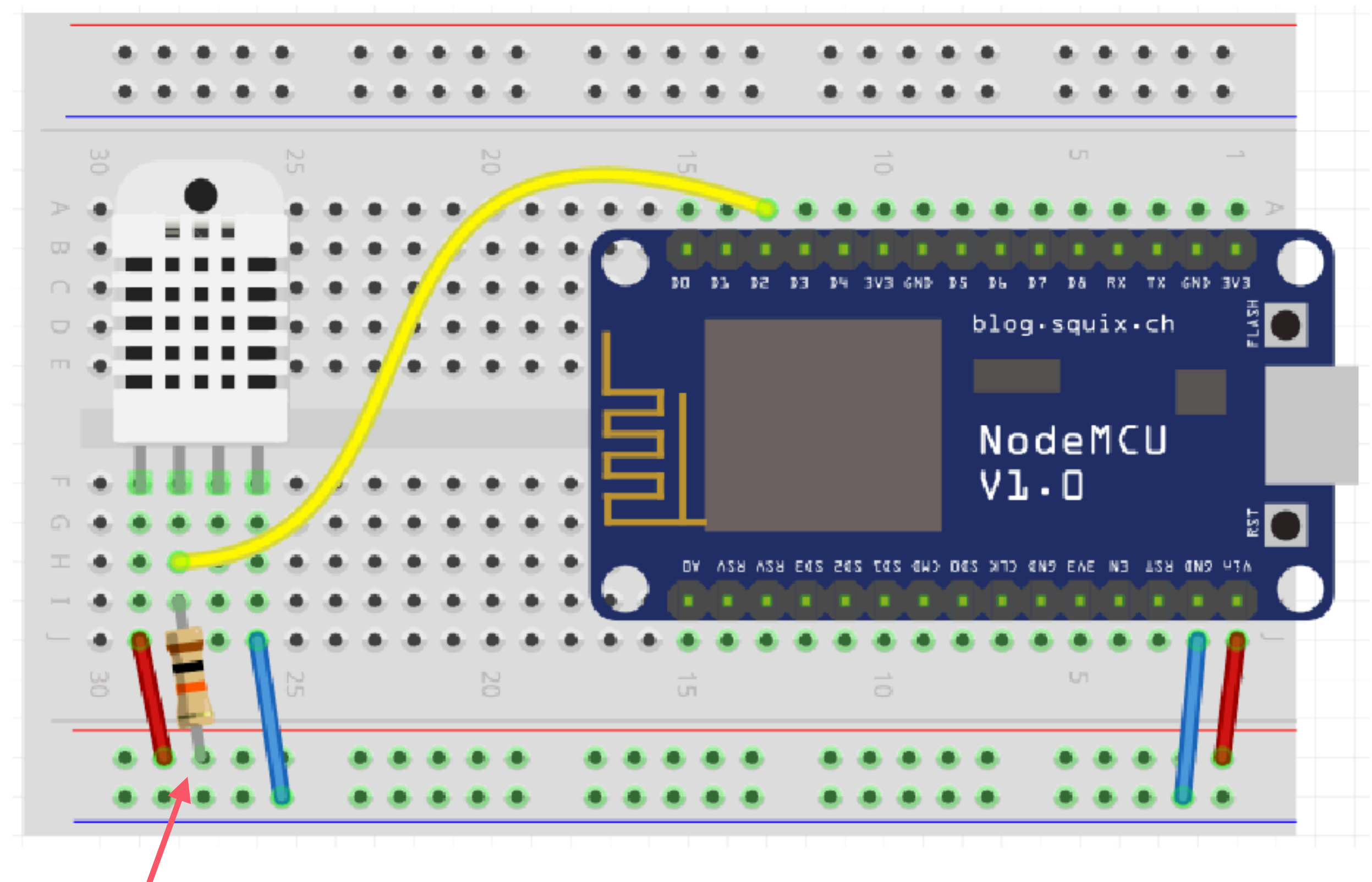


5 band resistor (brown, black, black, red)

## DHT22 Temperature and Humidity

### 3. The sketch (2\_DHT22.ino)

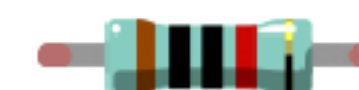
```
void sendSensorData() {  
  float theTemperature = dht.readTemperature();  
  float theHumidity = dht.readHumidity();  
  if (isnan(theHumidity) || isnan(theTemperature)) {  
    Serial.println("Failed to read from DHT sensor!");  
    return;  
  }  
  
  Serial.print(theTemperature);  
  Serial.print(" °C\t");  
  Serial.print(theHumidity);  
  Serial.println(" %");  
}
```



10 kΩ resistor



4 band resistor (brown, black, orange) or



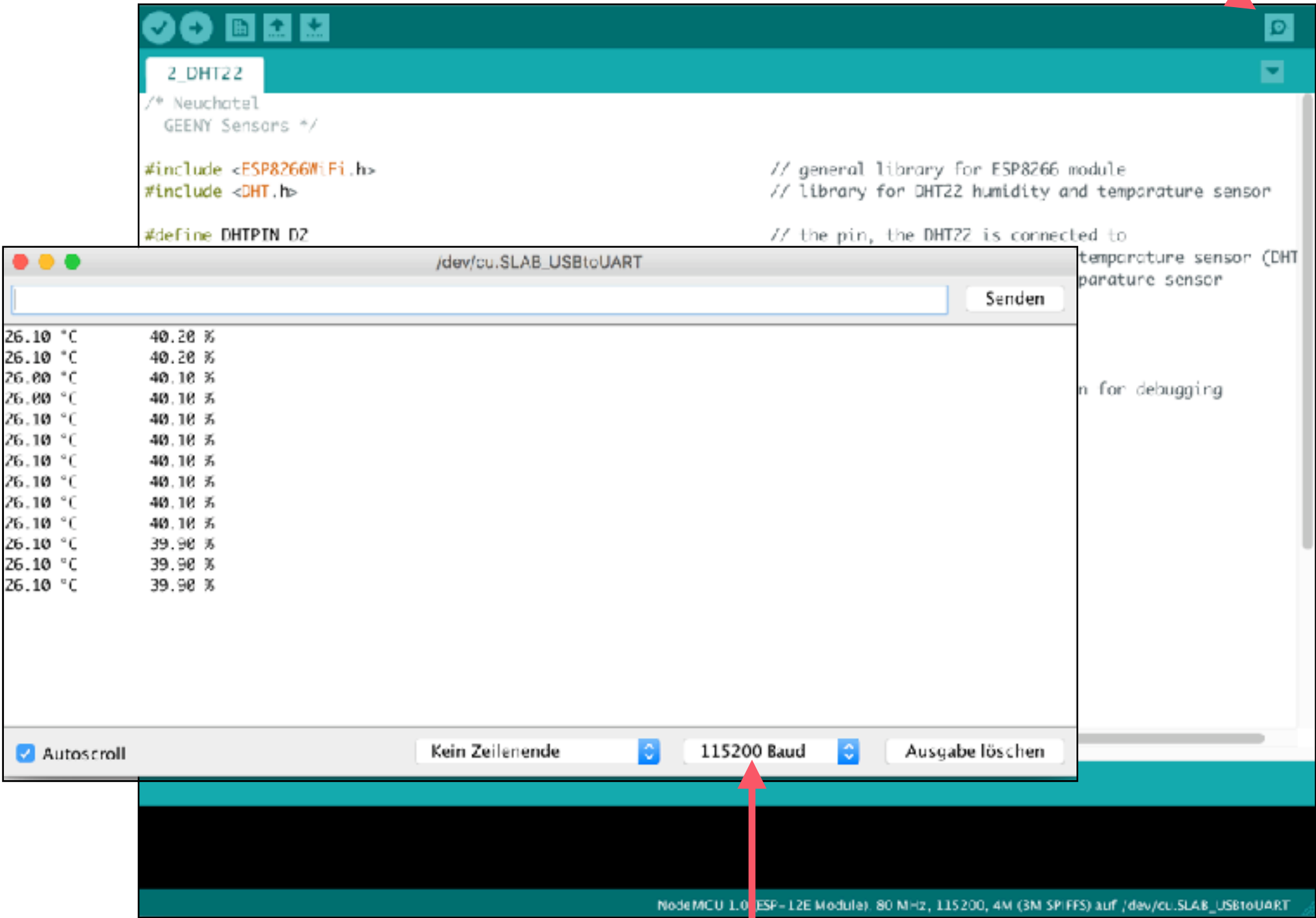
5 band resistor (brown, black, black, red)



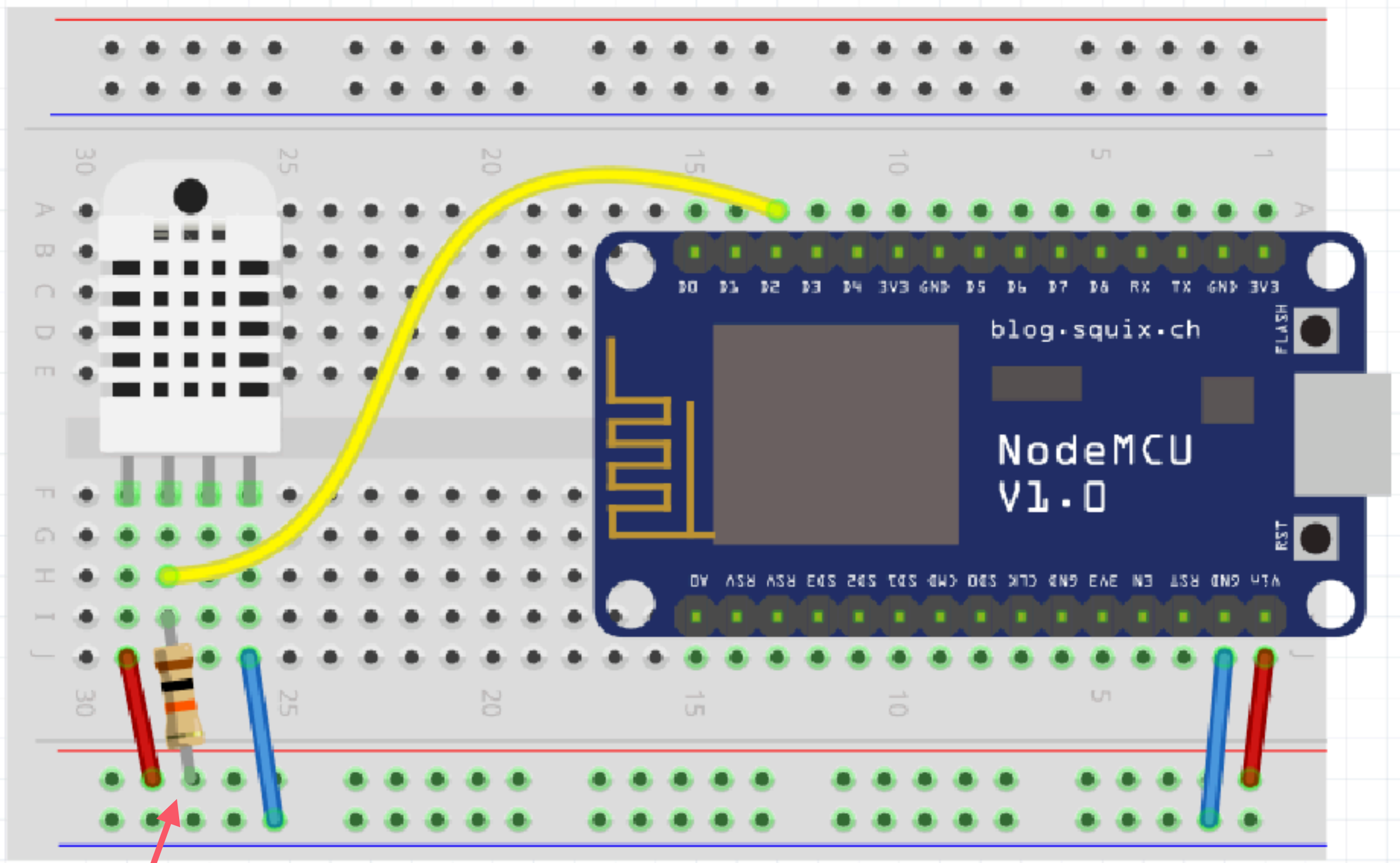
# DHT22 Temperature and Humidity

## 3. The sketch (2\_DHT22.ino)

Click to open SerialMonitor

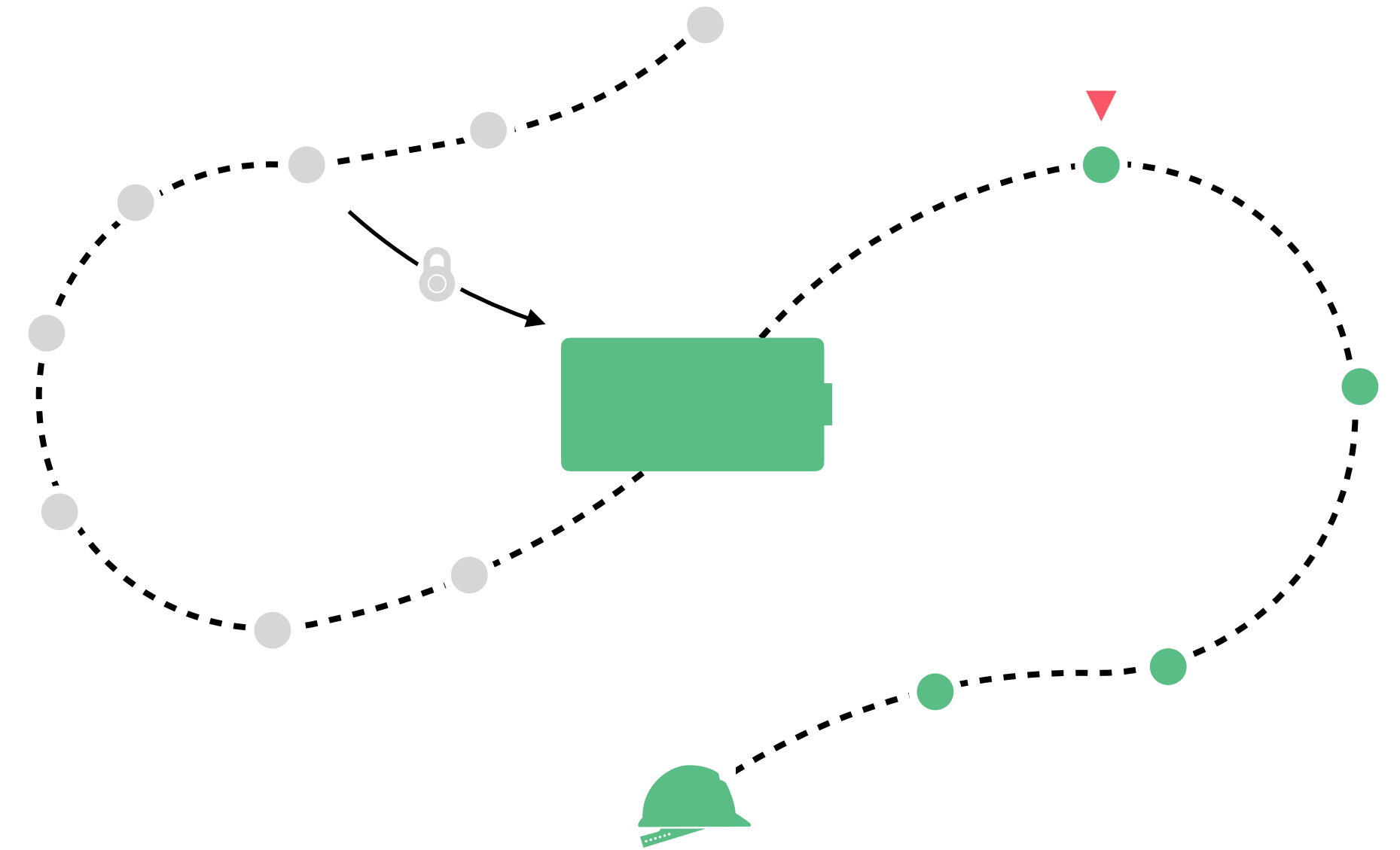


Make sure to set the baud rate to 115200



- 10 kΩ resistor
- 4 band resistor (brown, black, orange) or
- 5 band resistor (brown, black, black, red)

## Connect to Wifi



## DHT22 Temperature and Humidity

### 3. The sketch (3\_DHT22Wifi.ino)

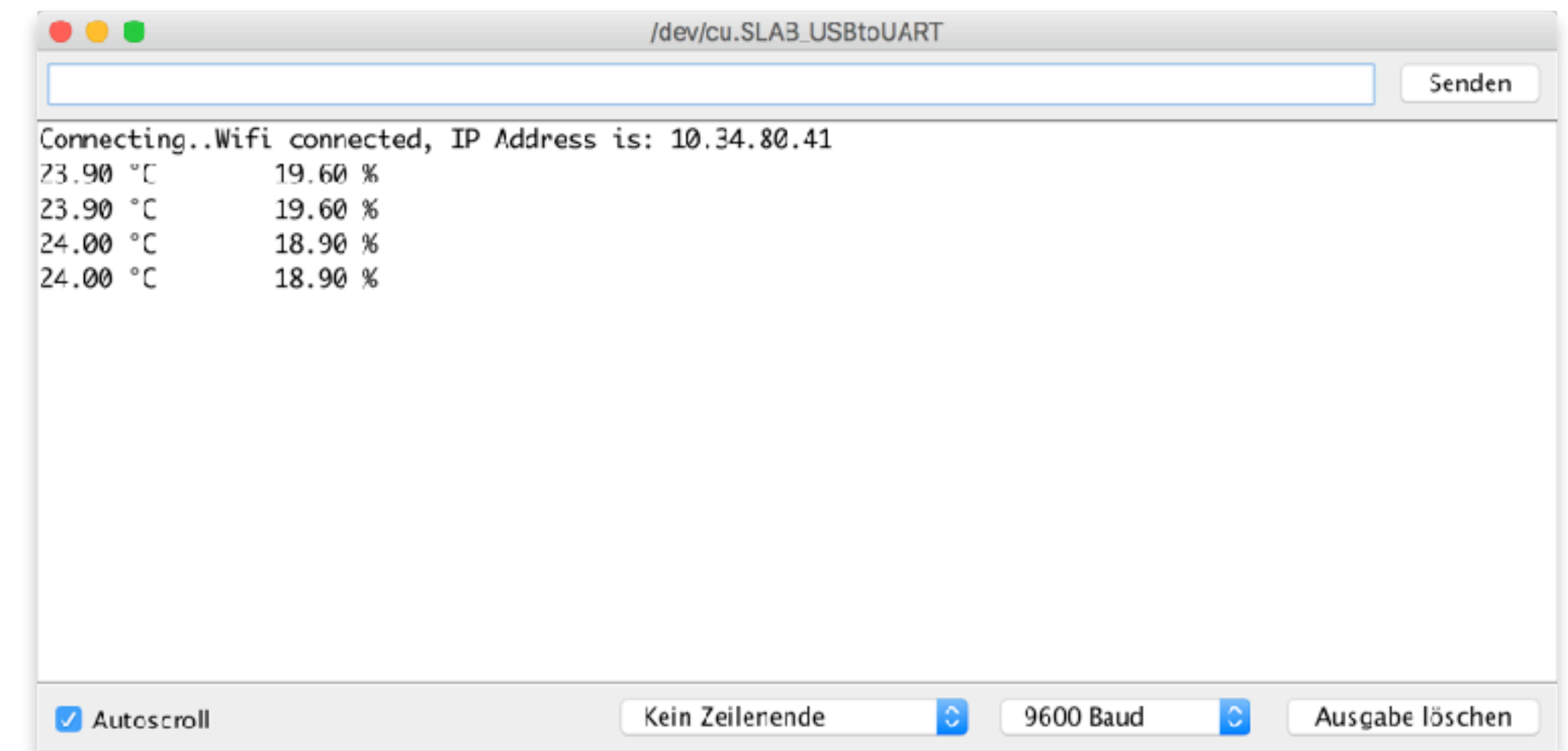
1. Change the wifi settings and upload the sketch

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

#define DHTPIN D2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
const char* ssid      = "#####";
const char* password = "#####";

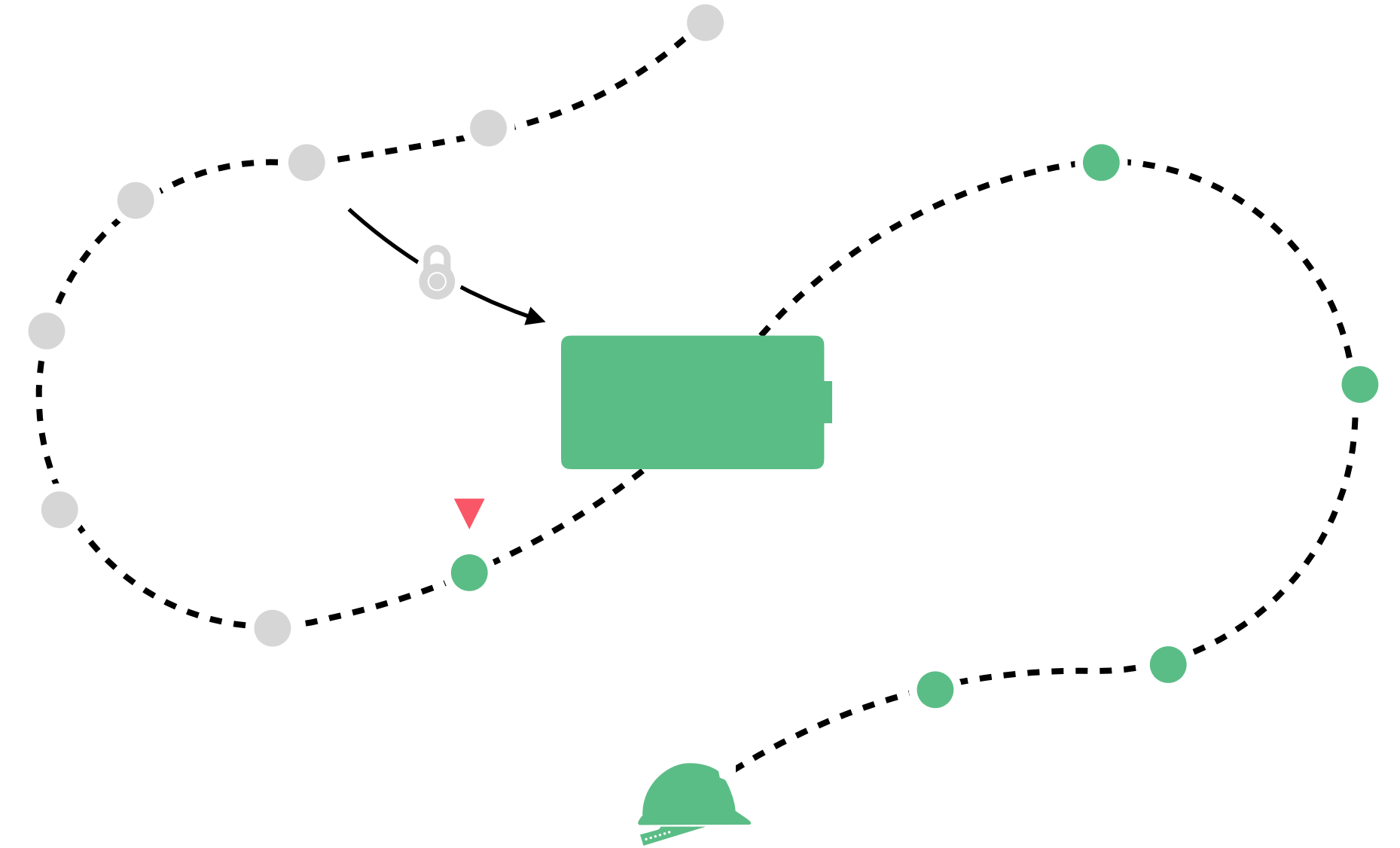
WiFiClient net;
unsigned long lastMillis = 0;
```

2. Open the Serial Monitor
3. Press the Reset button on the NodeMCU



Reset button

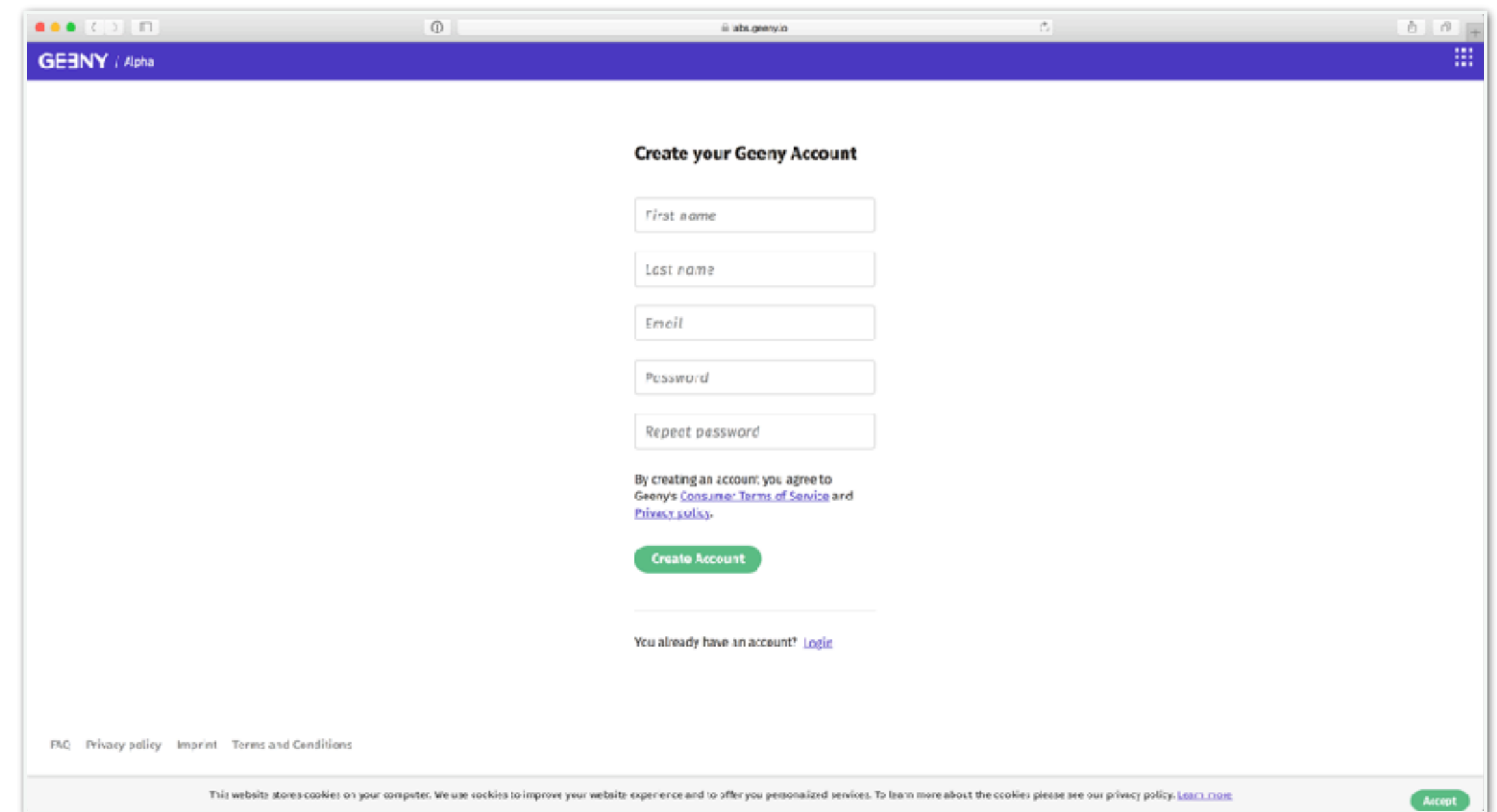
## Create Geeny Account



## Create Geeny Account

### 1. Create a Geeny account

1. Go to Geeny.io and create a new account
2. Click on *Become a developer*

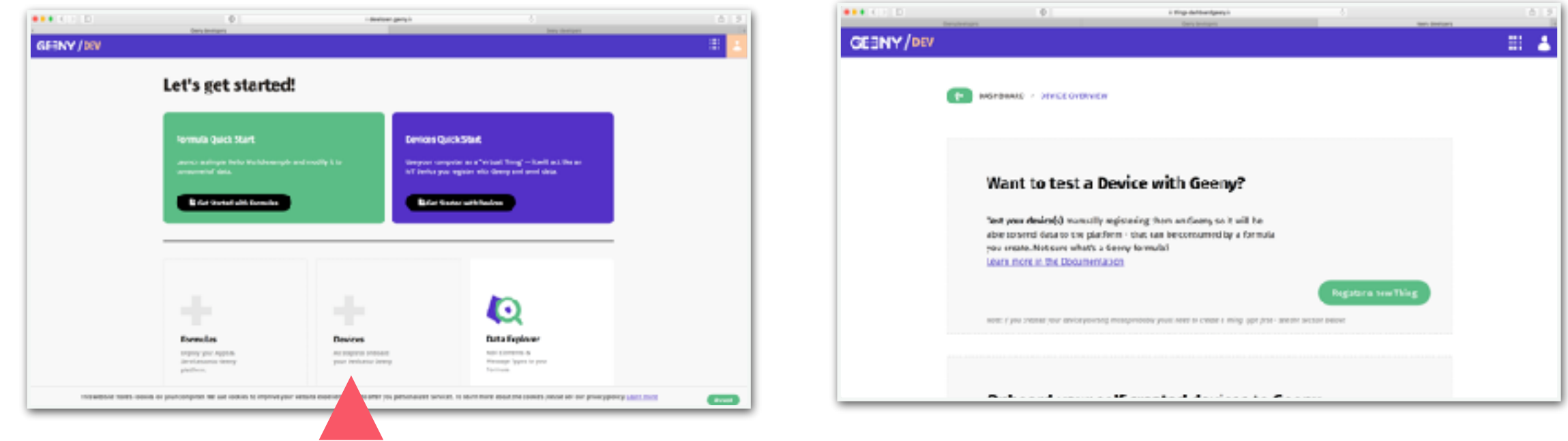


The screenshot shows a web browser window with the URL `http://alpha.geeny.io`. The page has a blue header with the "GEENY / Alpha" logo. The main content area is titled "Create your Geeny Account" and contains a registration form with the following fields: "First name", "Last name", "Email", "Password", and "Repeat password". Below the form, there is a text block stating: "By creating an account, you agree to Geeny's [Consumer Terms of Service](#) and [Privacy Policy](#)." A green "Create Account" button is positioned below this text. At the bottom of the form section, there is a link: "You already have an account? [Login](#)". The footer of the page includes links for "FAQ", "Privacy policy", "Imprint", and "Terms and Conditions". A cookie consent banner at the very bottom states: "This website stores cookies on your computer. We use cookies to improve your website experience and to offer you personalized services. To learn more about the cookies please see our privacy policy: [LINK TO OUR](#)" with an "Accept" button.

## Create Geeny Account

### 2. Create a new *Thing*

1. Click on *Devices*
2. Click on *Register a new Thing*
3. fill out the form:
  1. give it a name you like,
  2. serial number  
(a new UUID e.g. from [www.uuidgenerator.net](http://www.uuidgenerator.net))
  3. and select the Thing Type *NodeMCU+DHT22\_1*
4. Click on *Register this Thing*

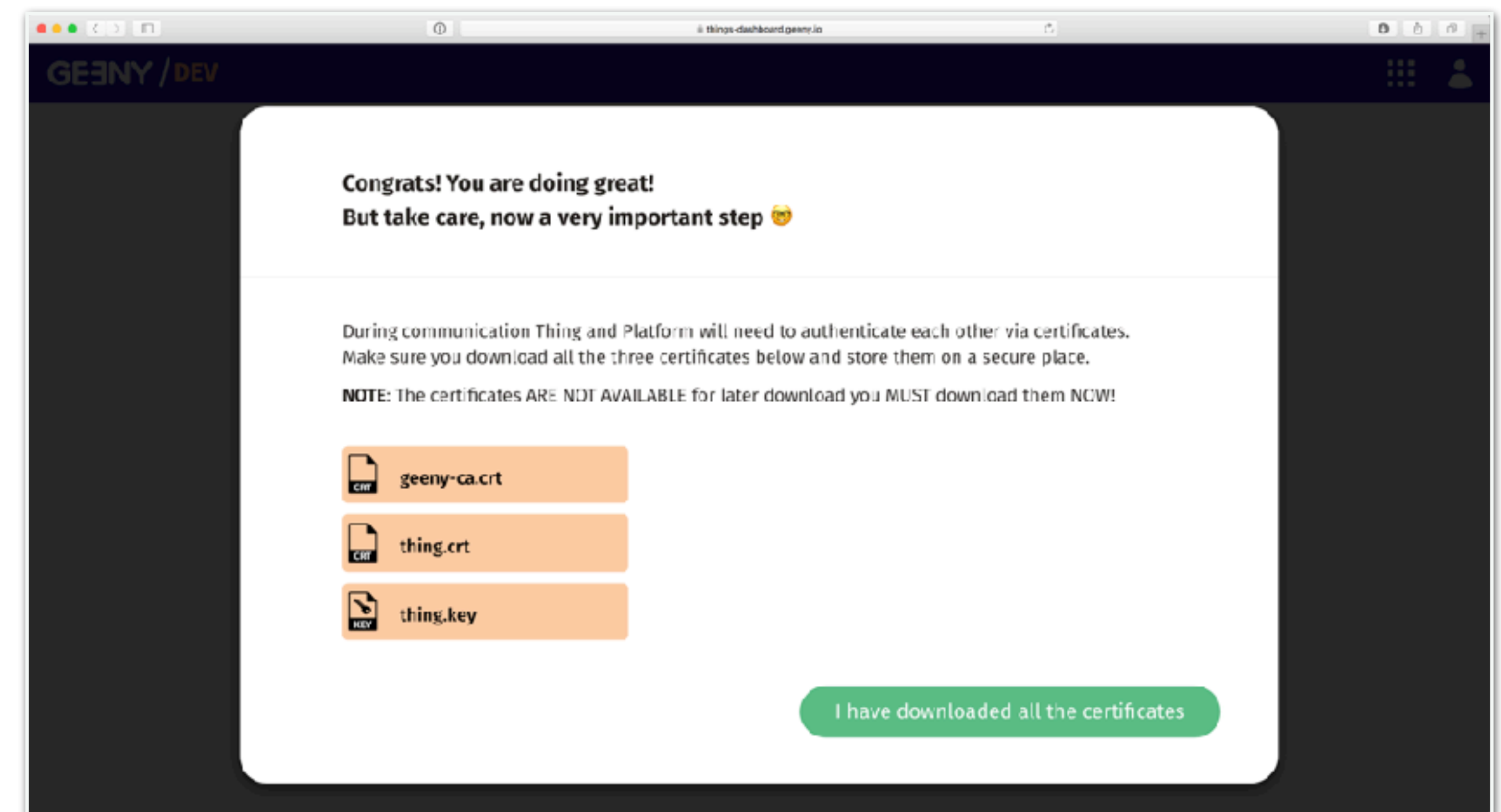
This screenshot shows the 'Register a new instance of NodeMCU+DHT22\_1' form. At the top, there's a breadcrumb 'DEVICE OVERVIEW > CREATE THING'. Below it, a message box says 'You are about to register on Geeny a specific instance of NodeMCU+DHT22\_1'. The form is titled 'Please describe your Thing' and contains three input fields: 'NAME (i)' with the value 'MyNodeMCU', 'SERIAL NUMBER (i)' with the value '3b403993-9409-445f-9fcb-f9eb', and 'THING TYPE (i)' with a dropdown menu showing 'NodeMCU+DHT22\_1'. A green 'Register this Thing' button is at the bottom right.



## Create Geeny Account

### 3. Download the certificates\*

1. Now a screen opens asking to download the certificate files. Save the files on your computer by clicking each file.



*\*Certificates are used to authenticate the participants in a secure communication process. In this example it authenticates the NodeMCU and Geeny.*

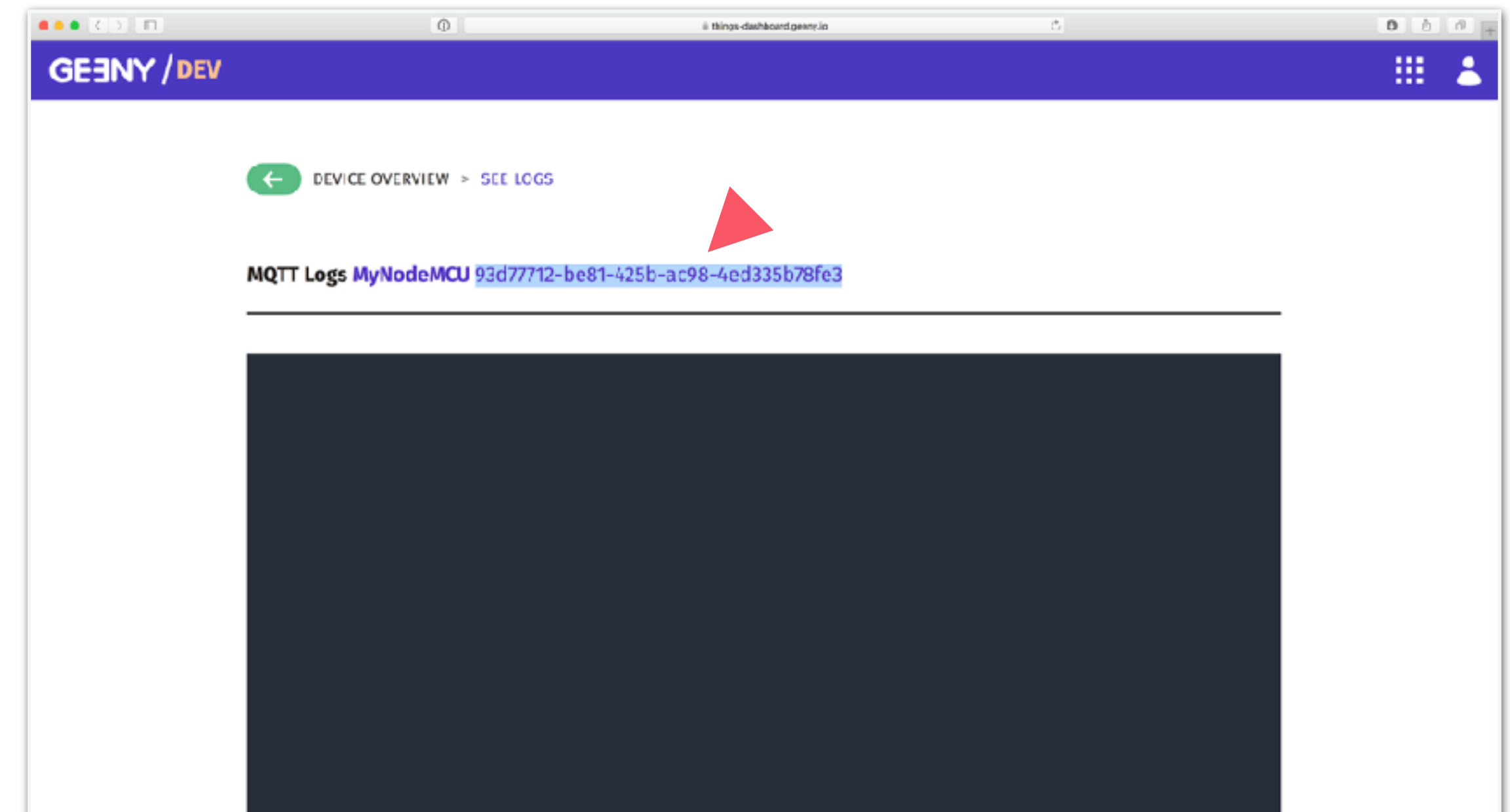
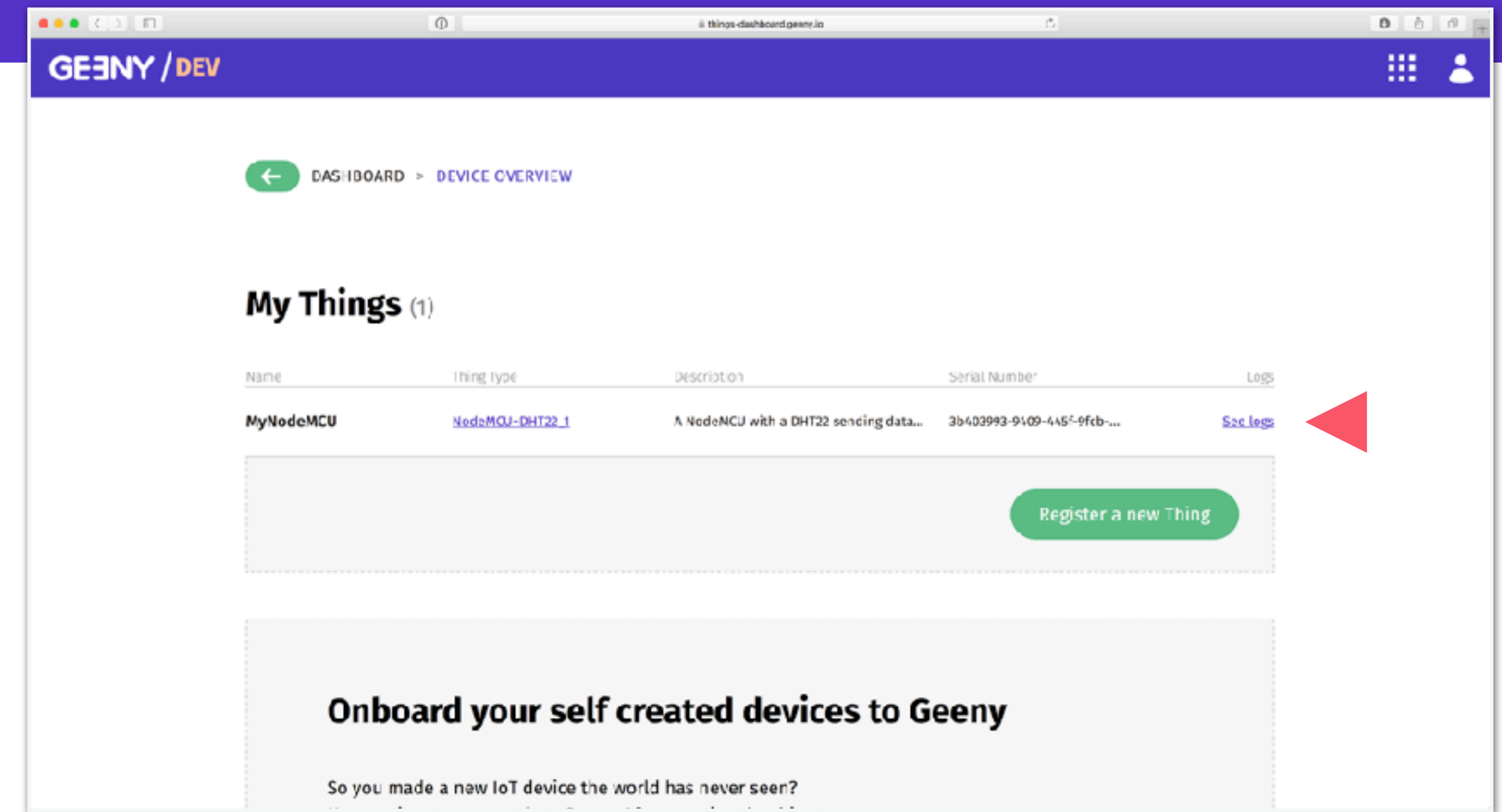


## Create Geeny Account

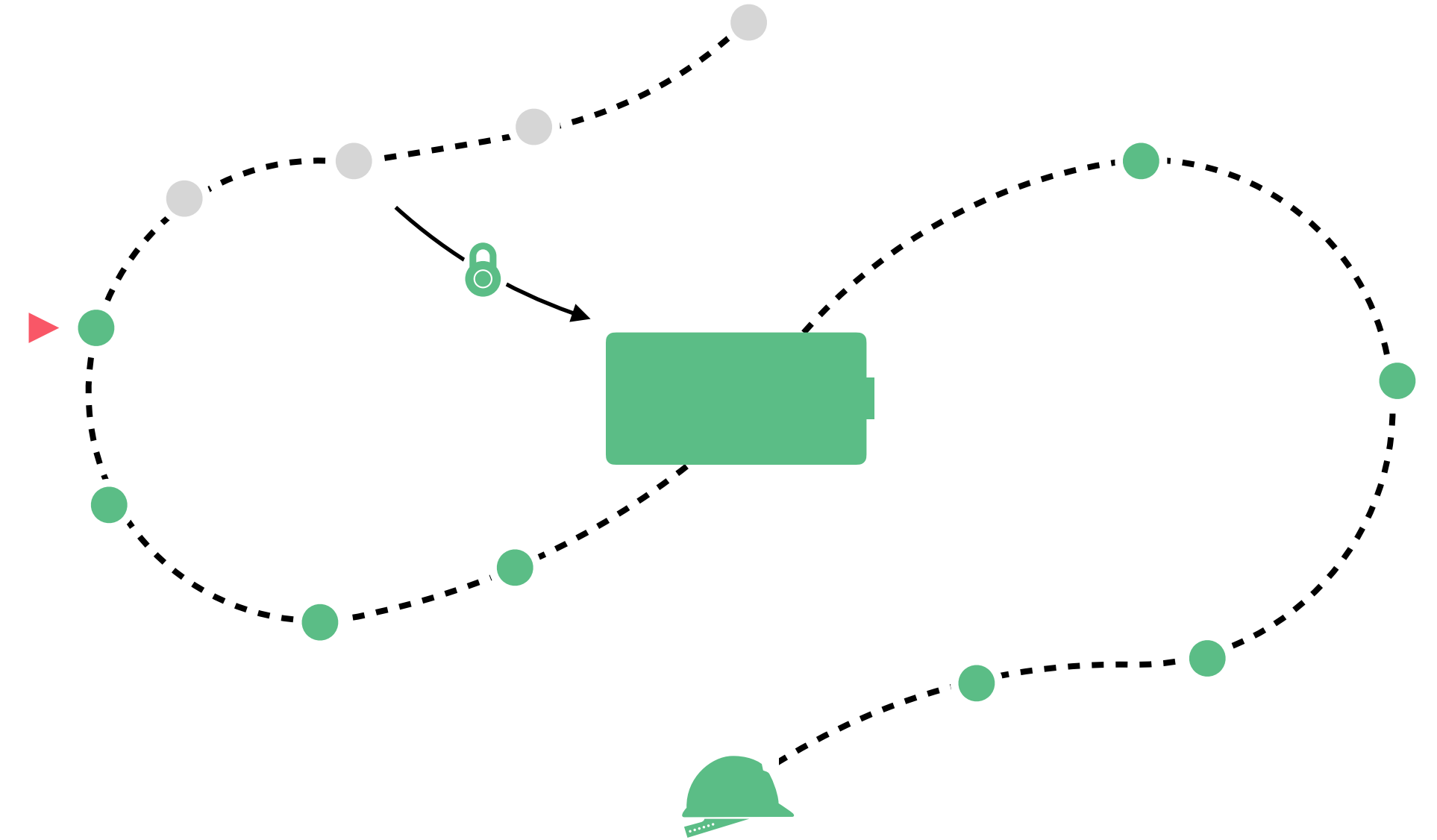
### 4. Get your *Thing-ID*

1. On the *Device overview* page you see your newly created Thing. Click on *See logs*
2. Copy the *Thing-ID* shown on top of the page and save it somewhere

BTW: this is where we will see the data later on



## Upload Certificates



## Upload Certificates

### 1. Generate DER files

#### OSX

1. Open a Terminal window and navigate to the folder where you stores the certificate files
2. Enter the two following commands

```
openssl rsa -outform der -in thing.key -out thing-key.der
openssl x509 -outform der -in thing.crt -out thing-crt.der
```
3. Now you should see two new files in your folder with the file extension .der
4. Copy this files into the *data* folder of your Sketch (*4\_DHT22Geeny*)

#### Windows

1. Download and install OpenSSL from [slproweb.com/products/Win32OpenSSL.html](http://slproweb.com/products/Win32OpenSSL.html) (File: *Win32 OpenSSL v1.1.0g Light*)
2. Copy the Certificates to *C:\Geeny*
3. Open a Command Window (Windows+R and enter CMD)
4. Go to *C:\OpenSSL-Win32\bin* and run *openssl.exe*
5. Enter the two following commands

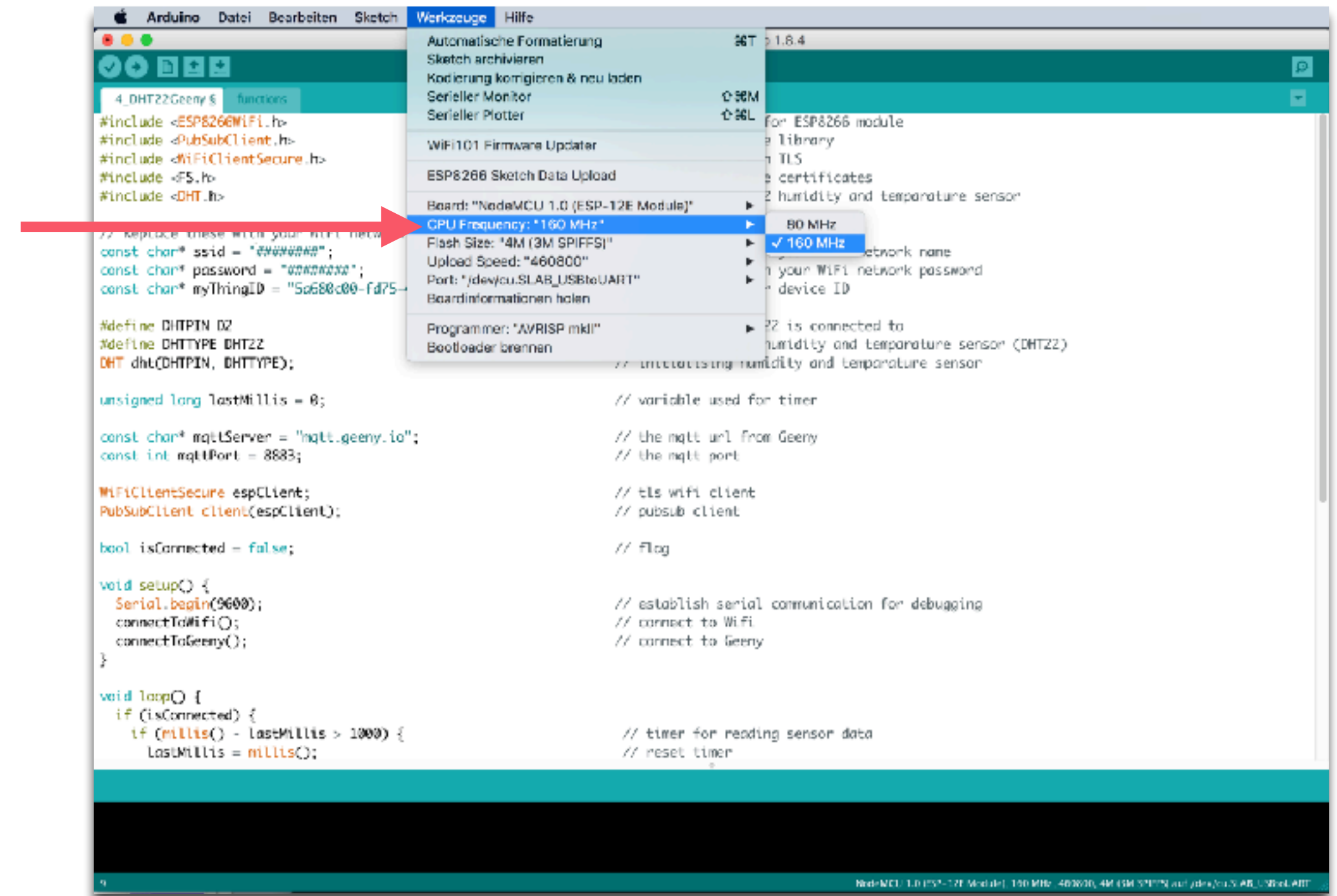
```
openssl rsa -outform der -in c:\Geeny\thing.key
-out c:\Geeny\thing-key.der

openssl x509 -outform der -in c:\Geeny\thing.crt
-out c:\Geeny\thing-crt.der
```
6. Now you should see two new files with the file extension .der in this folder
7. Copy this files into the *data* folder of your Sketch (*4\_DHT22Geeny*)

## Upload Certificates

### 2. Upload DER files to NodeMCU

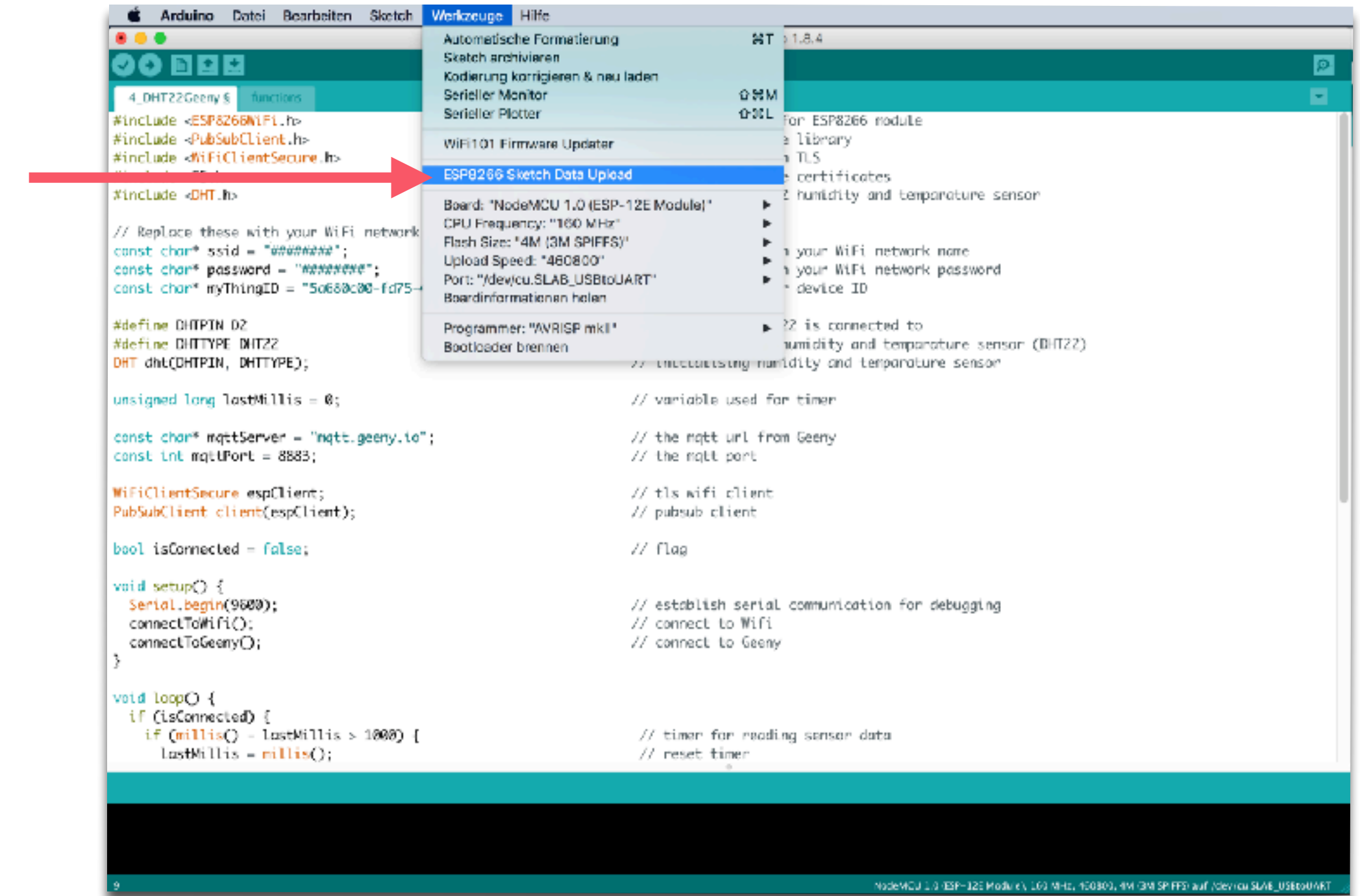
1. In order to upload the certificates to the NodeMCE, you need a plugin for the Arduino IDE. Download the plugin from: <https://github.com/esp8266/arduino-esp8266fs-plugin/releases/download/0.2.0/ESP8266FS-0.2.0.zip>
2. In your Arduino sketchbook directory, create tools directory if it doesn't exist yet
3. Unpack the tool into tools directory  
(../Arduino/tools/ESP8266FS/tool/esp8266fs.jar)
4. Restart Arduino IDE
5. In the *Tools* menu set the CPU frequency to 160 MHz



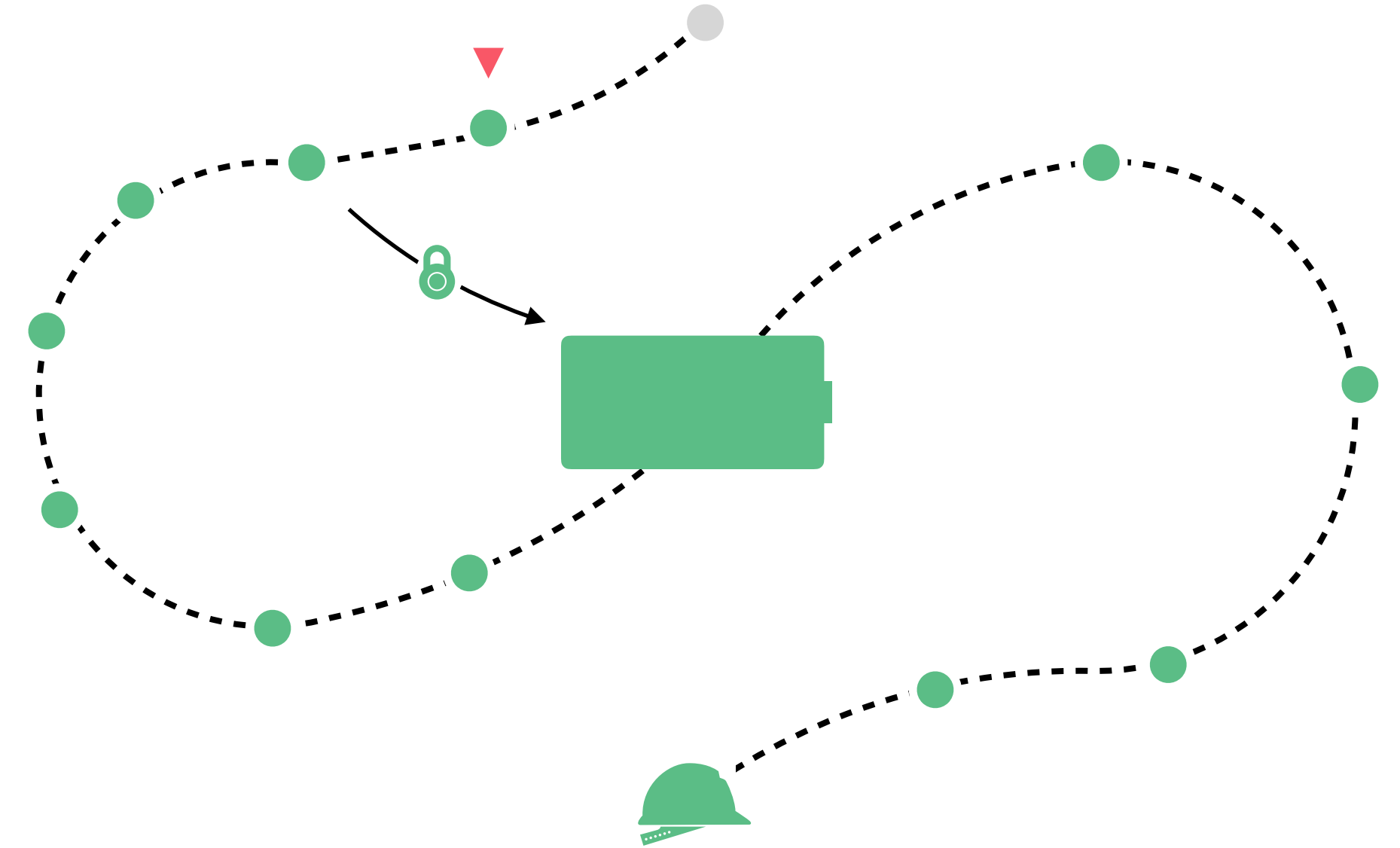
## Upload Certificates

### 2. Upload DER files to NodeMCU

1. Click on *Tools>ESP8266 Sketch Data Upload*



## The Final Sketch



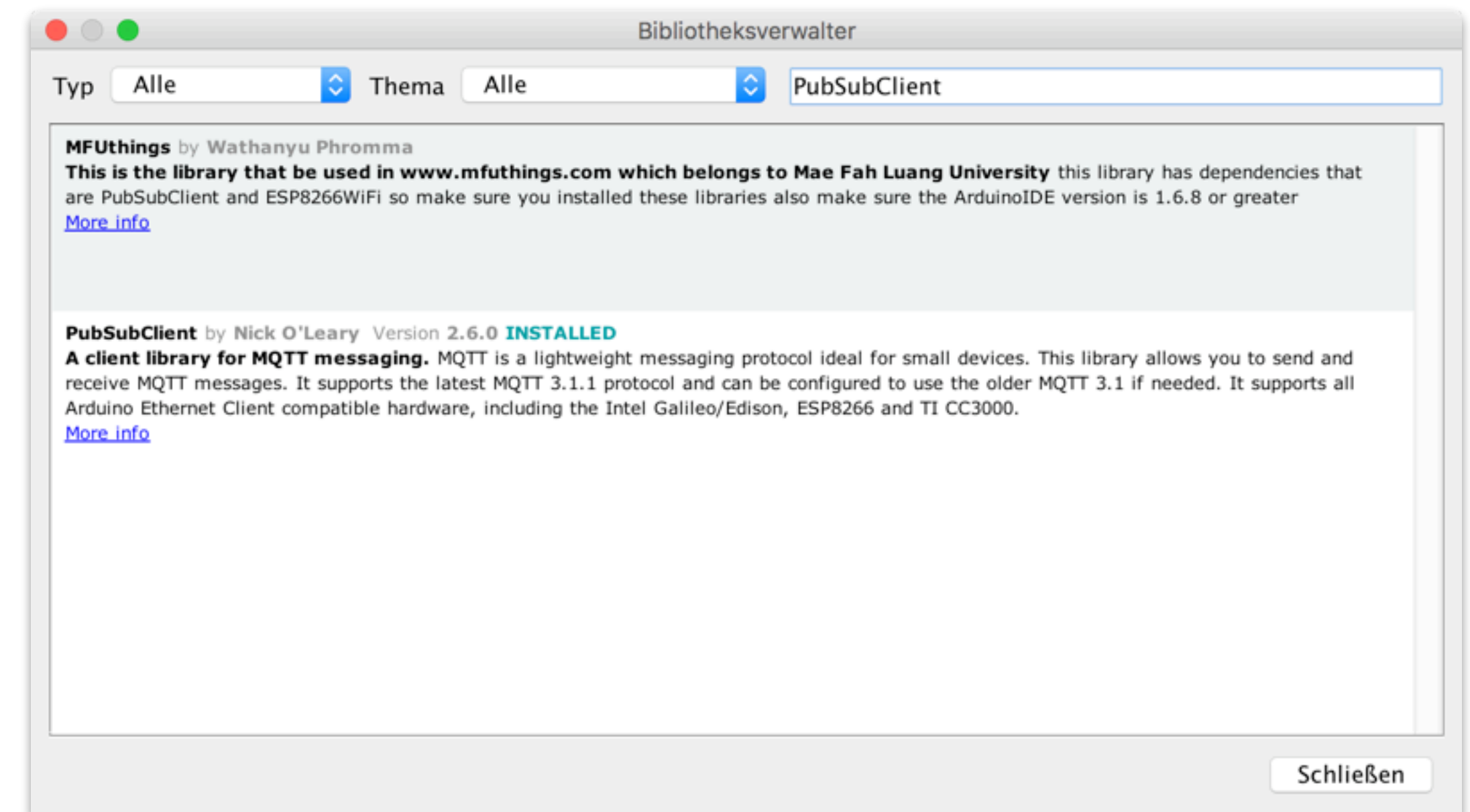




## Final Sketch

### 1. Install the MQTT Library

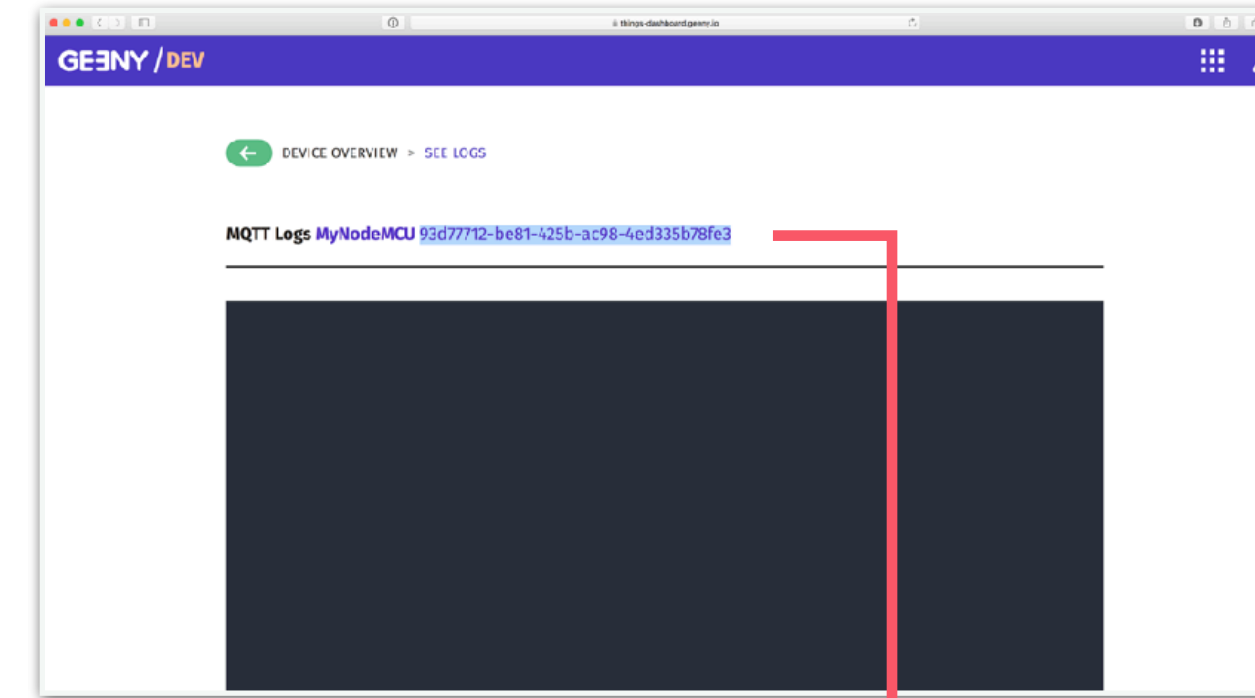
1. In the Arduino IDE go to *Sketch>Include Library>Manage libraries ...* and search for **PubSubClient**
2. Install the **PubSubClient** by Nick O'Leary library



## Final Sketch

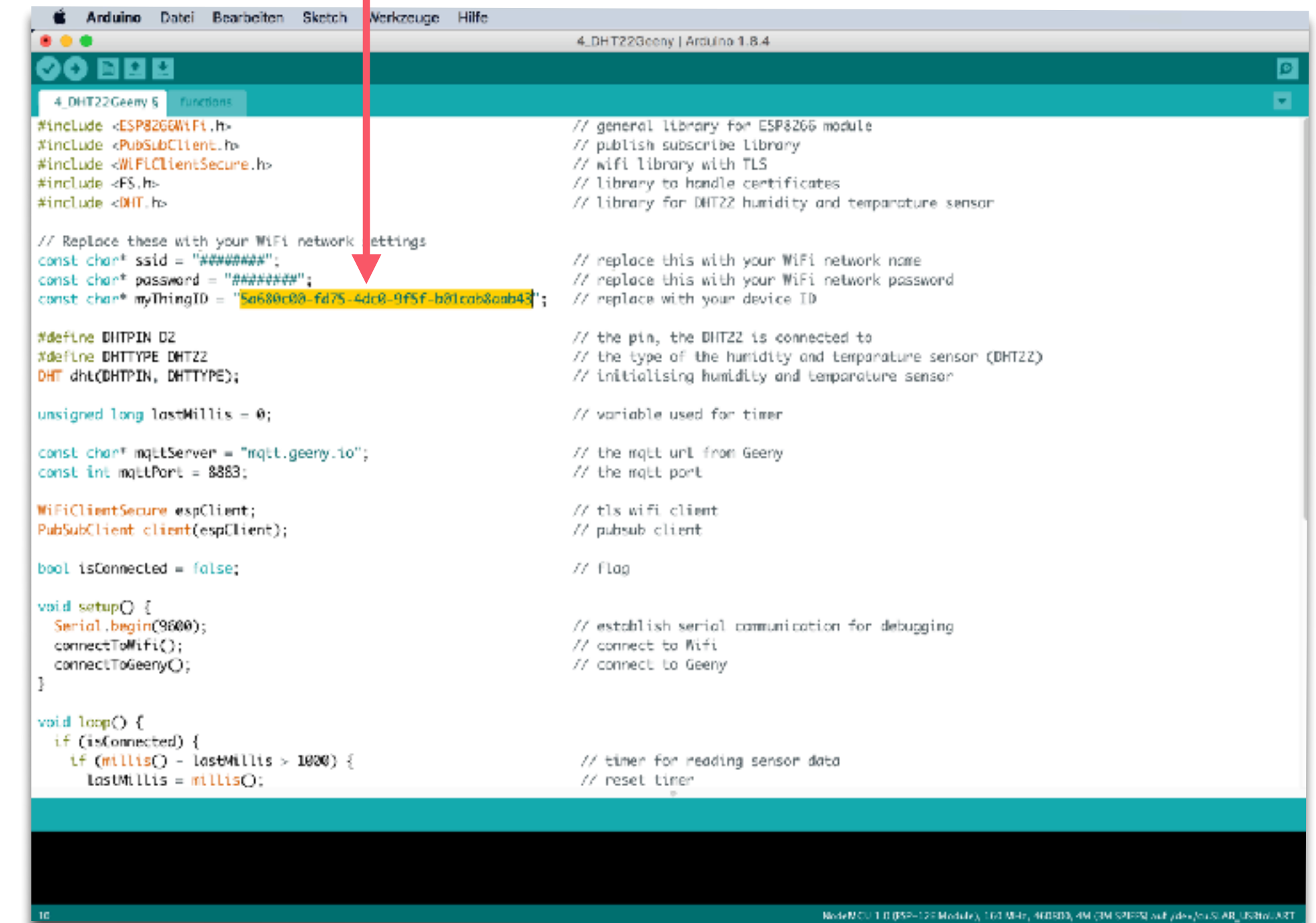
### 2. Upload sketch to NodeMCU

1. Open the Sketch *4\_DHT22Geeny*
2. Change your Wifi settings
3. Change myThingID to the one you copied from the *Device logs* page
4. Upload the sketch to your NodeMCU
5. You should now see data on the Device log page!



Device logs page

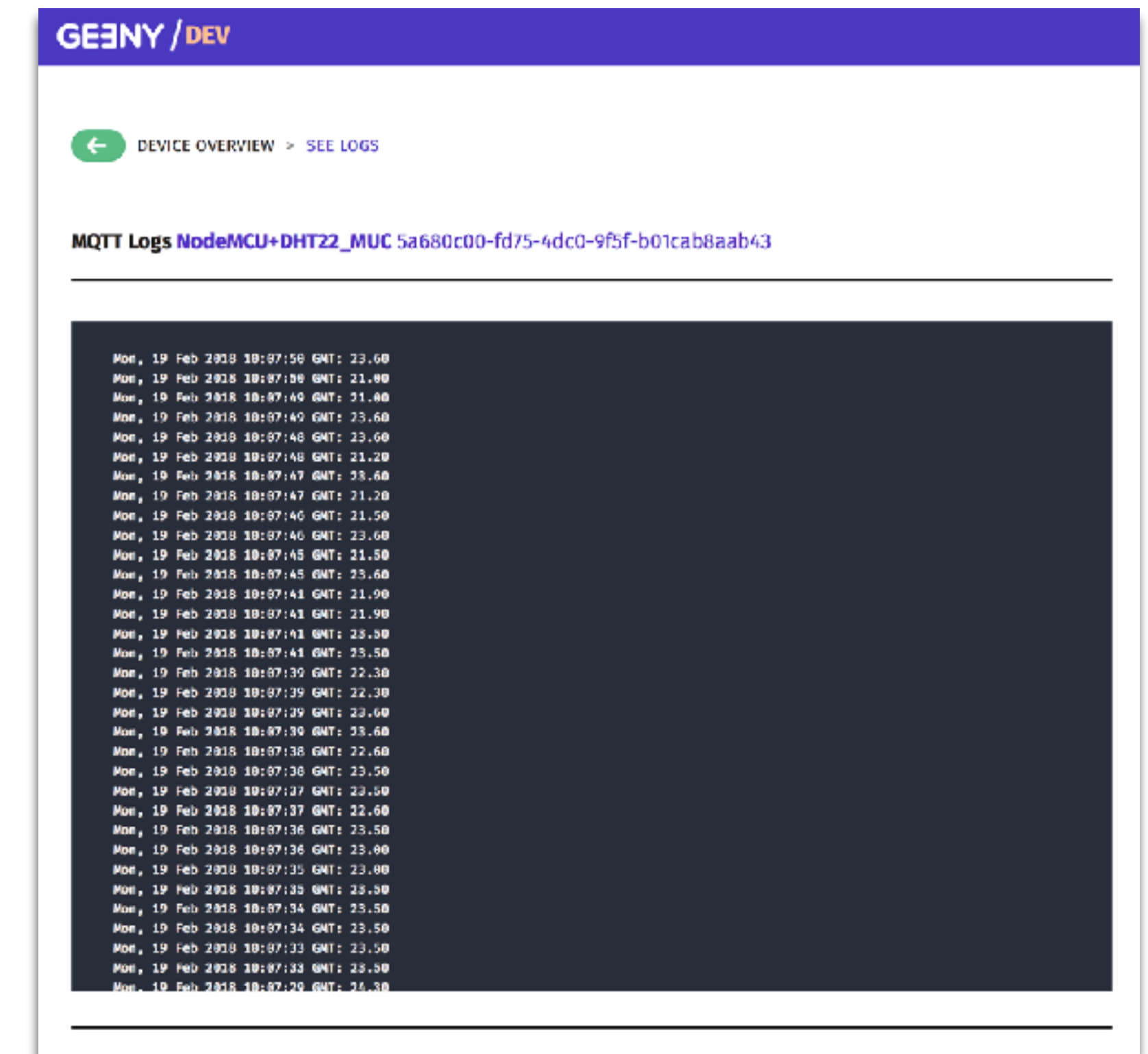
Things ID



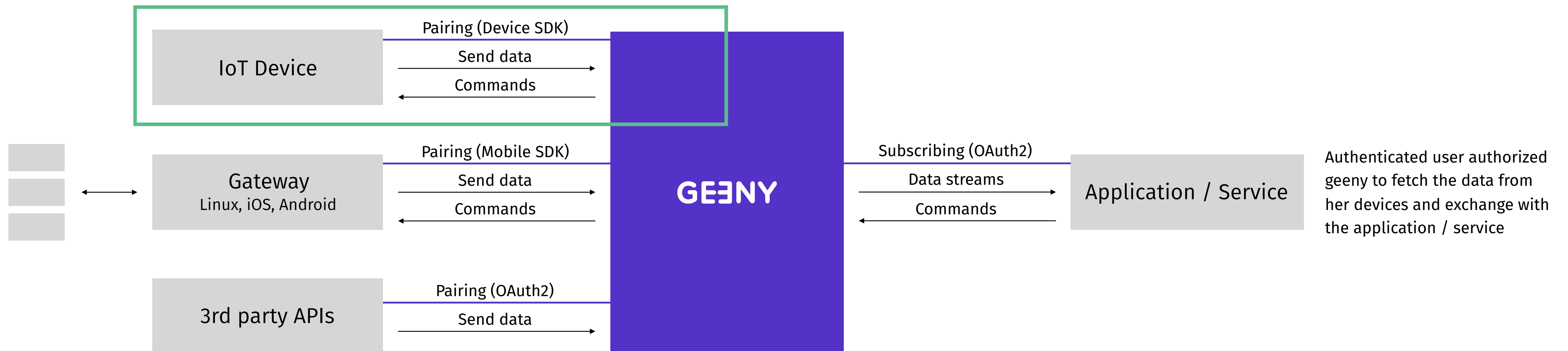
## Final Sketch

### 2. Upload sketch to NodeMCU

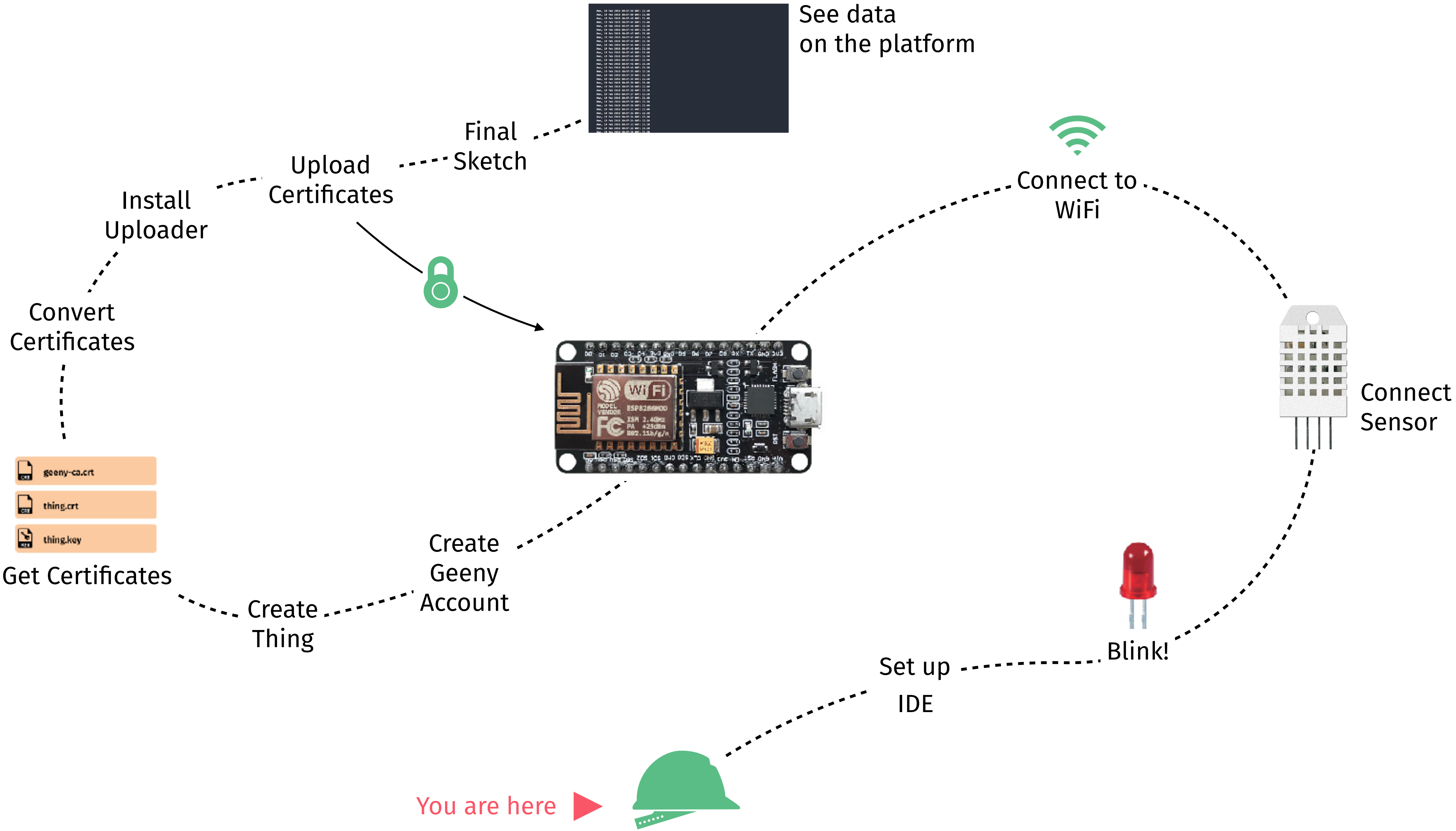
1. Open the Sketch [4\\_DHT22Geeny](#)
2. Change your Wifi settings
3. Change myThingID to the one you copied from the *Device logs* page
4. Upload the sketch to your NodeMCU
5. You should now see data on the Device log page!



# Outlook



# The Journey





## Outlook

### Check our Documentation Center

*docs.geeny.io*

### Let's stay in touch!

1. Geeny.io  
@geeny.io
2. Stefan Hermann  
stefan.h@geeny.io  
@stefanhermann
3. Joelle Katz  
joelle@geeny.io  
@joellekatz

