

# The PepWake

Britt van Wonderen (500814927)

Klas: Emerging Technologies

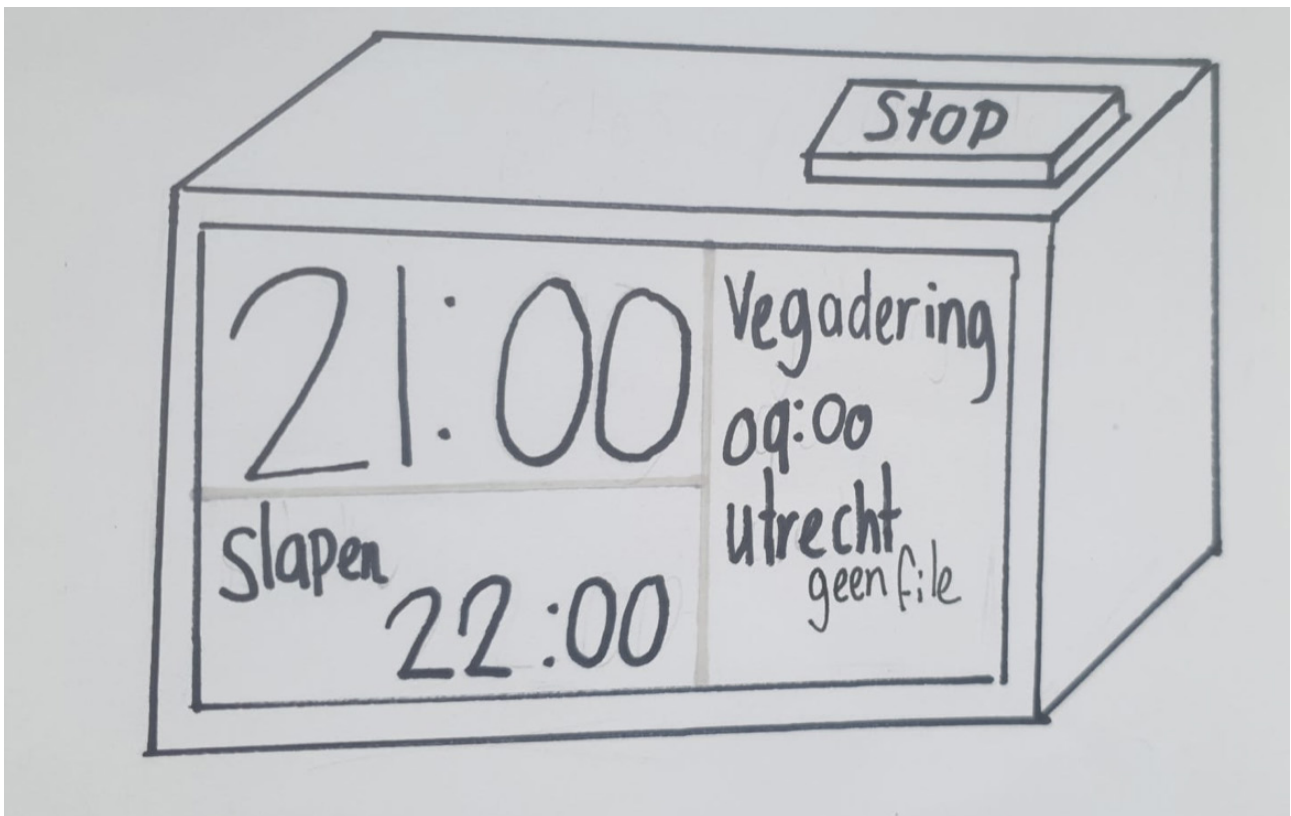
David de Vries & Harm van Vugt

27-10-2021

Internet of Things

## Introduction

For my final product of IoT, I had to make up an IoT product that was considered a level three product. I decided to make a alarm clock, that could calculate how late an user needed to switch off their devices (to minimise the exposure to blue lights) and when to go to bed so that they can get up and be well rested for the next day.



I decide to first focus on getting the time and date right and then I will look at how far I will come in regards to turning a led strip on and off at a certain time.

# Index

Tutorial 1	3
Step 1	3
Step 2	3
First problem	4
Step 2, part 2	4
Step 3	4
Second problem	6
Tutorial 2	8
Step 1	8
Step 2	8
Step 3	8
Tutorial 3	11
Step 1	11
Step 2	11
Step 3	11
Third problem	12
Source list	14

# Tutorial 1

## Step 1

### Make an Adafruit account

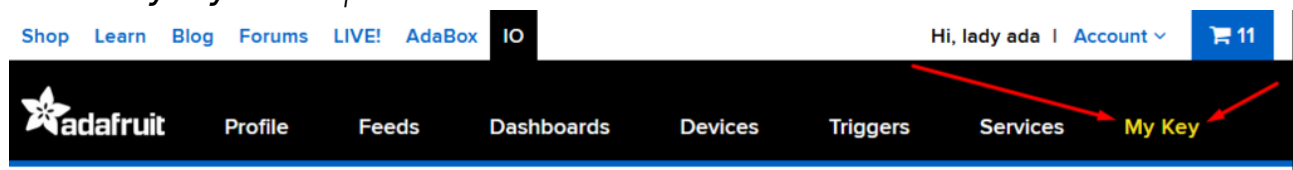
It's free! Visit <https://accounts.adafruit.com/> to register and make an account if you do not already have one (Adafruit, 2021).

Because of school I had already an account at this site. So I logged in and continue with the steps.

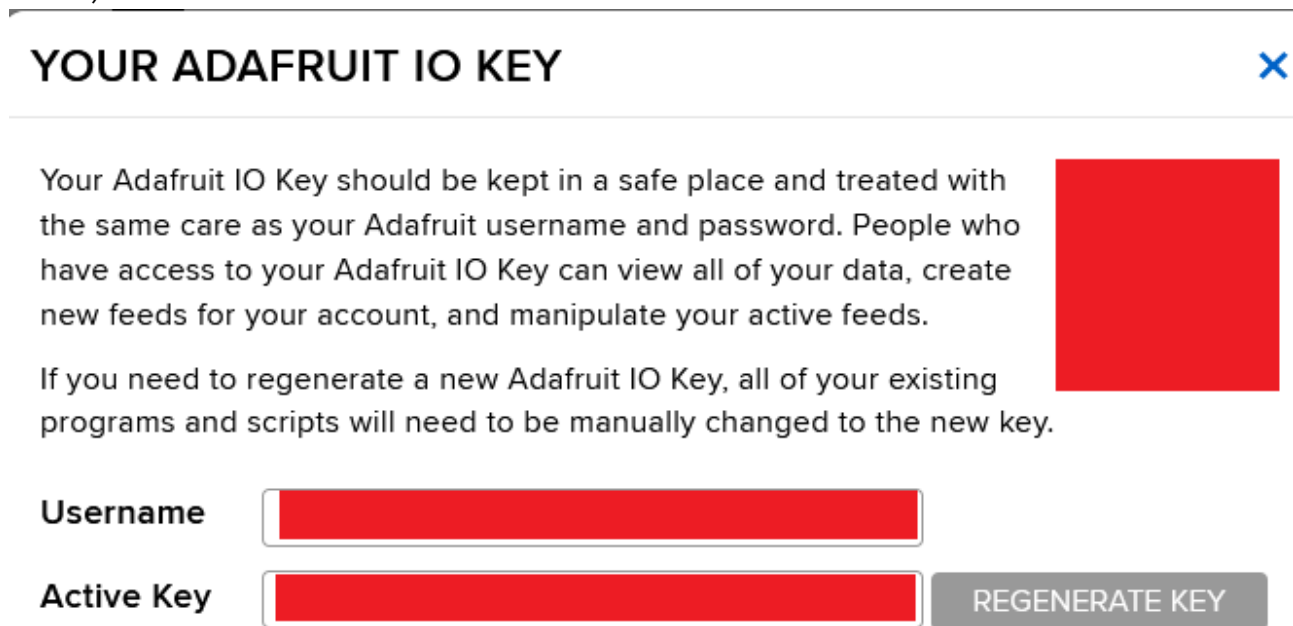
## Step 2

### Get your Adafruit IO Key

Click on **My Key** in the top bar



You will get a popup with your **Username** and **Key** (In this screenshot, we've covered it with red blocks)



[Hide Code Samples](#)

Go to your `secrets.py` file on your CIRCUITPY drive and add three lines for `aio_username`, `aio_key` and `timezone` so you get something like the following:

```
# This file is where you keep secret settings, passwords, and tokens!  
# If you put them in the code you risk committing that info or sharing it
```

```
secrets = {  
    'ssid' : 'home_wifi_network',  
    'password' : 'wifi_password',  
    'aio_username' : 'my_adafruit_io_username',
```

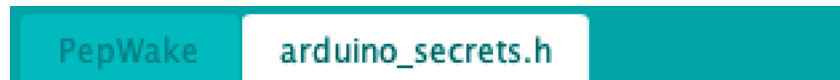
```
'aio_key' : 'my_adafruit_io_key',
'timezone' : "America/New_York", # http://worldtimeapi.org/timezones
}
```

The timezone is optional, if you don't have that entry, adafruit.io will guess your timezone based on geographic IP address lookup. You can visit <http://worldtimeapi.org/timezones> to see all the time zones available (even though we do not use worldtimeapi for time-keeping we do use the same time zone table) (Adafruit, 2021).

I got my username and password very easily but then I read that I needed to put it in my secrets.py file. I never heard of that before, so I didn't really know what to do. I first started Arduino, connected my board with my laptop and opened up a new file.

## First problem

I don't know how to open up a new tab. That's mainly because I'm fairly new with this program, I have just been using it for a few weeks. So that is why I looked up on the Internet how to open up a new tab and how to make a secret file. I looked at a site that explained how I could safely store my secrets in Arduino (Grandi, 2020) and I watched a Youtube tutorial (Arduino Tutorial, 2016). This helped me to get back on track, and I was able to make a new tab.



## Step 2, part 2

After I finally figured out how to make a new tab, I copied the code that is shown above and pasted that in the new tab. Afterwards I filled in all the required information. For the wifi I used my hotspot on my phone. I had to look up how I needed to fill in the timezone correctly, but I found a link in the tutorial and put in the correct timezone (*Full List of Timezones, via World Time API: Simple JSON/Plain-Text API to Obtain the Current Time in, and Related Data about, a Timezone., z.d.*).

## Step 3

### Upload Test Python Code

This code is like the Internet Test code from before, but this time it will connect to adafruit.io and get the local time

[Download File](#)

[Copy Code](#)

```
import ipaddress
import ssl
import wifi
import socketpool
import adafruit_requests
import secrets
```

```
TEXT_URL = "http://wifitest.adafruit.com/testwifi/index.html"
JSON_QUOTES_URL = "https://www.adafruit.com/api/quotes.php"
JSON_STARS_URL = "https://api.github.com/repos/adafruit/circuitpython"
```

```

# Get wifi details and more from a secrets.py file
try:
    from secrets import secrets
except ImportError:
    print("WiFi secrets are kept in secrets.py, please add them there!")
    raise

# Get our username, key and desired timezone
aio_username = secrets["aio_username"]
aio_key = secrets["aio_key"]
location = secrets.get("timezone", None)
TIME_URL = "https://io.adafruit.com/api/v2/%s/integrations/time/strftime?x-aio-key=%s" %
(aio_username, aio_key)
TIME_URL += "&fmt=%25Y-%25m-%25d+%25H%3A%25M%3A%25S.%25L+%25j+%25u+%25z+%25Z"

print("ESP32-S2 Adafruit IO Time test")

print("My MAC addr:", [hex(i) for i in wifi.radio.mac_address])

print("Available WiFi networks:")
for network in wifi.radio.start_scanning_networks():
    print("\t%s\t\tRSSI: %d\tChannel: %d" % (str(network.ssid, "utf-8"),
        network.rssi, network.channel))
wifi.radio.stop_scanning_networks()

print("Connecting to %s"%secrets["ssid"])
wifi.radio.connect(secrets["ssid"], secrets["password"])
print("Connected to %s!"%secrets["ssid"])
print("My IP address is", wifi.radio.ipv4_address)

ipv4 = ipaddress.ip_address("8.8.4.4")
print("Ping google.com: %f ms" % wifi.radio.ping(ipv4))

pool = socketpool.SocketPool(wifi.radio)
requests = adafruit_requests.Session(pool, ssl.create_default_context())

print("Fetching text from", TIME_URL)
response = requests.get(TIME_URL)
print("-" * 40)
print(response.text)
print("-" * 40)
(Adafruit, 2021).

```

I put the code that is shown above in the first tab that I have. Then I clicked on uploading and watched how the program tried to upload, but unfortunately I had an error midway.

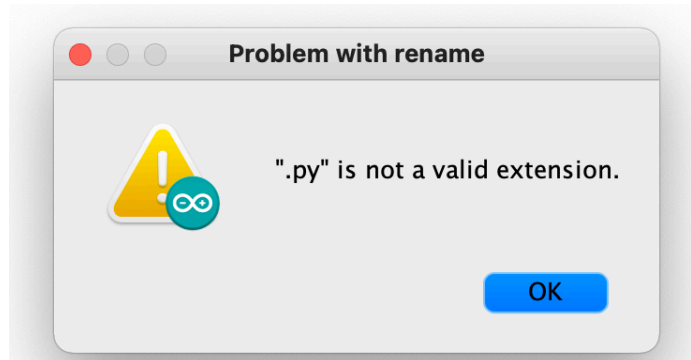
## Second problem

I looked at the error to try to understand what it meant, and then I saw that my second file was named differently than is reported in the first file. In my first tab, it reads; secrets.py, my second tab is named arduino\_secrets.h. I first tried to rename the second file, but unfortunately .py is not a valid extension. Then I tried to rename all the secrets.py to arduino\_secrets.h.

I tried to upload it again, but unfortunately it seems like I did not fix the problem. I deleted the line of code that is highlighted in the document, to see what would happen. Unfortunately other error's popped up, I decided undo the deletes that I just did and look at the serial monitor. In the serial monitor I saw a few weird signs, but then it reported that it connected to my hotspot. I decided to look for the answers, regarding the errors, on the internet, to see if that could help me.

I unfortunately could not find the answer to my problems on the internet.

These are the errors that I saw in my document;



```
PepWake: arduino_secrets.h
import ipaddress
import ssl
import wifi
import socketpool
import adafruit_requests
import secrets

TEXT_URL = "http://wifitest.adafruit.com/testwifi/index.html"
JSON_QUOTES_URL = "https://www.adafruit.com/api/quotes.php"
JSON_STARTS_URL = "https://api.github.com/repos/adafruit/circuitpython"

# Get wifi details and more from a secrets.py file
try:
    from secrets import secrets
except ImportError:
    print("WiFi secrets are kept in secrets.py, please add them there!")
    raise

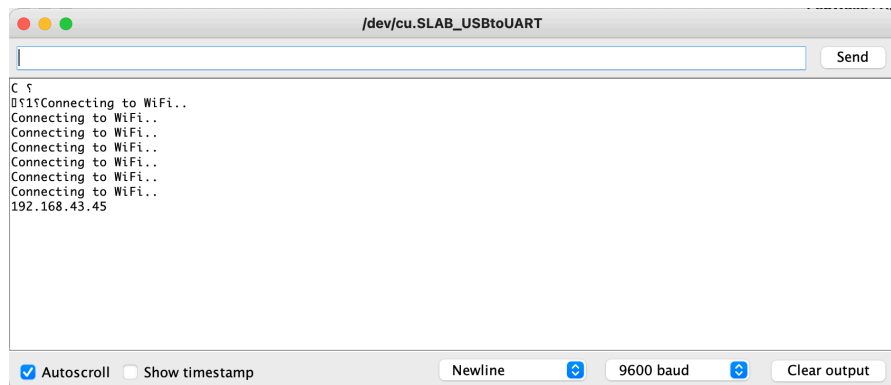
# Get our username, key and desired timezone
aio_username = secrets["aio_username"]
aio_key = secrets["aio_key"]
location = secrets.get("timezone", None)
TIME_URL = "https://io.adafruit.com/api/v2/%s/integrations/time/strftime?x-aio-key=%s" % (aio_username, aio_key)
TIME_URL += "&format=%25Y-%25m-%25d+%25H%M%S%25p%25Z"

print("ESP32-S2 Adafruit IO Time test")

print("My MAC addr:", [hex(i) for i in wifi.radio.mac_address])

print("Available WiFi networks:")
for network in wifi.radio.start_scanning_networks():
    print("\t\t\t\t\tRSSI: %d\tChannel: %d" % (str(network.rssi), "utf-8"),
          network.rssi, network.channel)
wifi.radio.stop_scanning_networks()

invalid preprocessing directive #Get
PepWake:13:3: error: invalid preprocessing directive #Get
13 | # Get wifi details and more from a secrets.py file
    | ^~~
PepWake:20:3: error: invalid preprocessing directive #Get
20 | # Get our username, key and desired timezone
    | ^~~
exit status 1
invalid preprocessing directive #Get
```



Arduino: 1.8.16 (Mac OS X), Board: "NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm\_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200"

PepWake:13:3: error: invalid preprocessing directive #Get  
13 | # Get wifi details and more from a arduino\_secrets.h file  
| ^~~

PepWake:20:3: error: invalid preprocessing directive #Get  
20 | # Get our username, key and desired timezone

| ^~~

exit status 1

invalid preprocessing directive #Get

This report would have more information with  
"Show verbose output during compilation"  
option enabled in File -> Preferences.

## Tutorial 2

I decided to try another tutorial for getting the time and date, I found another tutorial online.

### Step 1

But before venturing further into this tutorial, you should have the ESP8266 add-on installed in your Arduino IDE. Follow below tutorial to prepare your Arduino IDE to work with the ESP8266, if you haven't already (Engineers, 2018).

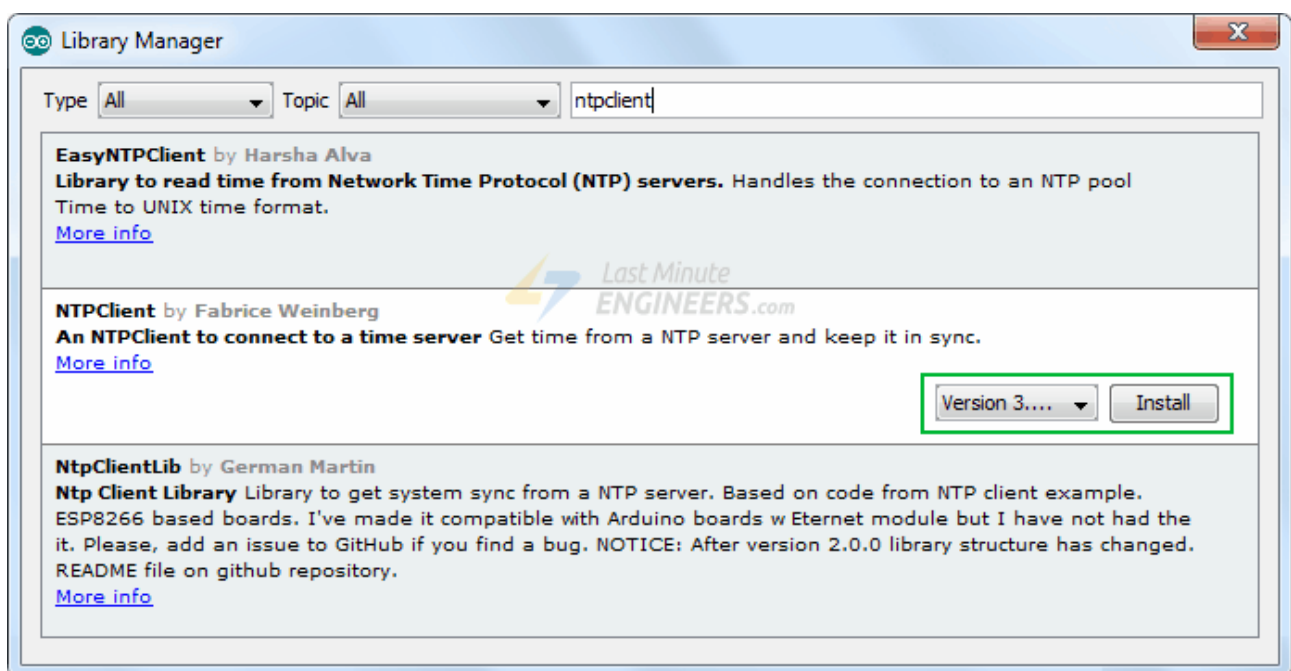
Even though I just begun using Arduino, I do have the required Arduino IDE ESP8266 already on my laptop.

### Step 2

#### Installing NTP Client Library

The easiest way to get date and time from an NTP server is using an [NTP Client from arduino libraries](#). Follow the next steps to install this library in your Arduino IDE.

Navigate to the Sketch > Include Library > Manage Libraries...Wait for Library Manager to



download libraries index and update list of installed libraries.

Filter your search by typing 'ntpclient'. There should be a couple entries. Look for NTPClient by Fabrice Weinberg. Click on that entry, and then select Install (Engineers, 2018).

I started a new file, went to my library and installed NTPclient. After I successfully downloaded the library, I went to the next step.

### Step 3

#### Getting Current Day and Time from NTP Server

The following sketch will give you complete understanding on how to get current day and time from the NTP Server.



Before you head for uploading the sketch, you need to make some changes to make it work for you.

- You need to modify the following two variables with your network credentials, so that ESP8266 can establish a connection with existing network.

```
const char* ssid = "YOUR_SSID";
```

- ```
const char* password = "YOUR_PASS";
```

- You need to adjust the UTC offset for your timezone in milliseconds. Refer the [list of UTC time offsets](#). Here are some examples for different timezones:

- For UTC -5.00 :  $-5 * 60 * 60$  : -18000
- For UTC +1.00 :  $1 * 60 * 60$  : 3600
- For UTC +0.00 :  $0 * 60 * 60$  : 0

```
const long utcOffsetInSeconds = 3600;
```

Once you are done, go ahead and try the sketch out.

```
#include <NTPClient.h>
```

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

```
#include <WiFiUdp.h>
```

```
const char *ssid = "YOUR_SSID";
```

```
const char *password = "YOUR_PASS";
```

```
const long utcOffsetInSeconds = 3600;
```

```
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",  
"Friday", "Saturday"};
```

```
// Define NTP Client to get time
```

```
WiFiUDP ntpUDP;
```

```
NTPClient timeClient(ntpUDP, "pool.ntp.org", utcOffsetInSeconds);
```

```
void setup(){
```

```
  Serial.begin(115200);
```

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

```
  while ( WiFi.status() != WL_CONNECTED ) {
```

```
    delay ( 500 );
```

```
    Serial.print ( "." );
```

```
  }
```

```
  timeClient.begin();
```

```
}
```

```
void loop() {
```

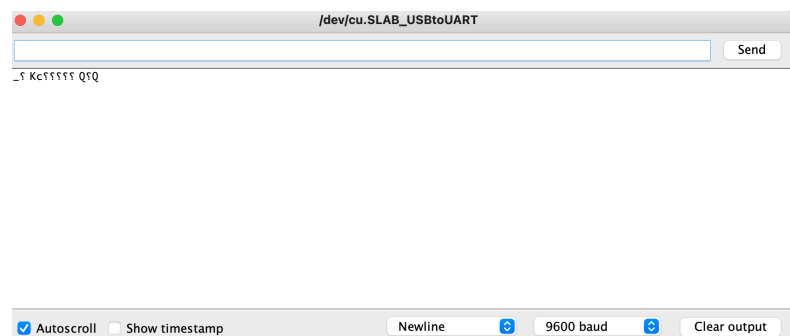
```
timeClient.update();

Serial.print(daysOfTheWeek[timeClient.getDay()]);
Serial.print(", ");
Serial.print(timeClient.getHours());
Serial.print(":");
Serial.print(timeClient.getMinutes());
Serial.print(":");
Serial.println(timeClient.getSeconds());
//Serial.println(timeClient.getFormattedTime());

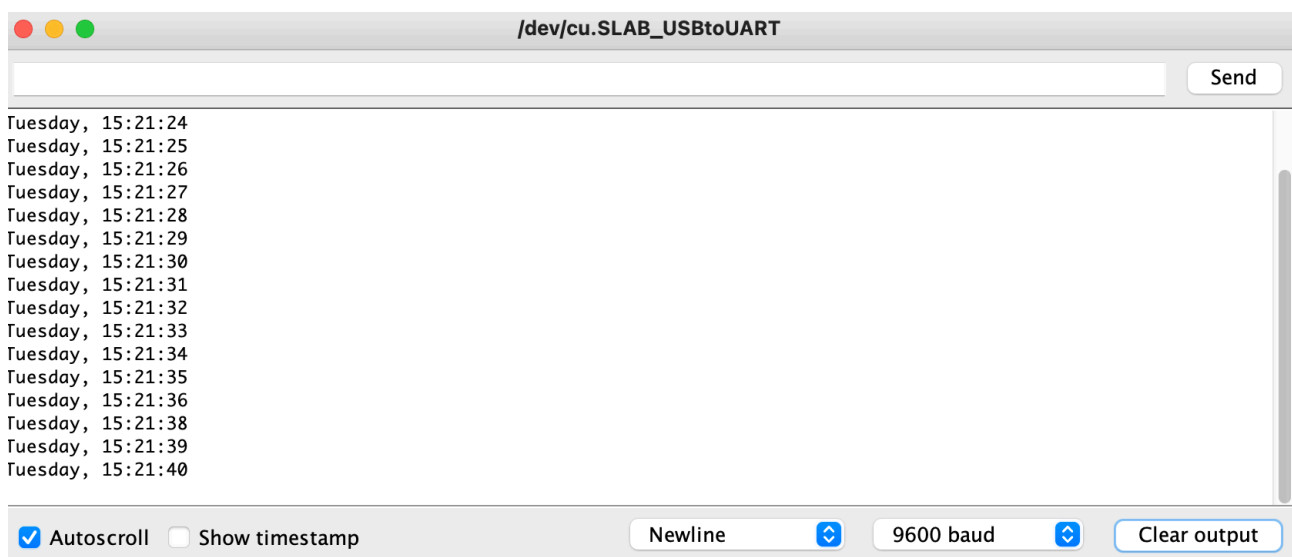
delay(1000);
}
(Engineers, 2018).
```

I added a second tab in my new file, so that I could input my hotspot data. I was about to look for the name for the second tab, when I realised that I do not have to do that. I put in the code that is shown above and added my own personal data. In the tutorial they used the same timezone that I am currently in.

I then uploaded the sketch to see what it would do. The uploading worked, but when I looked in the serial monitor, all I could see were a few weird symbols. I then realised that I should change the Baud rate in the document. I changed it and uploaded it again.



It worked this time. I could clearly see what day it is and what time it is currently.



## Tutorial 3

I decided add the code of a led light strip that will go on and off at an appointed time, to the document in Arduino that I just did. In class I once connected a led light strip with Adafruit neopixel. I decided to use this as a base for what I want to do.

The tutorial for my class was unfortunately in Dutch so I translated it to the best of my abilities.

### Step 1

*Instal the library:*

1. *Sketch > include library > Manage Libraries...*

2. *Search for adafruit neopixel*

3. *Instal Adafruit NeoPixel*

*(Aansluiten sensoren en actuatoren NodeMCU, z.d.)*

I already had this library included in Arduino.

### Step 2

*Connecting wires (THE WIRES MAY HAVE OTHER COLORS)*

*Wire from 5V to 3V3*

*Middle wire (Din) to D5*

*GND wire to GND*

*(Aansluiten sensoren en actuatoren NodeMCU, z.d.)*



I connected the wires, as seen on the example picture above.

### Step 3

*Code*

*File -> Examples -> Neopixel -> Simple (you have to scroll all the way down)*

*Change the value behind PIN to D5*

*Enter the number of LEDs in your strip after NUMPIXELS.*

*(Aansluiten sensoren en actuatoren NodeMCU, z.d.)*

Because I didn't want to compromise my code that I already did with the time and date, I decided to make a new tab in my document, I named it LED. Next I open up file —> examples —> Neopixel —> simple, and got a new documents with all the code already in it. I just had to change the value number behind PIN to D5. After I done that I counted how many NUMPIXELS there are on my led strip that I have. I counted 10 LED lights on my strip, in the documents I put in the amount of led lights.

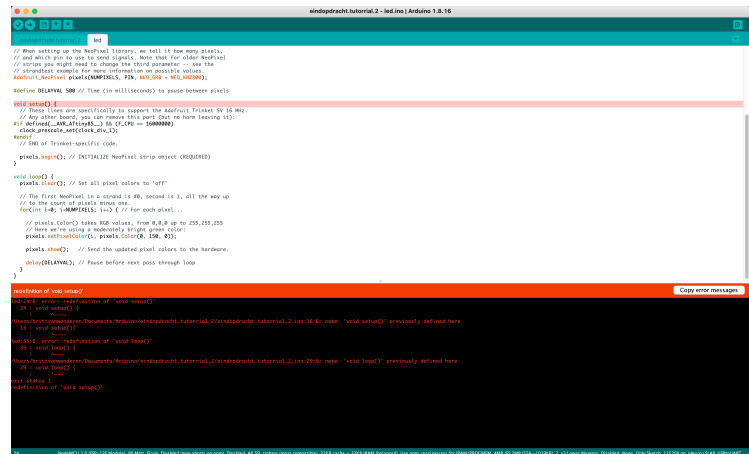
Afterwards I clicked on upload to see what it would do, but unfortunately I've got a whole row of errors.

## Third problem

When I look at the errors that are listed below, I don't really understand what they mean. I tried to read them one by one, to see if I could understand them but unfortunately I don't.

These are the error's in my document;

Arduino: 1.8.16 (Mac OS X), Board: "NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm\_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200"



led:24:6: error: redefinition of 'void setup()'

```
24 | void setup() {  
    | ^~~~~~
```

/Users/brittvanwonderen/Documents/Arduino/eindopdracht.tutorial.2/

eindopdracht.tutorial.2.ino:16:6: note: 'void setup()' previously defined here

```
16 | void setup(){  
    | ^~~~~~
```

led:35:6: error: redefinition of 'void loop()'

```
35 | void loop() {  
    | ^~~~~~
```

/Users/brittvanwonderen/Documents/Arduino/eindopdracht.tutorial.2/

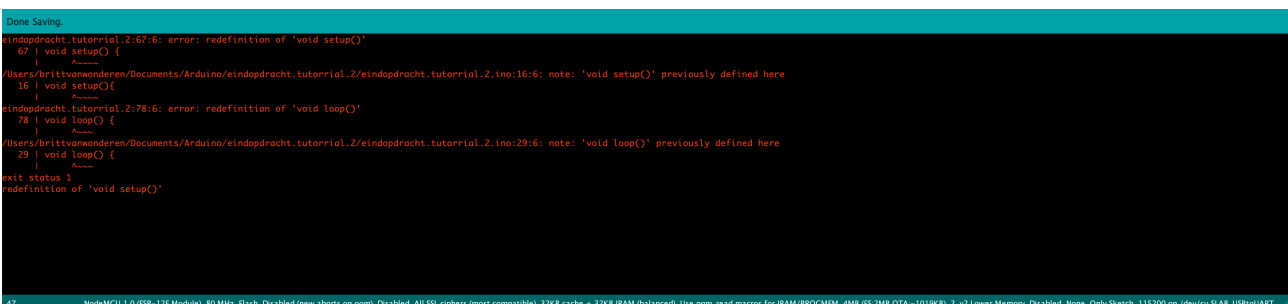
eindopdracht.tutorial.2.ino:29:6: note: 'void loop()' previously defined here

```
29 | void loop() {  
    | ^~~~~~
```

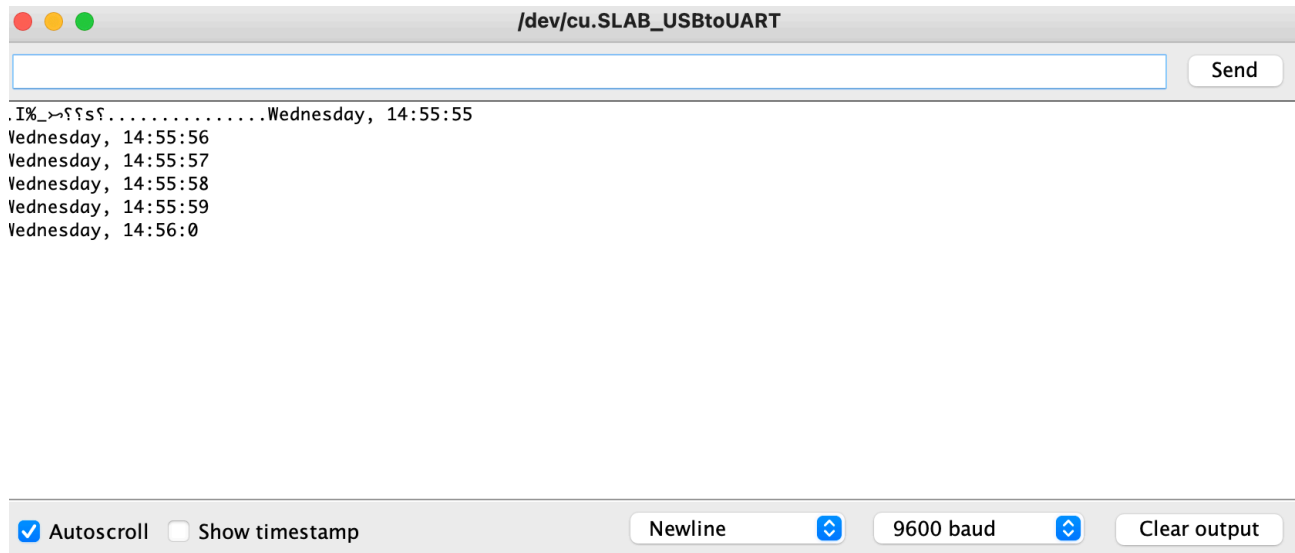
exit status 1

redefinition of 'void setup()'

This report would have more information with  
"Show verbose output during compilation"  
option enabled in File -> Preferences.



When I look in my serial monitor, I can still see the time and date. I can't see anything in the serial monitor regarding the led strip.



I decided to conclude my project here, I cannot go further because I don't know what the errors in my code mean.

## Source list

*Aansluiten sensoren en actuatoren NodeMCU.* (z.d.). Google Docs. Consulted on 27 oktober 2021, from [https://docs.google.com/document/d/1l\\_9xpU-n7Rhxi-L3wl72Kvc3GQ9q6tFI955J2NTYF9k/edit](https://docs.google.com/document/d/1l_9xpU-n7Rhxi-L3wl72Kvc3GQ9q6tFI955J2NTYF9k/edit)

Arduino Tutorial | *How to Organize your Code using Multi File Tabs on the Arduino IDE.* (2016, 9 december). [Video]. YouTube. <https://www.youtube.com/watch?v=gxLpob4fiWQ>

Adafruit. (2021, 19 mei). *Getting The Date & Time.* Consulted on 26 oktober 2021, from <https://learn.adafruit.com/adafruit-metro-esp32-s2/getting-the-date-time>

*Full list of Timezones, via World Time API: Simple JSON/plain-text API to obtain the current time in, and related data about, a timezone.* (z.d.). Worldtimeapi. Consulted on 26 oktober 2021, from <http://worldtimeapi.org/timezones>

Grandi, A. (2020, 16 december). *How to safely store Arduino secrets.* Andrea Grandi. Consulted on 26 oktober 2021, from <https://www.andreagrandi.it/2020/12/16/how-to-safely-store-arduino-secrets/>

Engineers, L. M. (2020, 18 december). *Getting Date & Time From NTP Server With ESP8266 NodeMCU.* Last Minute Engineers. Consulted on 26 oktober 2021, from <https://lastminuteengineers.com/esp8266-ntp-server-date-time-tutorial/>