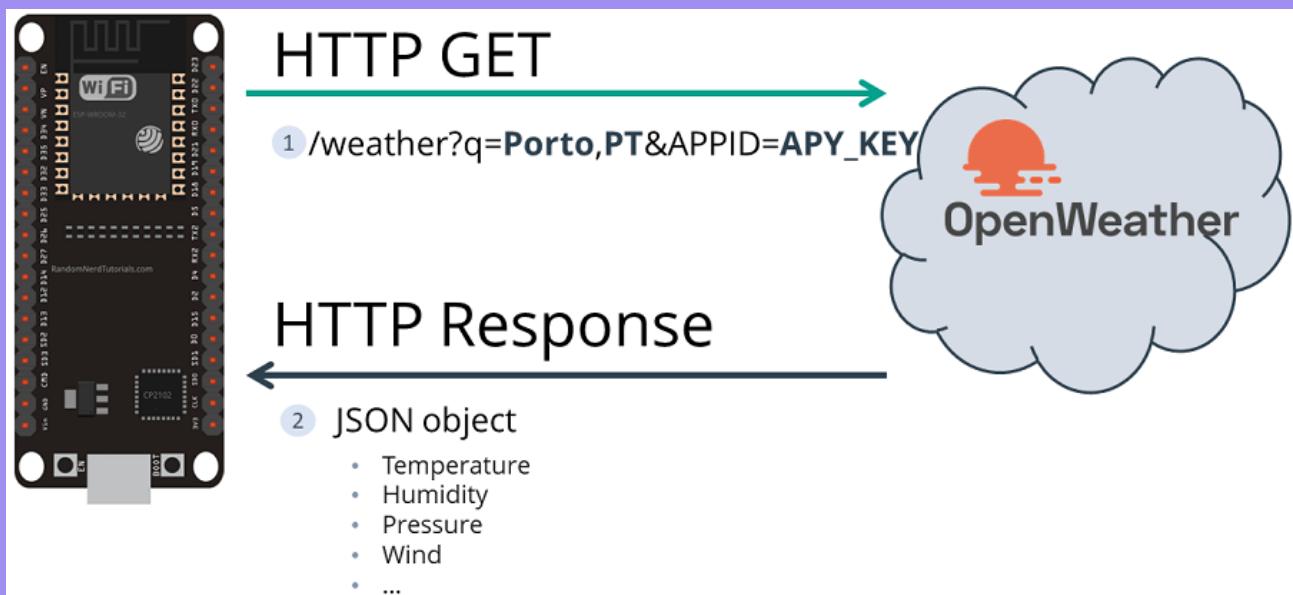


# Technical Manual

## API & Wearable



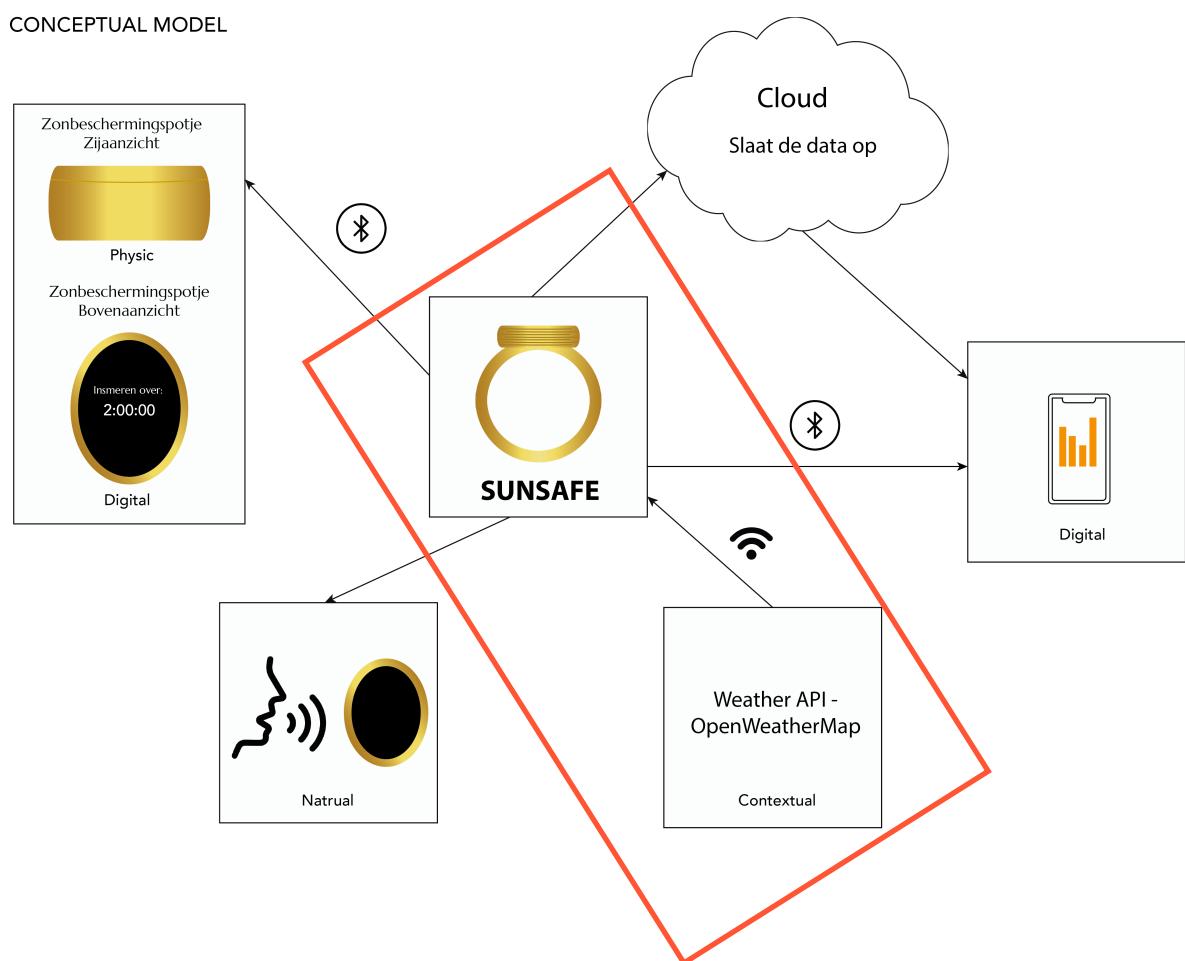
# Inhoudsopgave

|                                       |    |
|---------------------------------------|----|
| 1.1 Idea Technical Manual             | 3  |
| 1.2 Preparing for my technical manual | 4  |
| 1.3 Technical Manual                  | 5  |
| Sources                               | 12 |

## 1.1 Idea Technical Manual

In class I ran into the problem that the conceptual model was not very clear for me. After the lesson I went to David to ask where I could focus on for the technical manual. I said that I found the API interesting. That's why I eventually chose to focus on the API & the ring.

My idea is that I can measure the UV index on the basis of the API and translate this into different colors on my LED strip based on a time.

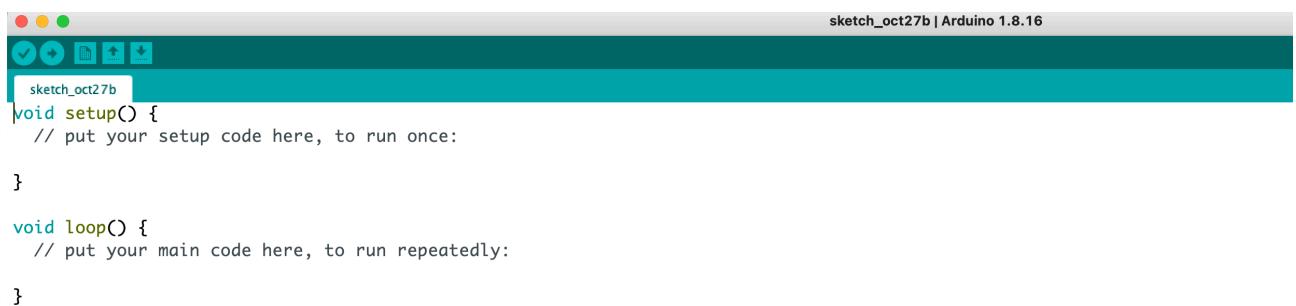


## 1.2 Preparing for my technical manual

First I was looking on the internet for a API that could tell the UV-index on different locations over the whole world. I found OpenWeather.



I will also use Arduino for the API and my led strip.



The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_oct27b | Arduino 1.8.16". The main window displays a blank sketch with the following code:

```
sketch_oct27b
void setup() {
  // put your setup code here, to run once:
}

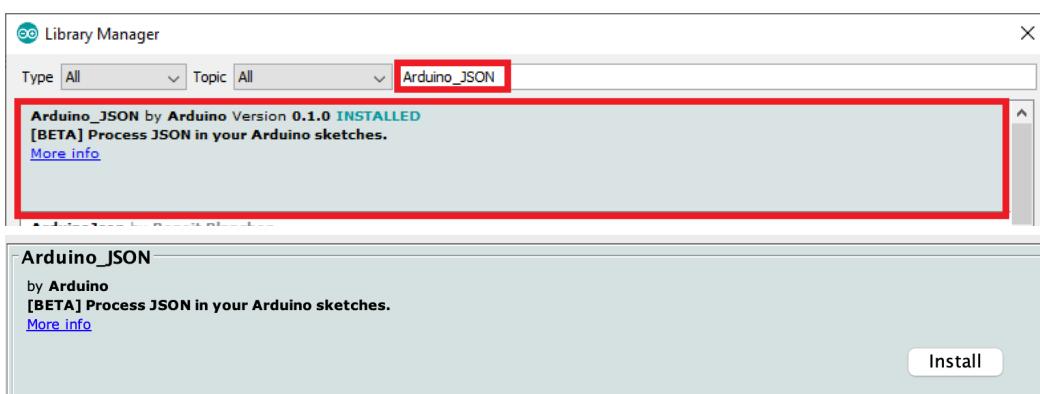
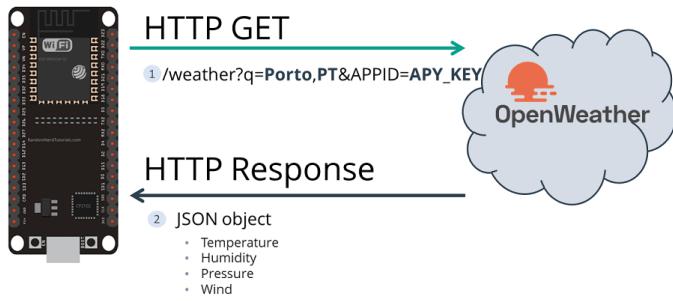
void loop() {
  // put your main code here, to run repeatedly:
}
```

Finally I need an LED strip and node mcu.



## 1.3 Technical Manual

The first step is to install ARDUINO\_JSON in the library in ARDUINO.



Then I started OpenWeather.

A screenshot of the OpenWeather API documentation for the "Ultraviolet Index (Deprecated)" endpoint. The top navigation bar includes links for Guide, API, Pricing, Maps, Our Initiatives, Partners, Blog, Marketplace, Sign in, and Support. The "Sign in" button is highlighted with a red box. The main content area features a heading "Ultraviolet Index (Deprecated)". A note states: "The product retired on 1st April 2021, please find UV data in One Call API. One Call API includes current, hourly forecast for 7 days and 5 days historical UV data." Below this, a paragraph says: "Access current, forecast and historical UV data for any location on Earth including over 200,000 cities! Data is available in the JSON format." To the right, a sidebar lists related endpoints: "Call current UV data By geographic coordinates", "Forecast UV data By geographic coordinates", "Historical UV data for By geographic coordinates", "Server response format Codes of requests". At the bottom, a section titled "Call current UV data" provides "Parameters of API call" with fields for "lat, lon" (required Geographical coordinates) and "appid" (required Your unique API key). A note indicates that the API key can be found under the "API key" tab on the account page.

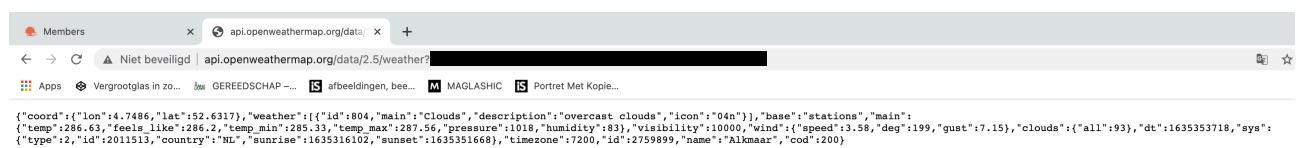
I go to Suze. I will click on the "My API Key".



Now I need to paste the Api openweathermap CURL in the webbrowser with my city name, country and UniqueAPIKey.

```
http://api.openweathermap.org/data/2.5/weather?  
q=yourCityName,yourCountryCode&APPID=yourUniqueAPIkey
```

I changed this in the name of my city, country and Unique APIKey. After that I pasted this code in the webbrowser.



I changed this in the name of my city, country and Unique APIKey. After that I pasted this code in the webbrowser.

After that I pasted the code from the site "Randomnerdtutorials" in mijn Arduino.

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <Arduino_JSON.h>

const char* ssid = "Rootring";
const char* password = "R00tring!";

// Your Domain name with URL path or IP address with path
String openWeatherMapApiKey = "e073e477d0d3b596987e37fd0530db53";
// Example:
//String openWeatherMapApiKey = "bd939aa3d23ff33d3c8f5dd1dd435";

// Replace with your country code and city
String city = "Alkmaar";
String countryCode = "NL";

// THE DEFAULT TIMER IS SET TO 10 SECONDS FOR TESTING PURPOSES
// For a final application, check the API call limits per hour/minute to avoid getting blocked/banned
unsigned long lastTime = 0;
// Timer set to 10 minutes (600000)
//unsigned long timerDelay = 600000;
// Set timer to 10 seconds (10000)
unsigned long timerDelay = 10000;

String jsonBuffer;

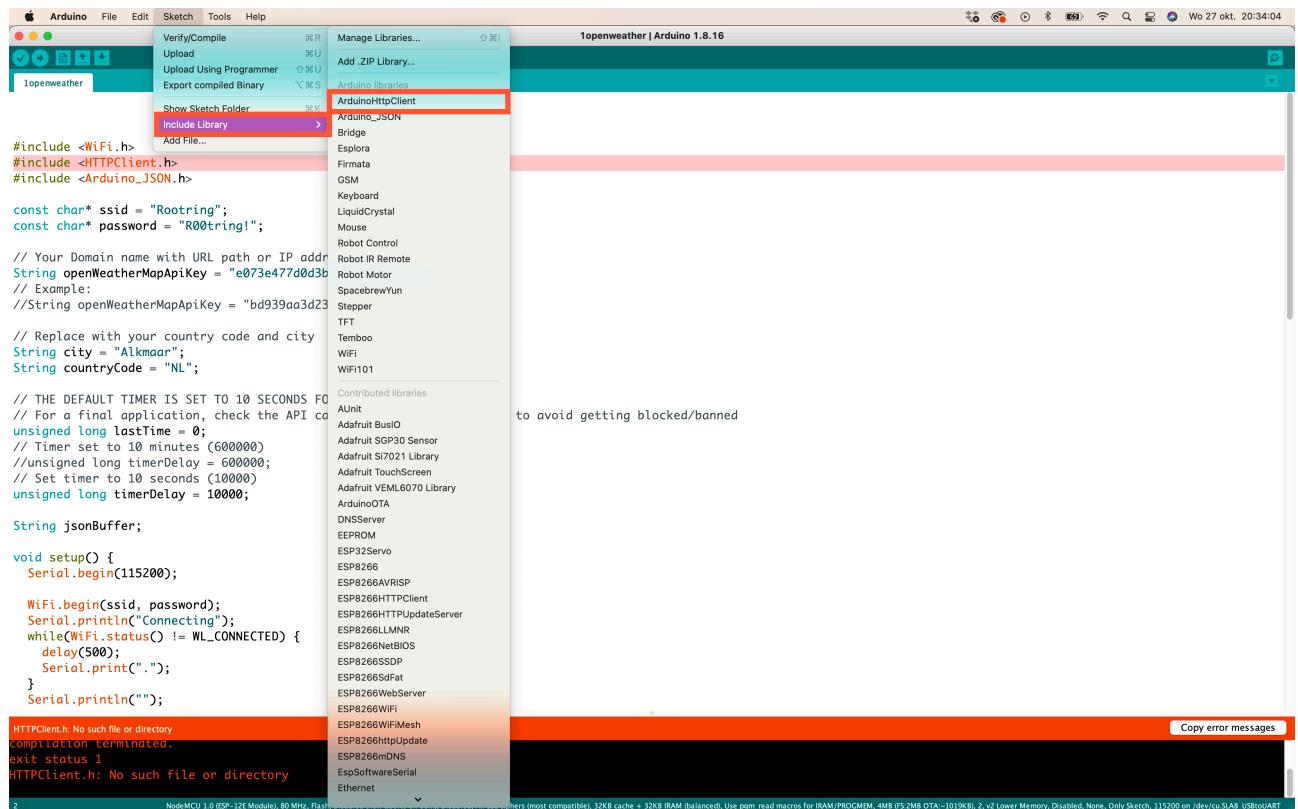
void setup() {
  Serial.begin(115200);

  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
}

HTTPClient.h: No such file or directory
Compilation terminated.
exit status 1
HTTPClient.h: No such file or directory
```

I got the error that there is something wrong with my `HTTPClient.h`. I have looked on the internet, but I did not find anything that could help me with this problem.

After 1 hour of searching I thought maybe there is something wrong with my library.



```

WiFi.begin(ssid, password);
Serial.println("Connecting");
while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.print("Connected to WiFi network with IP Address: ");
Serial.println(WiFi.localIP());

Serial.println("Timer set to 10 seconds (timerDelay variable), it will take 10 seconds before publishing the first reading.");
}

void loop() {

```

invalid conversion from 'const char\*' to 'char\*' [-fpermissive]

| ^~~~~~

exit status 1

invalid conversion from 'const char\*' to 'char\*' [-fpermissive]

[Copy error messages](#)

The above problem is now solved, only now I came to the following problem.  
This something with Wifi.begin(ssid, password);

I couldn't get any further until here.

I spent half the afternoon looking for videos of linking an API index with an LED strip. All videos were about a UV index sensor instead of an API.

I found this very unfortunate and this also took a lot of time.

Because I didn't get any further, I started to delve further into setting various colors via the LED strip.

I found an article about - LEDStrip effecten voor NeoPixel and Fastled

First I learned some information about the color codes

"

### Tool: Color Picker

This tool can be useful when choosing colours: LED colors are made up of 3 numbers:

red, green and blue (RGB). Each number is a byte, so each number can have a value of 0 ... 255 (in hexadecimal: 00 ... FF).

Now our gray cells (usually) don't work with RGB numbers, and that's where this color picker can come in handy. You can choose a color here which returns the numbers in hexadecimal. Note that LED strips colors may be slightly different - they are not calibrated after all.

### Color Picker: FF0000

Usage: Click on the input field and a popup will appear where you can choose a color. After clicking on a color, the hexadecimal value will appear. We use this hexadecimal value in an Arduino Sketch as follows:

The first 2 characters represent RED,

The second setset of two characters are for GREEN and last 2 characters for BLUE.

We have to type '0x' in front of the values to indicate that they are hexadecimal numbers.

Example: For purple , the hexadecimal value is B700FE. So red is B7, green is 00 and blue is FE.

The first thing was to implement the fastLED framework

```
1 #include "FastLED.h"
2 #define NUM_LEDS 60
3 CRGB leds[NUM_LEDS];
4 #define PIN 6
5
6 void setup()
7 {
8     FastLED.addLeds<WS2811, PIN, GRB>(leds, NUM_LEDS).setCorrection( TypicalLEDStrip );
9 }
10
11 // *** REPLACE FROM HERE ***
12 void loop() {
13     // ---> here we call the effect function <---
14 }
15
16 // ---> here we define the effect function <---
17 // *** REPLACE TO HERE ***
18
19 void showStrip() {
20 #ifdef ADAFRUIT_NEOPixel_H
21     // NeoPixel
22     strip.show();
23 #endif
24 #ifndef ADAFRUIT_NEOPixel_H
25     // FastLED
26     FastLED.show();
27 #endif
28 }
29
30 void setPixel(int Pixel, byte red, byte green, byte blue) {
31 #ifdef ADAFRUIT_NEOPixel_H
32     // NeoPixel
33     strip.setPixelColor(Pixel, strip.Color(red, green, blue));
34 #endif
35 #ifndef ADAFRUIT_NEOPixel_H
36     // FastLED
37     leds[Pixel].r = red;
38     leds[Pixel].g = green;
39     leds[Pixel].b = blue;
40 #endif
41 }
42
43 void setAll(byte red, byte green, byte blue) {
```

On the yellow part there is different code that I can paste.  
I choose for the LEDStrip Effect - Fade In en Fade with my own color

The code was

```
1 void loop() {
2   FadeInOut(0xff, 0x77, 0x00);
3 }
4
5 void FadeInOut(byte red, byte green, byte blue){
6   float r, g, b;
7
8   for(int k = 0; k < 256; k=k+1) {
9     r = (k/256.0)*red;
10    g = (k/256.0)*green;
11    b = (k/256.0)*blue;
12    setAll(r,g,b);
13    showStrip();
14  }
15
16  for(int k = 255; k >= 0; k=k-2) {
17    r = (k/256.0)*red;
18    g = (k/256.0)*green;
19    b = (k/256.0)*blue;
20    setAll(r,g,b);
21    showStrip();
22  }
23 }
```

I was curious if could do some different Colors. After trying I understood that I need to paste more FadeInOut under the first one.

```
void loop() {
  FadeInOut(0x53, 0xff, 0x38);
  FadeInOut(0xf8, 0xff, 0x38);
  FadeInOut(0xff, 0x71, 0x1c);

}

void FadeInOut(byte red, byte green, byte blue){
  float r, g, b;

  for(int k = 0; k < 256; k=k+1) {
    r = (k/256.0)*red;
    g = (k/256.0)*green;
    b = (k/256.0)*blue;
    setAll(r,g,b);
    showStrip();
  }

  for(int k = 255; k >= 0; k=k-2) {
    r = (k/256.0)*red;
    g = (k/256.0)*green;
    b = (k/256.0)*blue;
    setAll(r,g,b);
    showStrip();
  }
}
```

The result



## Sources

<https://randomnerdtutorials.com/esp32-http-get-open-weather-map-thingspeak-arduino/>

<https://www.tweaking4all.nl/hardware/arduino/arduino-led-strip-effecten/#NeoPixelFramework>