



elkim.no

Brukermanual for værstasjon

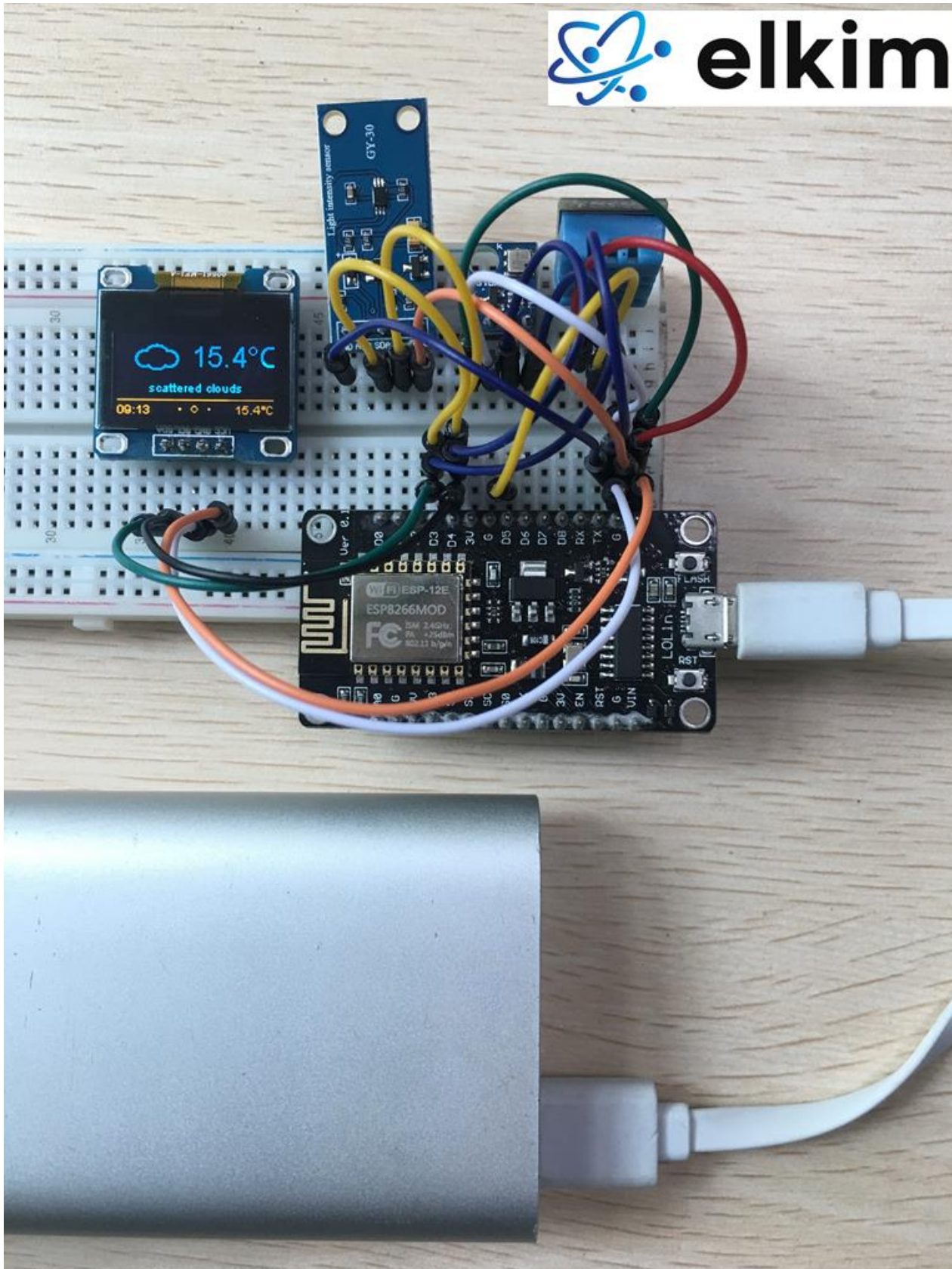
INNHOOLD

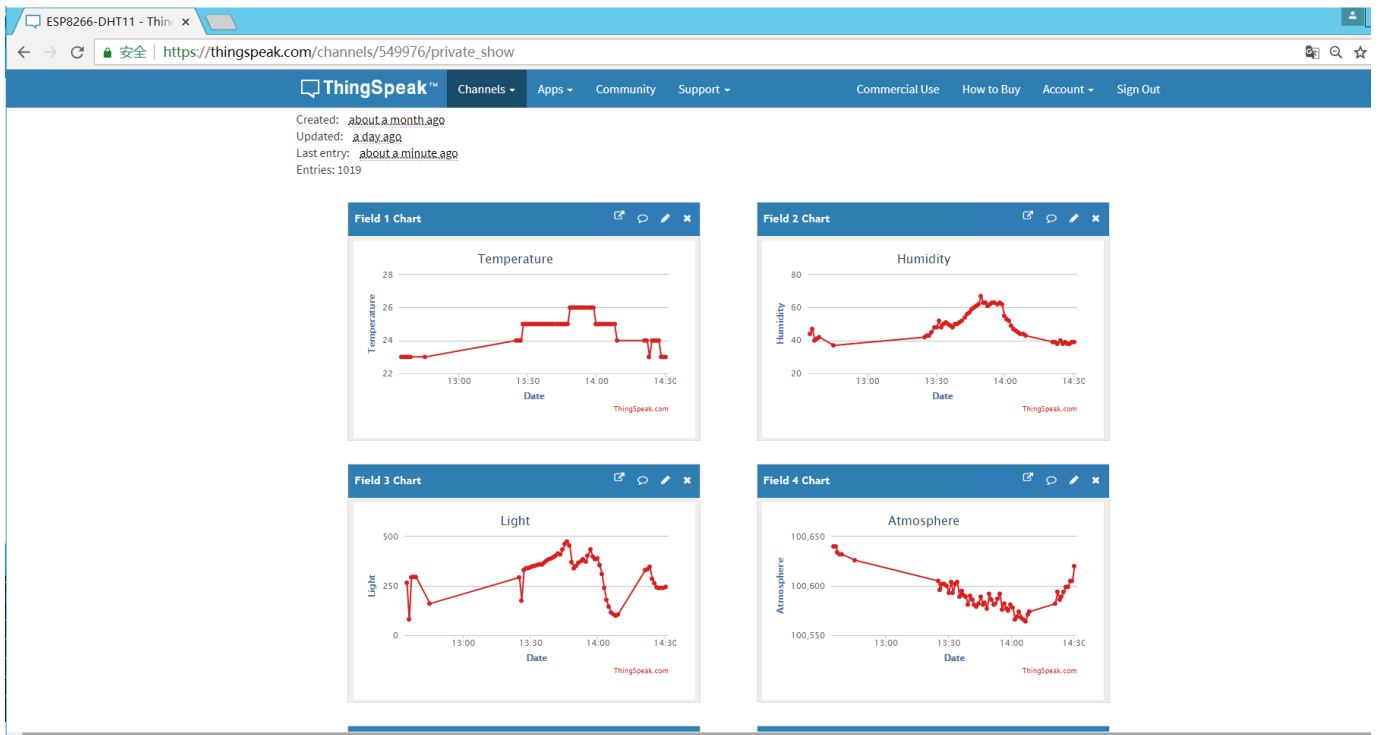
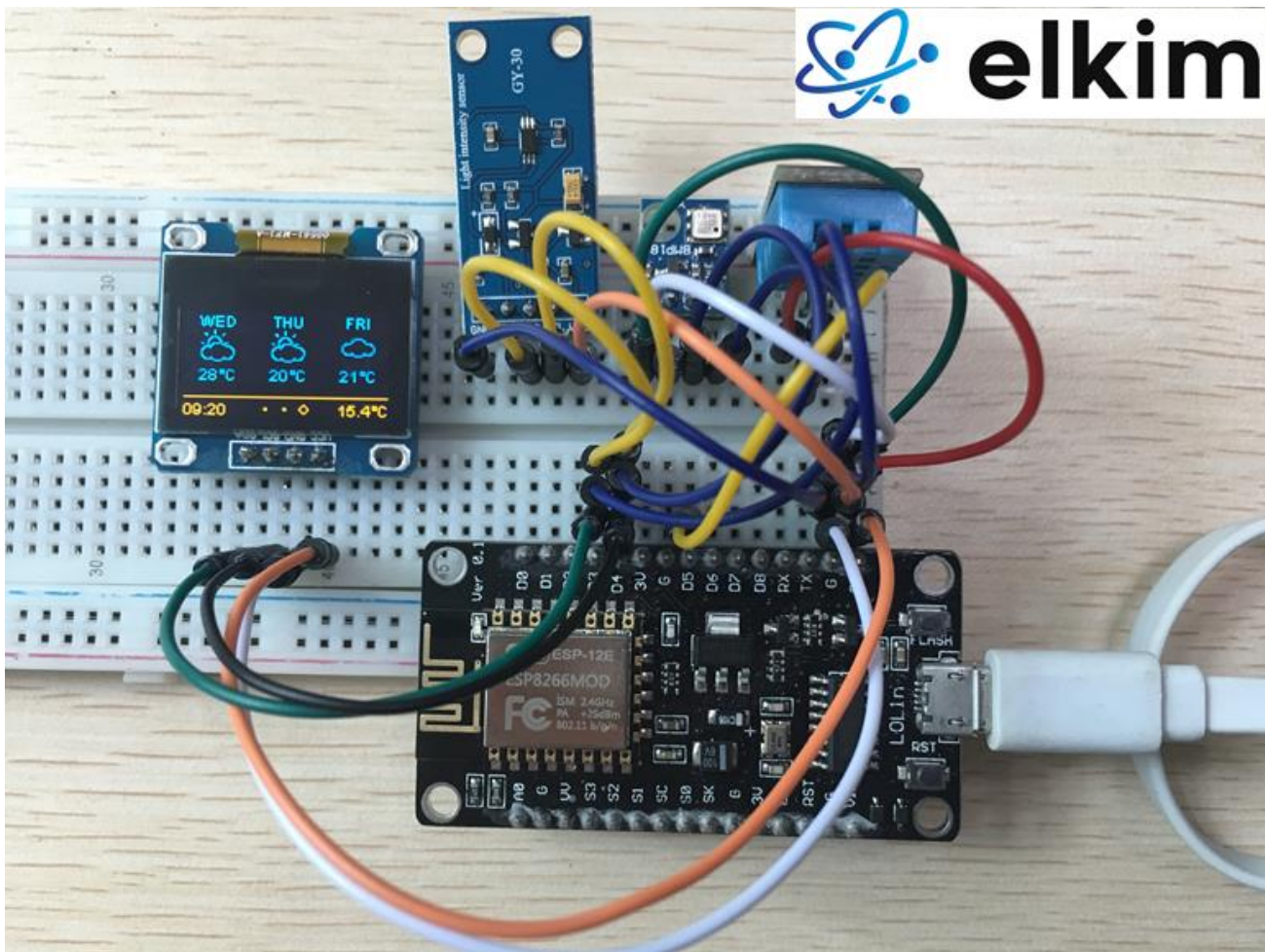
Veiledning: Hvordan du lager en miniatyr værstasjon ved hjelp av Arduino IDE	2
Funksjoner:.....	2
Resultat:	3
Koblingskjema.....	5
DRIVER installering.....	6
Steg 2: Installer Arduino IDE	7
Steg 3: BRENNE FASTVARE TIL ESP8266.....	9
Steg 4: Koble til komponenter	11
Steg 5: Registrer OpenWeathermap, thingspeak ny konto	14
Steg 6: IMPORTER VÆRSTASJONSKODEN I ARDUINO IDE.....	17
Steg 7: Legg til Library / Bibliotek.....	19
Steg 8: Modifiser WeatherStation.ino filen	21
Steg 9: Sett opp Board og Port igjen	24
Steg 10: Overfør kode til ESP8266	25
Steg 11: Resultat	26

FUNKSJONER:

1. Hent værdata fra OpenWeathermap, som viser været i dag og værmeldingen for de neste 3 dagene i hvilken som helst by i verden.
2. Les av gjeldende temperatur og luftfuktighet.
3. Les av atmosfærisk trykk og lysintensitet.
4. Last opp temperatur-, luftfuktighets-, atmosfærisk trykk- og lysintensitetsdata til thingspeak.com med jevne mellomrom.
5. Vis værmelding på skjermen og vis miljøovervåkingsdiagrammet på thingspeak.com.

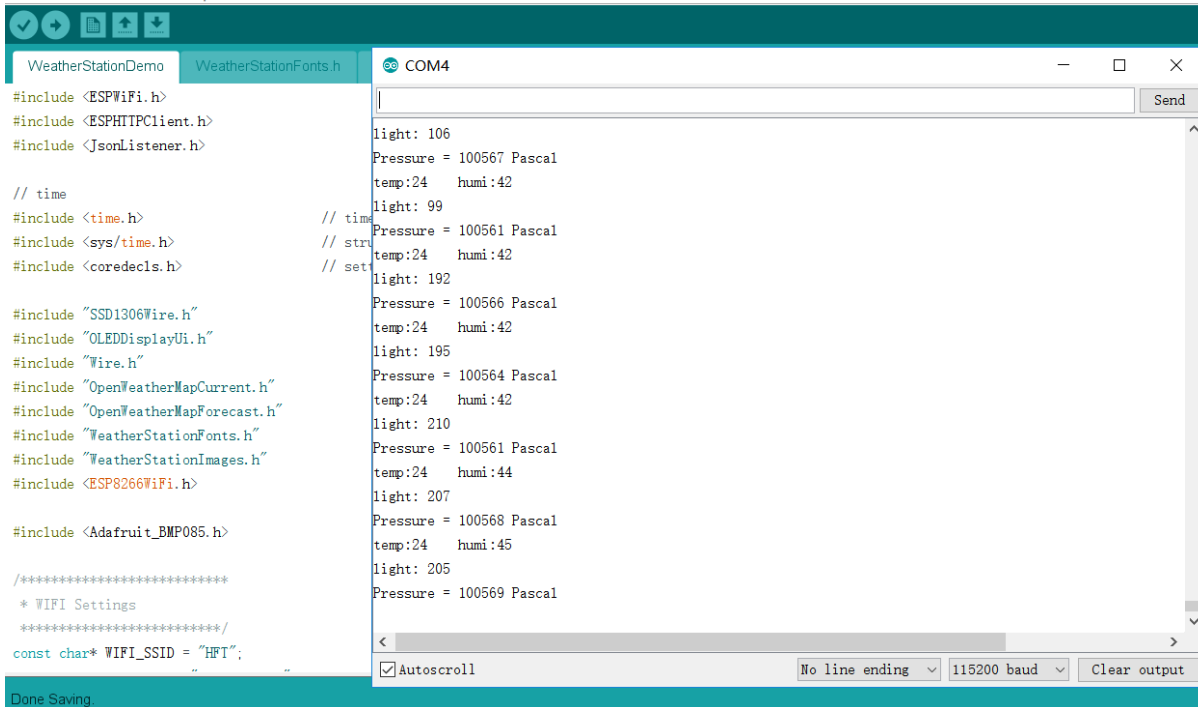
RESULTAT:





WeatherStationDemo | Arduino 1.8.4

File Edit Sketch Tools Help



```

#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <JsonListener.h>

// time
#include <time.h>
#include <sys/time.h>
#include <coredecls.h>

#include "SSD1306Wire.h"
#include "OLEDDisplayUi.h"
#include "Wire.h"
#include "OpenWeatherMapCurrent.h"
#include "OpenWeatherMapForecast.h"
#include "WeatherStationFonts.h"
#include "WeatherStationImages.h"
#include <ESP8266WiFi.h>

#include <Adafruit_BMP085.h>

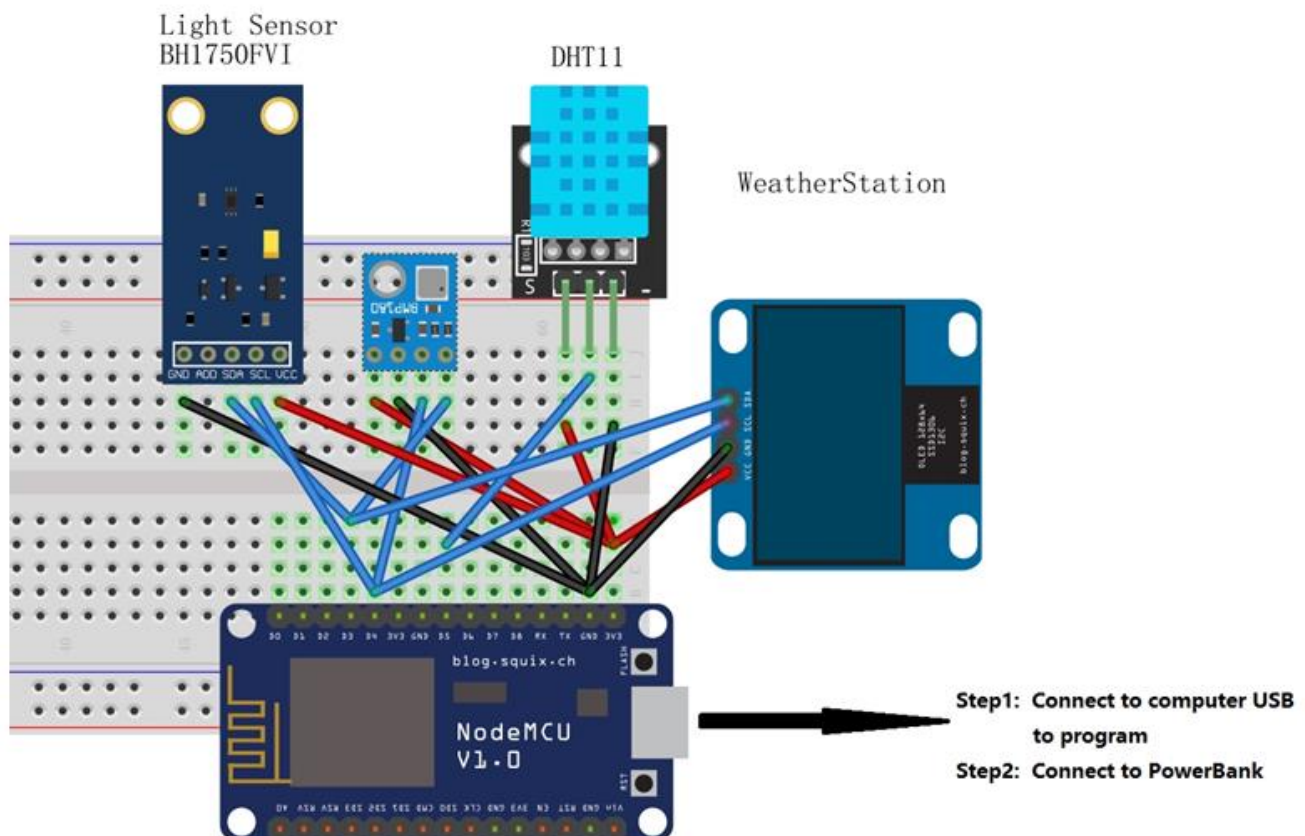
/*****
 * WIFI Settings
 *****/
const char* WIFI_SSID = "HFT";
  
```

```

light: 106
Pressure = 100567 Pascal
temp:24 humi:42
light: 99
Pressure = 100561 Pascal
temp:24 humi:42
light: 192
Pressure = 100566 Pascal
temp:24 humi:42
light: 195
Pressure = 100564 Pascal
temp:24 humi:42
light: 210
Pressure = 100561 Pascal
temp:24 humi:44
light: 207
Pressure = 100568 Pascal
temp:24 humi:45
light: 205
Pressure = 100569 Pascal
  
```

Autoscroll | No line ending | 115200 baud | Clear output


KOBLINGSSKJEMA



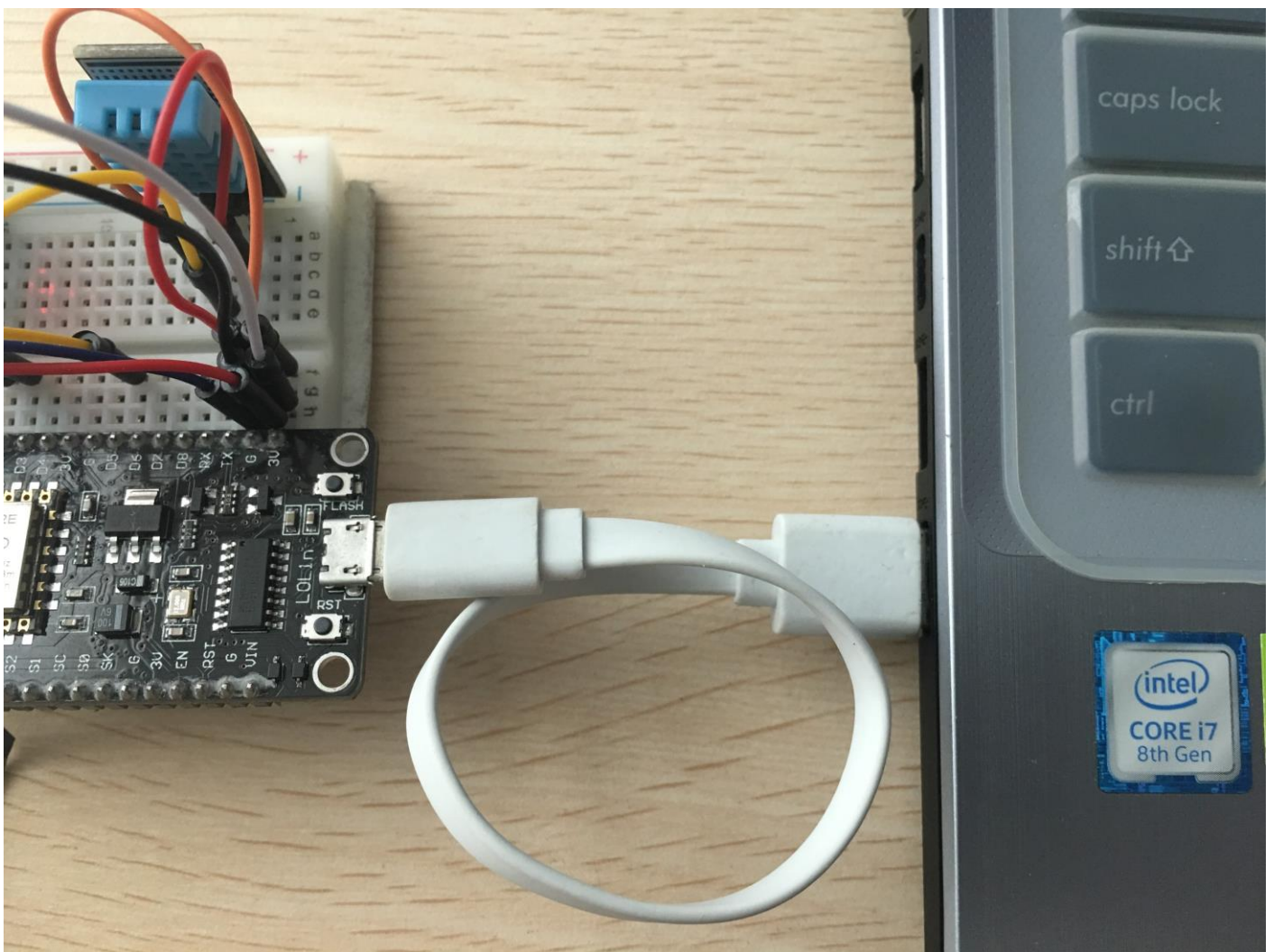
DRIVER INSTALLERING

I hovedsak trenger du ikke installere drivere på din datamaskin. Dersom du ikke får tak i USB kortet kan du installere de medfølgende driverne.

1) Vi trenger at datamaskinen gjenkjenner seriellporten, så installerer USB-til-seriell:

 [usb-to-serial-win10.zip](#)

2) Etter installasjonen, koble til ESP8266 og datamaskinen vil kjenne igjen enheten:



STEG 2: INSTALLER ARDUINO IDE

Last ned, og installer ARDUINO IDE:

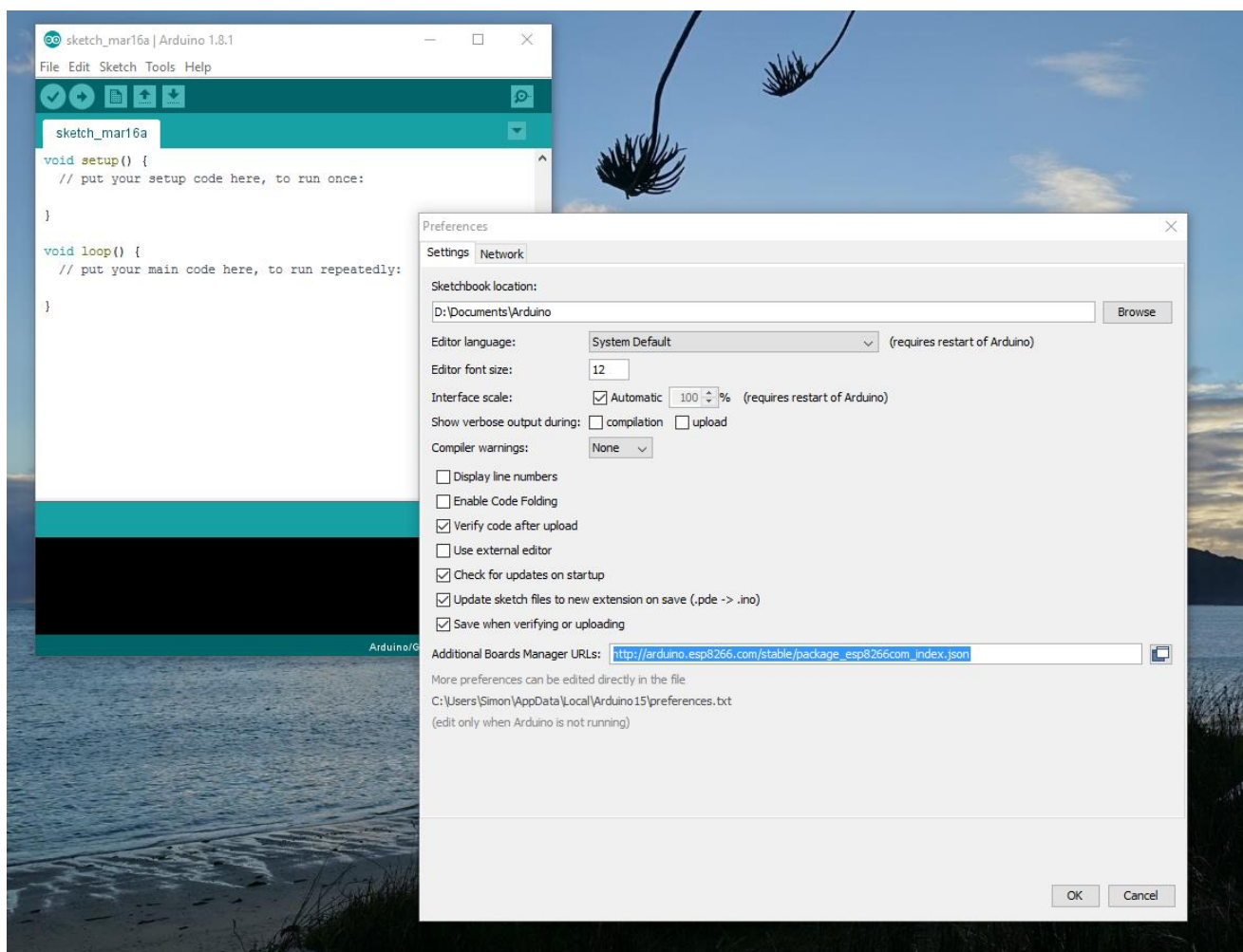
<https://www.arduino.cc/en/software>

Åpne Arduino IDE:

File > Preferences

Legg inn følgende link i "Additional Boards Manager URLs:" felte:

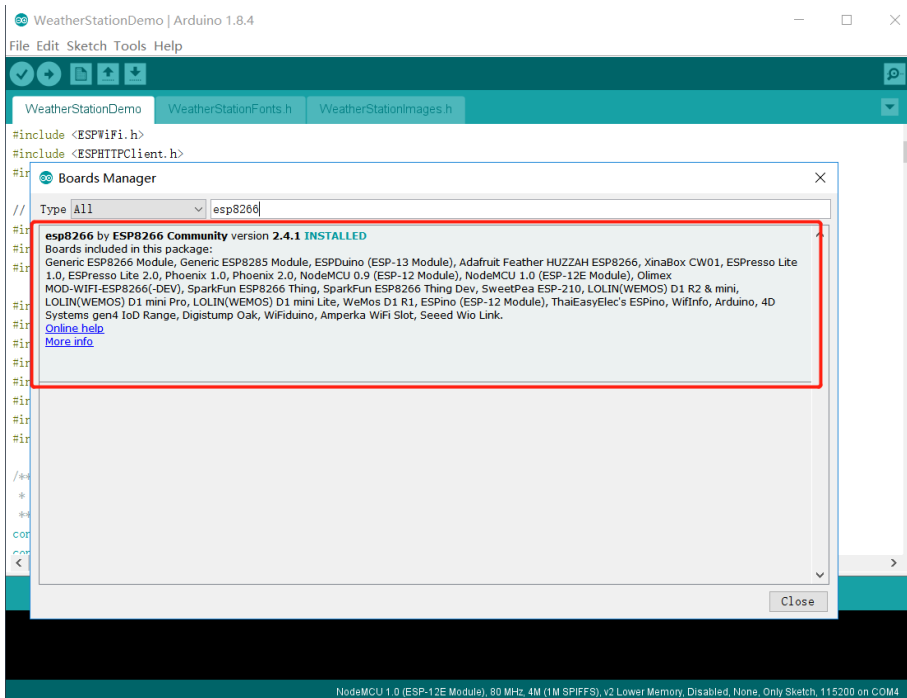
http://arduino.esp8266.com/stable/package_esp8266com_index.json



4) Trykk ok for å lukke dialog boksen.

5) Gå til Tools > Board > Board Manager

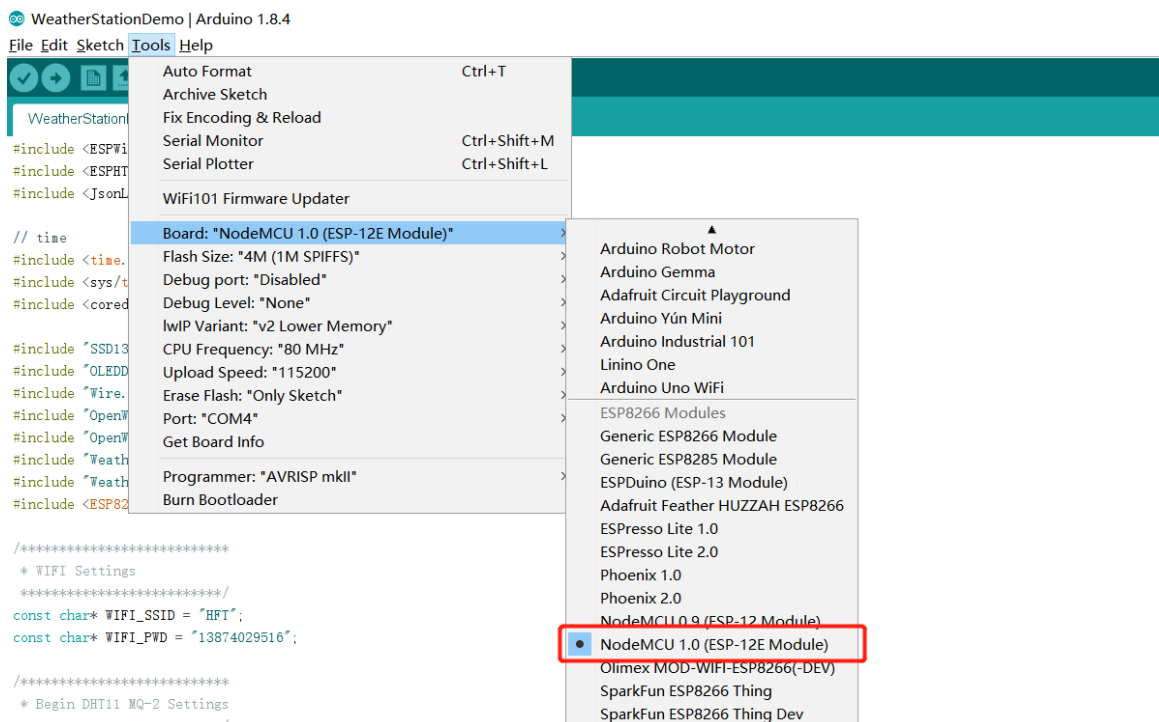
6) Søk etter "ESP8266" og velg "ESP8266 by ESP8266 Community"



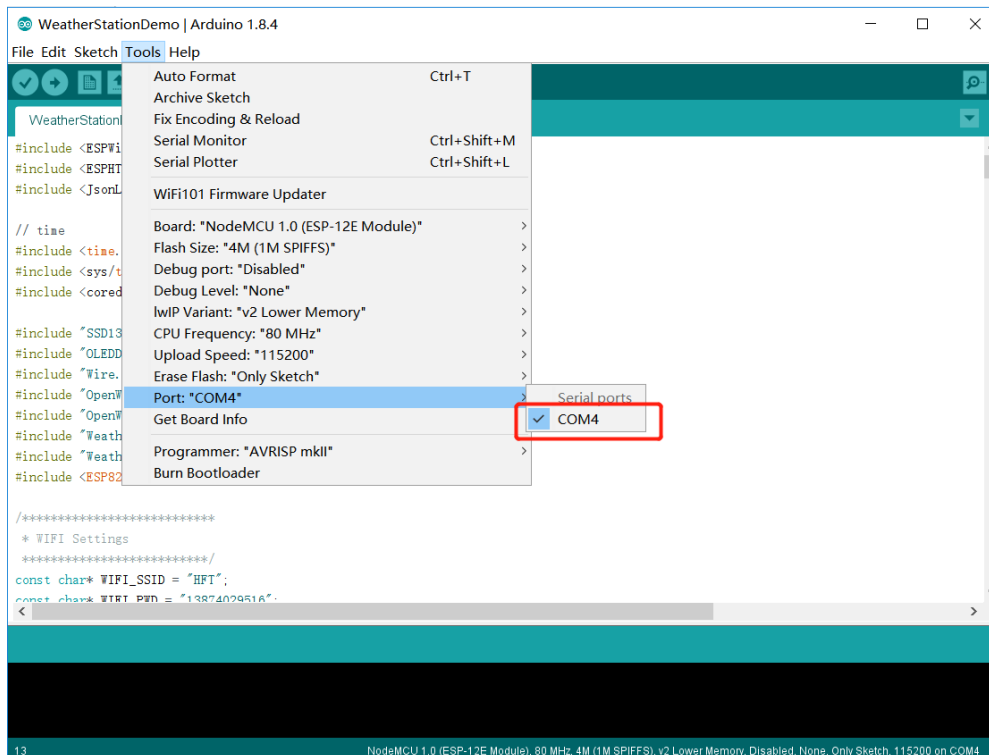
7) Trykk på "Install" for å installere.

8) Etter at installasjonen er ferdig (det tar ca 1-2 minutter) så kan du lukke dialogboksen (Close knappen)

9) Gå tilbake til to Tools > Boards og du må legge til noen "nye" brett nederst på listen. Velg "NodeMCU 1.0 (ESP-12E Module)".



10) Velg korrekt COM port (du får som regel kun opp én port):

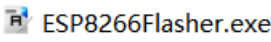
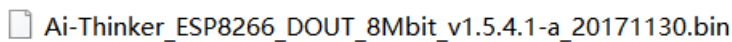


STEG 3: BRENNE FASTVARE TIL ESP8266

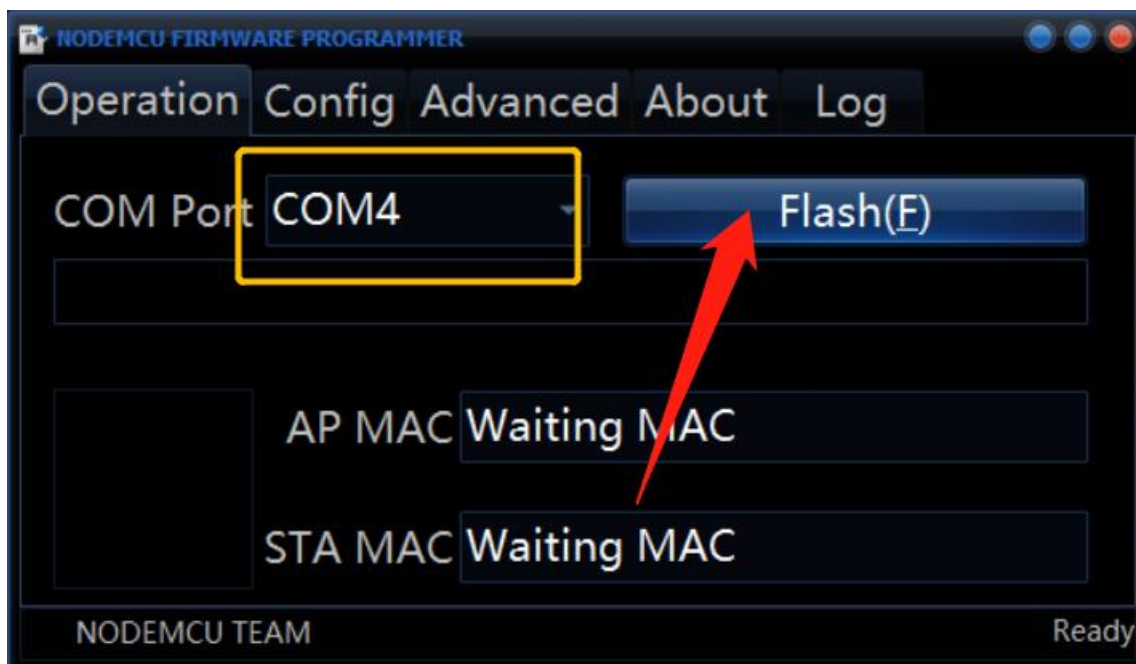
1) ESP8266 er koblet til datamaskinen:

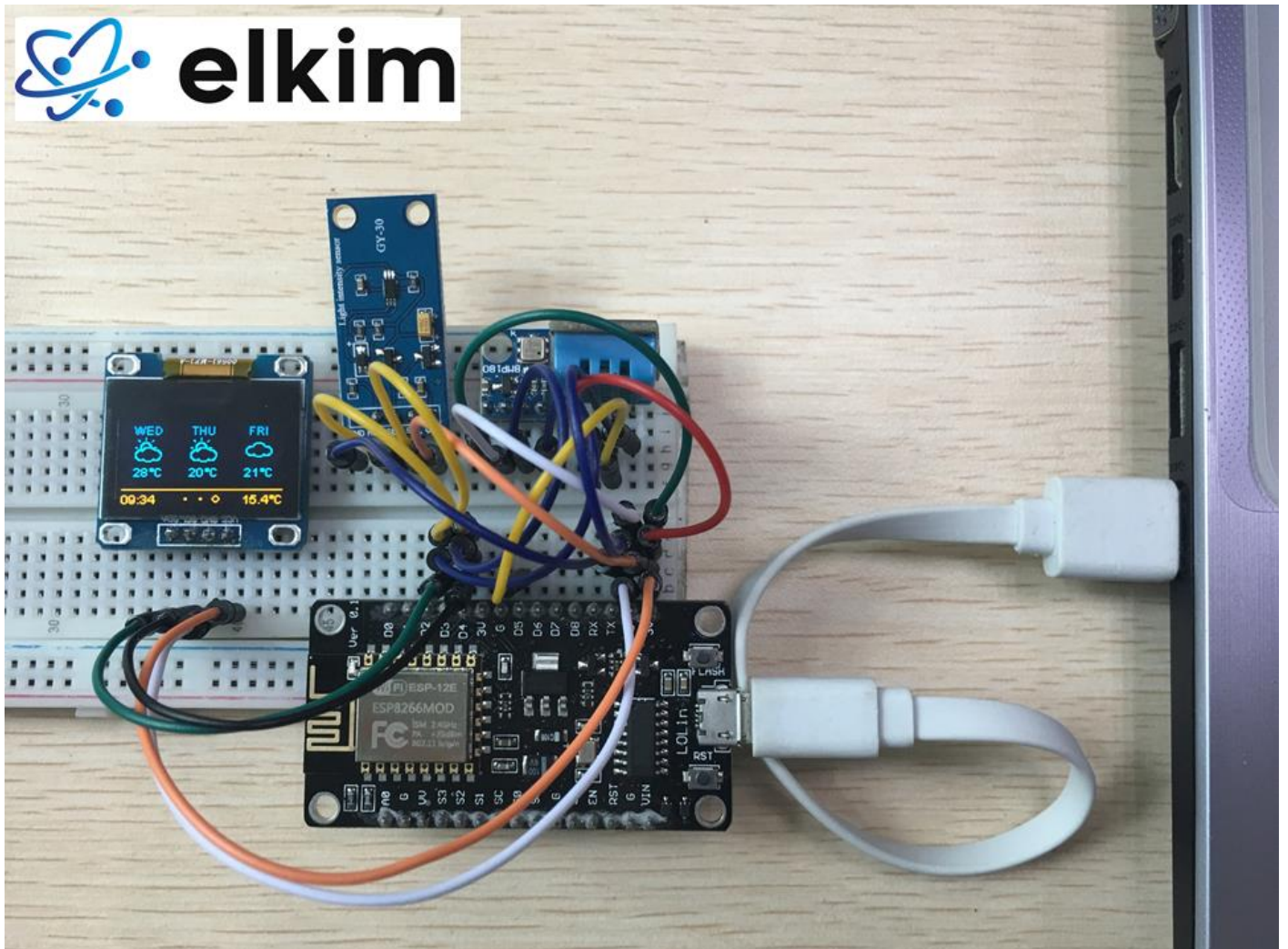


2) Benytt ESP8266Flasher.exe Burning Ai-Thinker_ESP8266_DOUT_8Mbit_v1.5.4.1-a_20171130.bin

til ESP8266  ESP8266Flasher.exe  Ai-Thinker_ESP8266_DOUT_8Mbit_v1.5.4.1-a_20171130.bin

3) Som vist:



STEG 4: KOBLE TIL KOMPONENTER

1) Tilkoblinger

ESP8266-12E	OLED
3.3V	VCC
GND	GND
D3	SDA
D4	SCL

ESP8266-12E	DHT11
3.3V	VCC
GND	GND
D5	DATA

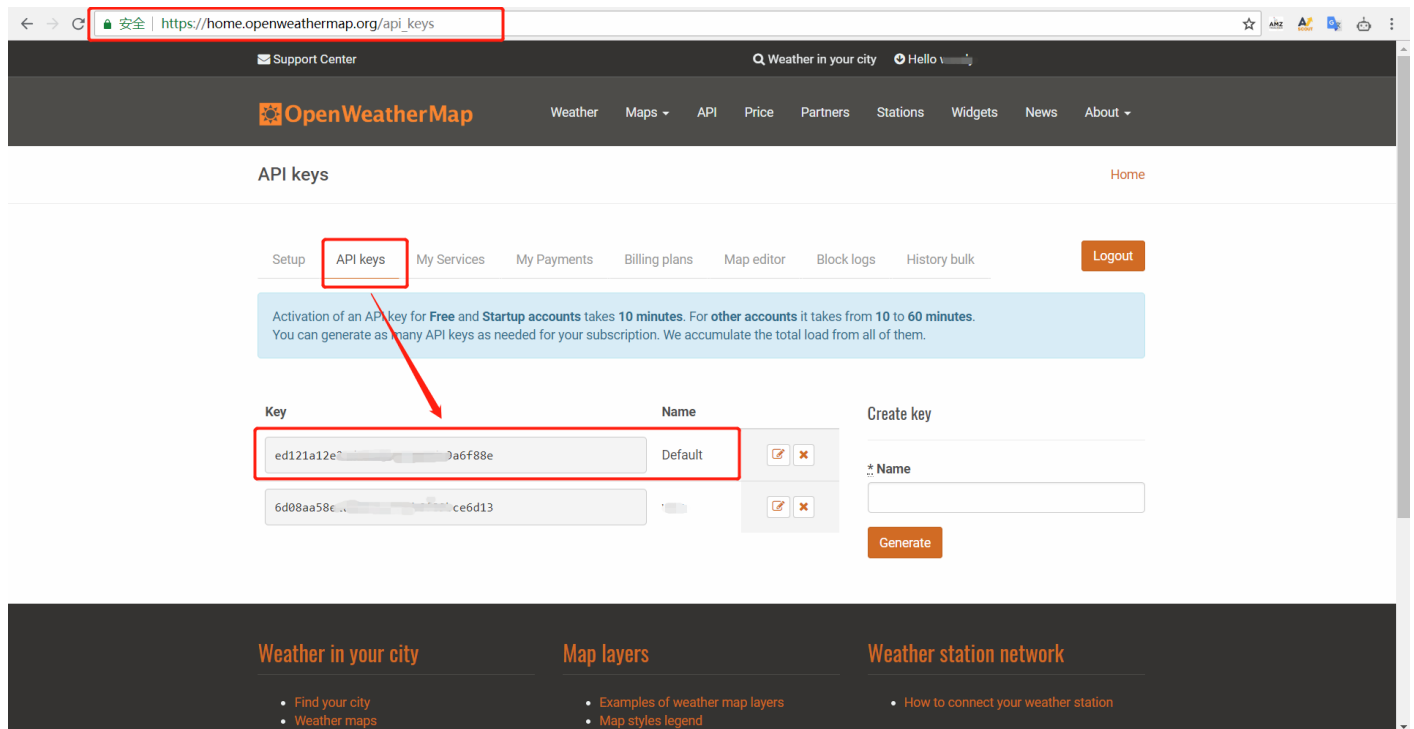
ESP8266-12E	BH1750FVI
3.3V	VCC
GND	GND
D3	SDA
D4	SCL

ESP8266-12E	BMP180
3.3V	VCC
GND	GND
D3	SDA
D4	SCL

ESP8266-12E	Computer
Micro-USB	USB

STEG 5: REGISTRER OPENWEATHERMAP, THINGSPEAK NY KONTO

1. Opprett en ny konto på https://home.openweathermap.org/users/sign_up for å få API-nøklene til openweathermap:



API keys

Setup **API keys** My Services My Payments Billing plans Map editor Block logs History bulk Logout

Activation of an API key for **Free** and **Startup** accounts takes **10 minutes**. For other accounts it takes from **10 to 60 minutes**. You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name
ed121a12e...a6f88e	Default
6d08aa58e...ce6d13	

Create key

* Name

Generate

Weather in your city Map layers Weather station network

- Find your city
- Weather maps
- Examples of weather map layers
- Map styles legend
- How to connect your weather station

1) Opprett ny konto hos <https://thingspeak.com>

A) Opprett en ny kanal og motta temperatur-, luftfuktighets-, lys- og atmosfærisk data fra ESP8266-12E (værstasjonen).

Channel Settings - Thin x

安全 | https://thingspeak.com/channels/549976/edit

ThingSpeak™ Channels Apps Community Support Commercial Use How to Buy Account Sign Out

ESP8266-DHT11

Channel ID: 549976 | ESP8266 and DHT11
Author: ideaspark
Access: Public

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Channel Settings

Percentage complete 50%

Channel ID 549976

Name ESP8266-DHT11

Description ESP8266 and DHT11

Field 1	Temperature	<input checked="" type="checkbox"/>
Field 2	Humidity	<input checked="" type="checkbox"/>
Field 3	Light	<input checked="" type="checkbox"/>
Field 4	Atmosphere	<input checked="" type="checkbox"/>
Field 5		<input type="checkbox"/>
Field 6		<input type="checkbox"/>

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
 - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.

B) Sett kanal til Public

安全 | https://thingspeak.com/channels/558971/sharing

ThingSpeak™ Channels Apps Community Support Commercial Use How to Buy Account Sign Out

WeatherStation

Channel ID: 558971
Author: ideaspark
Access: Private

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Channel Sharing Settings

Keep channel view private
 Share channel view with everyone
 Share channel view only with the following users:

Email Address

Help

ThingSpeak allows you to control who can view the data in your channel. Irrespective of the settings on this tab, reading data from or writing data to the fields of a channel requires the appropriate API key for the channel.

Channel Sharing Settings

- Keep channel view private:** Selecting this option keeps your channel private. Only you will be able to see the channel view.
- Share channel view with everyone:** Selecting this option makes the public view of your channel viewable by anyone browsing the ThingSpeak website.
- Share channel view only with the following users:** Selecting this option shares the private view of your channel only with specific ThingSpeak users.

C) Få tak i API_keys for kanalen.

← → ↻ 安全 | https://thingspeak.com/channels/558971/api_keys

ThingSpeak™ Channels Apps Community Support Commercial Use How to Buy Account Sign Out

WeatherStation

Channel ID: 558971
Author: ideaspark
Access: Public

Private View Public View Channel Settings Sharing **API Keys** Data Import / Export

Write API Key

Key

[Generate New Write API Key](#)

Read API Keys

Key

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click [Generate New Write API Key](#).
- Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click [Generate New Read API Key](#) to generate an additional read key for the channel.
- Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

[Update a Channel Feed](#)

D) Senere kan du følge med på opplastet data.

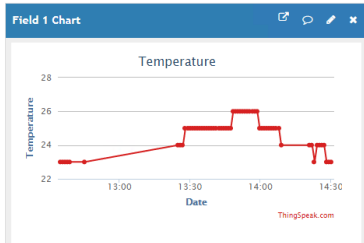
ESP8266-DHT11 - Thin x

← → ↻ 安全 | https://thingspeak.com/channels/549976/private_show

ThingSpeak™ Channels Apps Community Support Commercial Use How to Buy Account Sign Out

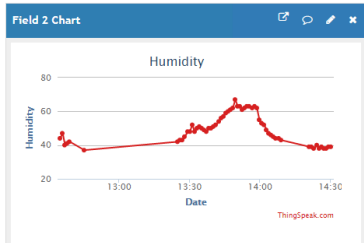
Created: [about a month ago](#)
Updated: [a day ago](#)
Last entry: [about a minute ago](#)
Entries: 1019

Field 1 Chart



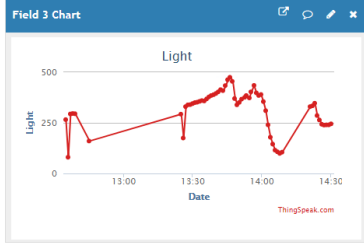
Temperature

Field 2 Chart



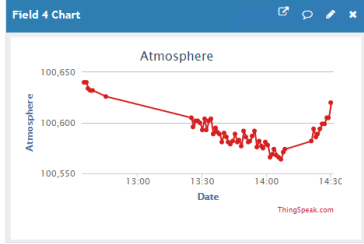
Humidity

Field 3 Chart



Light

Field 4 Chart



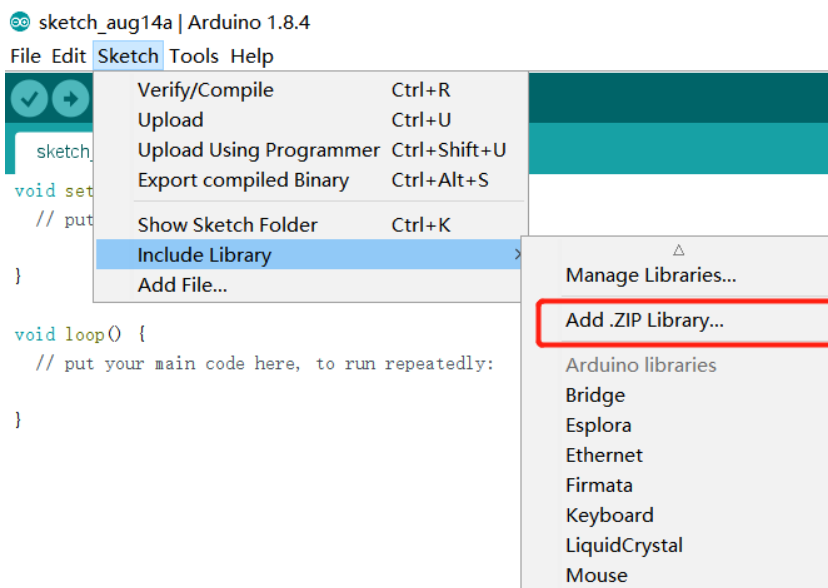
Atmosphere

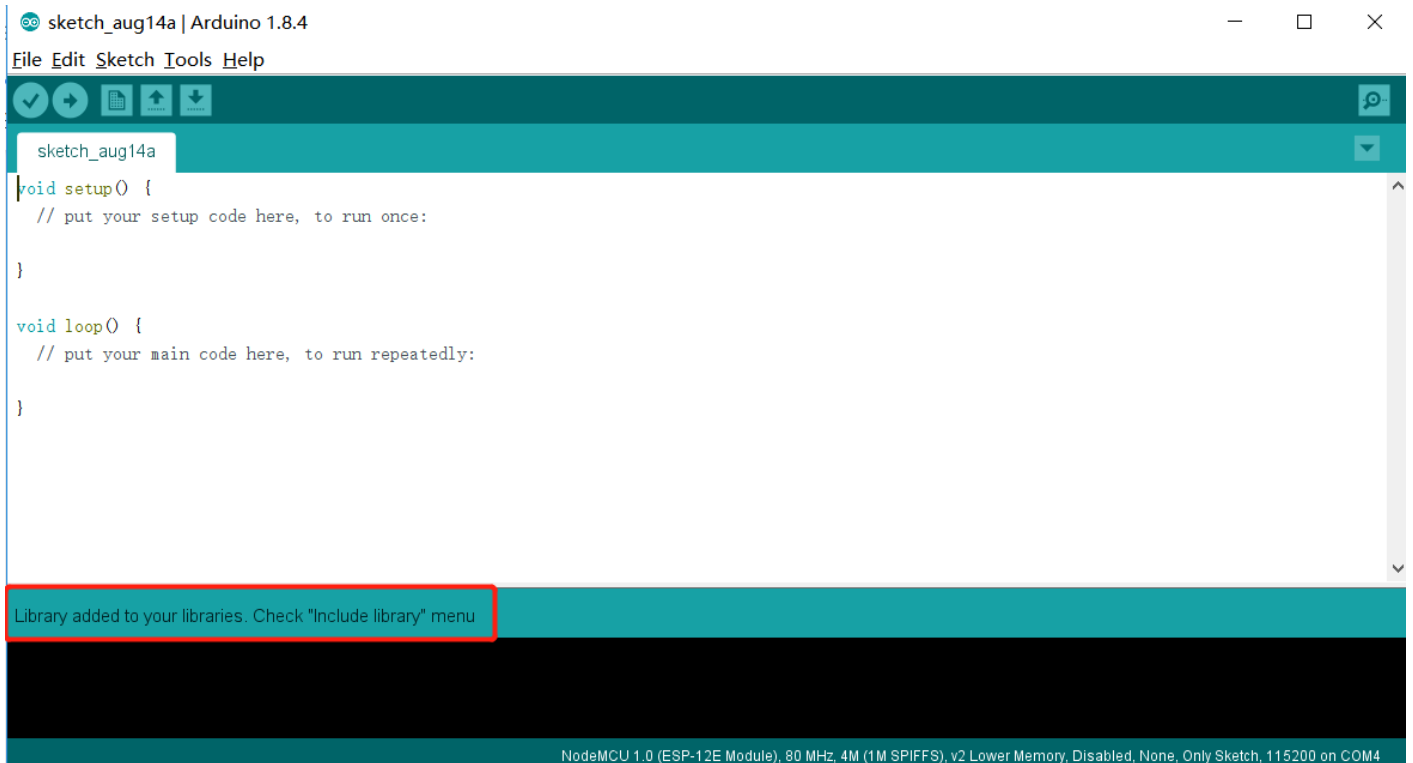
STEG 6: IMPORTER VÆRSTASJONSKODEN I ARDUINO IDE

1) Importer \Weather Station\Arduino IDE for ESP8266 Code\esp8266-weather-station-master.zip

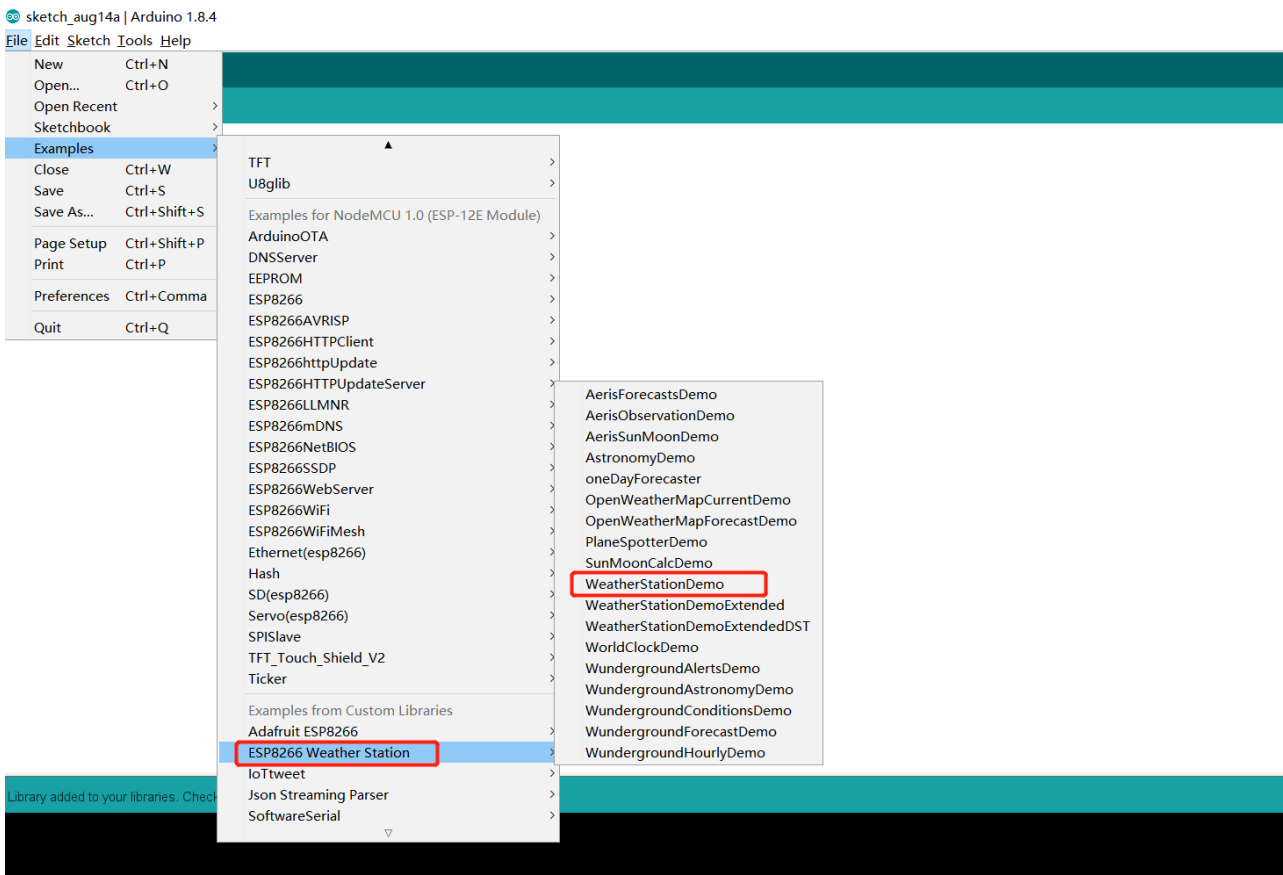


esp8266-weath
er-station-maste

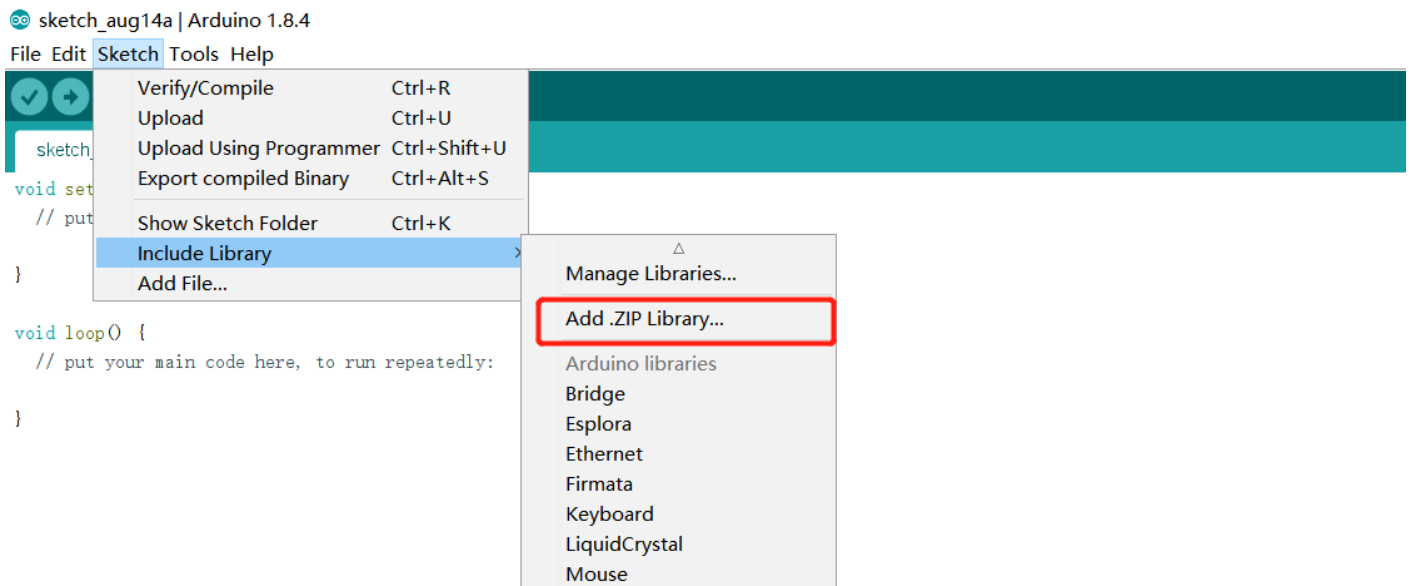




2) View Demo code :



STEG 7: LEGG TIL LIBRARY / BIBLIOTEK



1) Legg til \\Weather Station\Arduino IDE for ESP8266 Library\json-streaming-parser-master.zip



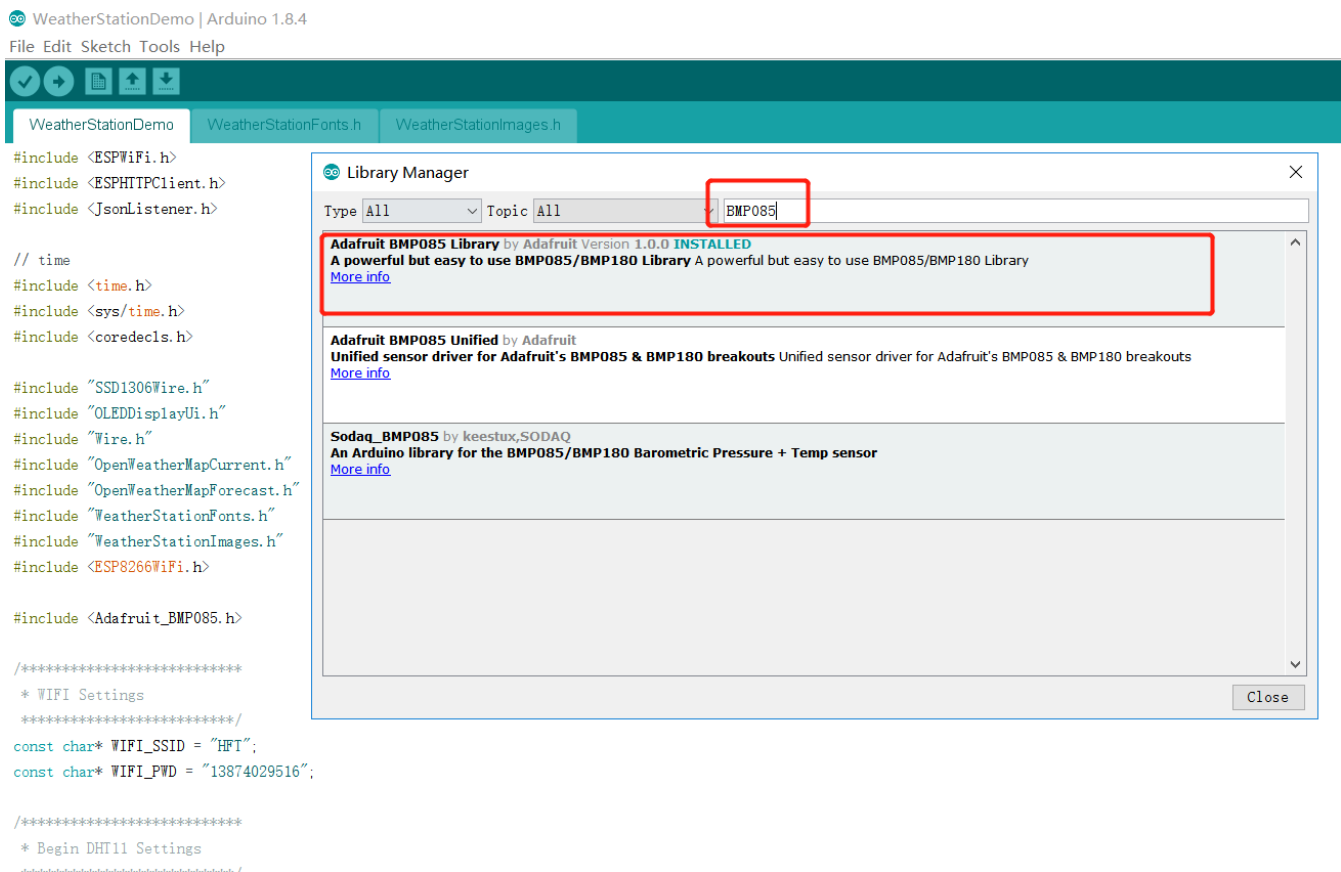
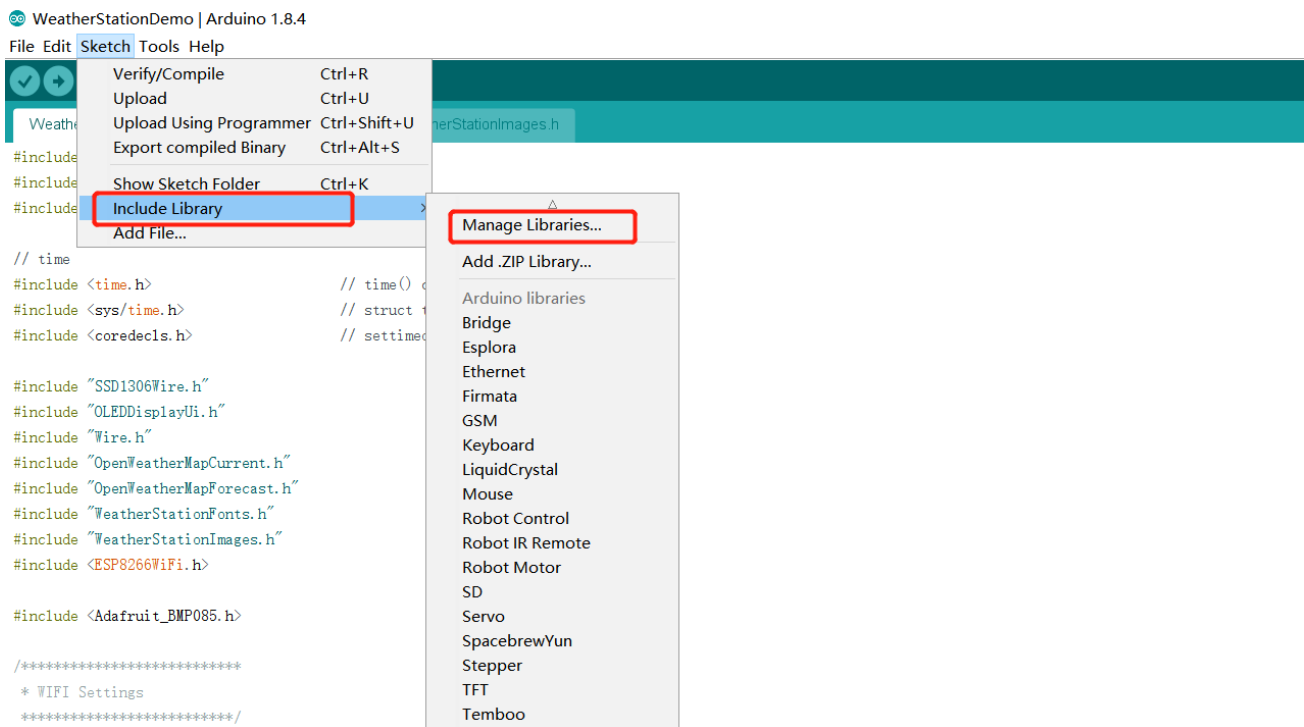
json-streaming-
parser-master.zip

2) Legg til \\Weather Station\Arduino IDE for ESP8266 Library\esp8266-oled-ssd1306-master.zip



esp8266-oled-s
sd1306-master.zip

3) Legg til Library Adafruit_BMP085

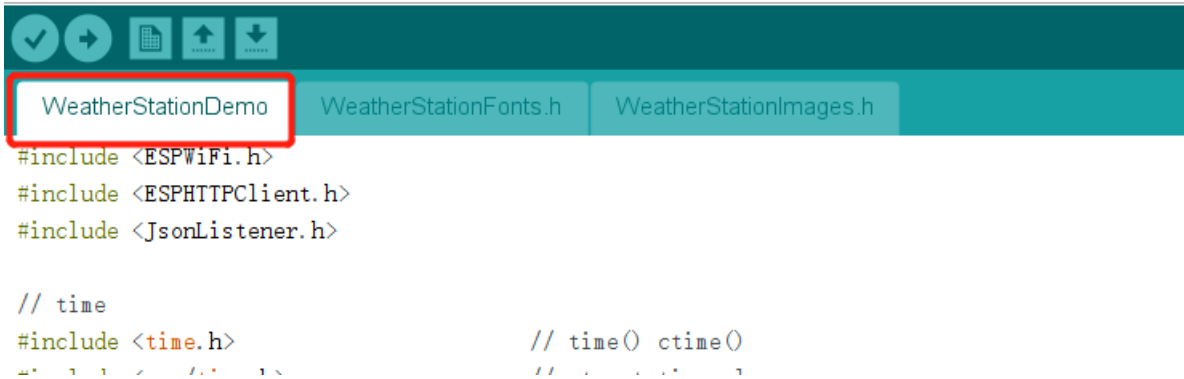


STEG 8: MODIFISER WEATHERSTATION.INO FILEN

1) Åpne WeatherStation.ino

WeatherStationDemo | Arduino 1.8.4

File Edit Sketch Tools Help



```

#include <ESPWiFi.h>
#include <ESPHTTPClient.h>
#include <JsonListener.h>

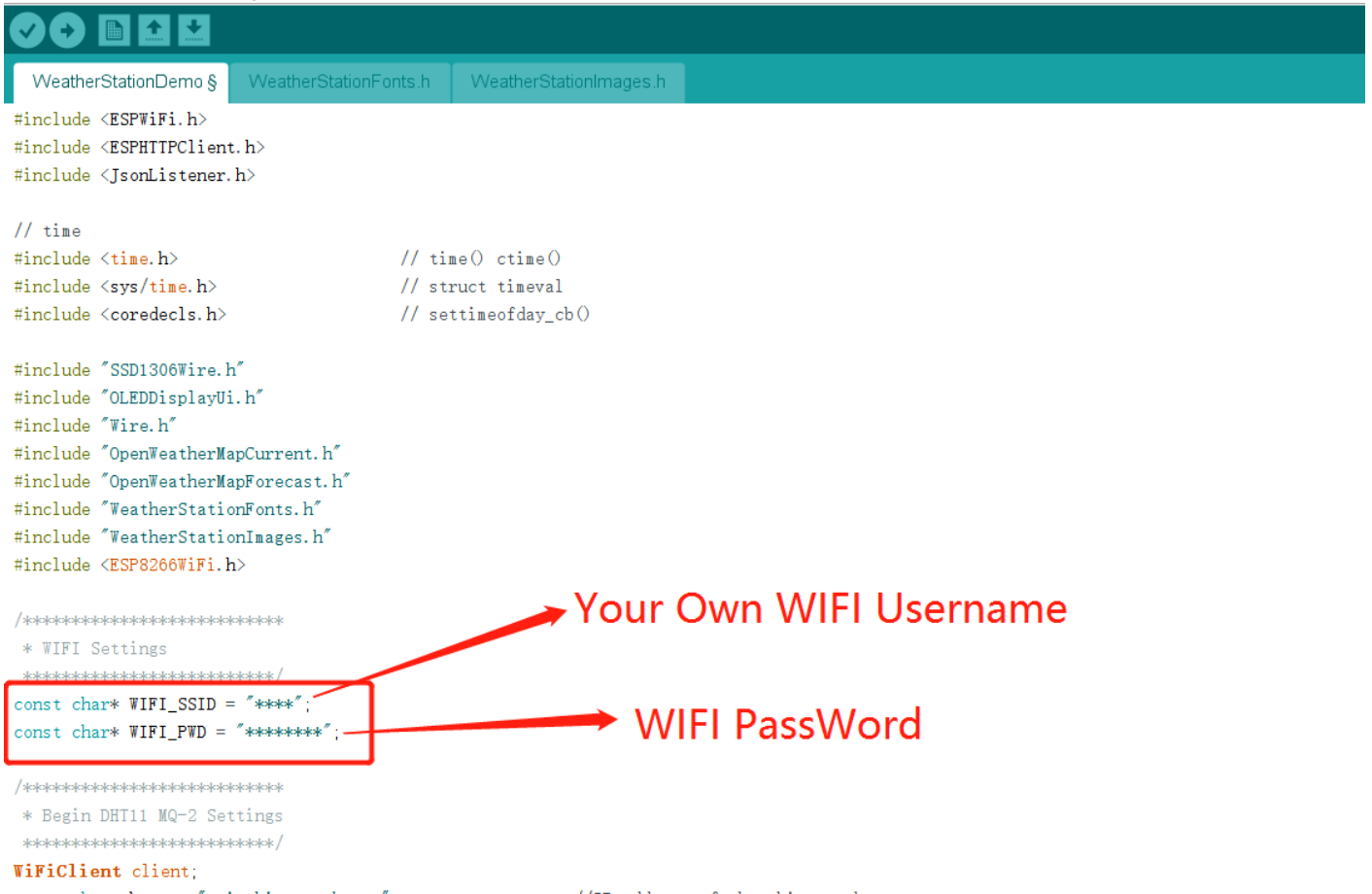
// time
#include <time.h> // time() ctime()

```

2) Legg inn ditt WiFi brukernavn og passord:

WeatherStationDemo | Arduino 1.8.4

File Edit Sketch Tools Help



```

#include <ESPWiFi.h>
#include <ESPHTTPClient.h>
#include <JsonListener.h>

// time
#include <time.h> // time() ctime()
#include <sys/time.h> // struct timeval
#include <coredecls.h> // settimeofday_cb()

#include "SSD1306Wire.h"
#include "OLEDDisplayUi.h"
#include "Wire.h"
#include "OpenWeatherMapCurrent.h"
#include "OpenWeatherMapForecast.h"
#include "WeatherStationFonts.h"
#include "WeatherStationImages.h"
#include <ESP8266WiFi.h>

/*****
 * WIFI Settings
 *****/
const char* WIFI_SSID = "****";
const char* WIFI_PWD = "*****";

/*****
 * Begin DHT11 MQ-2 Settings
 *****/
WiFiClient client;

```

3) Oppdater API_Keys for OpenWeatherMap.org

WeatherStationDemo | Arduino 1.8.4

File Edit Sketch Tools Help

```
WeatherStationDemo § WeatherStationFonts.h WeatherStationImages.h

const int SDA_PIN = GPIO5;
const int SDC_PIN = GPIO4
#endif

// OpenWeatherMap Settings
// Sign up here to get an API key:
const boolean IS_METRIC = true;
String OPEN_WEATHER_MAP_APP_ID = "*****";
String OPEN_WEATHER_MAP_LOCATION = "Zurich,CH";

// Pick a language code from this list:
// Arabic - ar, Bulgarian - bg, Catalan - ca, Czech - cz, German - de, Greek - el,
// English - en, Persian (Farsi) - fa, Finnish - fi, French - fr, Galician - gl,
// Croatian - hr, Hungarian - hu, Italian - it, Japanese - ja, Korean - kr,
// Latvian - la, Lithuanian - lt, Macedonian - mk, Dutch - nl, Polish - pl,
// Portuguese - pt, Romanian - ro, Russian - ru, Swedish - se, Slovak - sk,
// Slovenian - sl, Spanish - es, Turkish - tr, Ukrainian - ua, Vietnamese - vi,
// Chinese Simplified - zh_cn, Chinese Traditional - zh_tw.

String OPEN_WEATHER_MAP_LANGUAGE = "en";
const uint8_t MAX_FORECASTS = 4;

// Adjust according to your language
const String WDAY_NAMES[] = {"SUN", "MON", "TUE", "WED", "THU", "FRI", "SAT"};
const String MONTH_NAMES[] = {"JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL", "AUG", "SEP", "OCT", "NOV", "DEC"};

/*****
 * End Settings
 *****/
// Initialize the oled display for address 0x3c
SSD1306Wire display(I2C_DISPLAY_ADDRESS, SDA_PIN, SDC_PIN);
```

Your Own
OpenWeatherMap API_Keys

4) Oppdater API_Keys for thingspeak.com

WeatherStationDemo | Arduino 1.8.4

File Edit Sketch Tools Help

```
WeatherStationDemo $ WeatherStationFonts.h WeatherStationImages.h
#include "WeatherStationImages.h"
#include <ESP8266WiFi.h>

/*****
 * WIFI Settings
 *****/
const char* WIFI_SSID = "*****";
const char* WIFI_PWD = "*****";

/*****
 * Begin DHT11 MQ-2 Settings
 *****/
WiFiClient client;
const char *host = "api.thingspeak.com"; //IP address of the thingspeak server
const char *api_key = "*****"; //Your own thingspeak api_key
const int httpPort = 80;
#define pin 14 // ESP8266-12E D5 read emperature and Humidity data
#define gasPin A0 // ESP8266-12E A0 read gas data
int temp = 0; //temperature
int humi = 0; //humidity
int gasData = 0; //gas
void readTemperatureHumidityGas();
void uploadTemperatureHumidityGas();
long readTime = 0;
long uploadTime = 0;
/*****
 * Begin Settings
 *****/
#define TZ 2 // (utc+) TZ in hours
#define DST_MN 60 // use 60mn for summer time in some countries

// Setup
```

Your Own thingspeak API_Keys



const char *api_key = "*****";

//IP address of the thingspeak server
//Your own thingspeak api_key

STEG 9: SETT OPP BOARD OG PORT IGJEN

WeatherStationDemo | Arduino 1.8.4

File Edit Sketch Tools Help

WeatherStation

```
#include <ESPWi
#include <ESPHT
#include <JsonL

// time
#include <time.
#include <sys/t
#include <cored

#include "SSD13
#include "OLEDD
#include "Wire.
#include "OpenW
#include "OpenW
#include "Weath
#include "Weath
#include <ESP82
```

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Serial Monitor Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

WiFi101 Firmware Updater

Board: "NodeMCU 1.0 (ESP-12E Module)"

Flash Size: "4M (1M SPIFFS)"

Debug port: "Disabled"

Debug Level: "None"

lwIP Variant: "v2 Lower Memory"

CPU Frequency: "80 MHz"

Upload Speed: "115200"

Erase Flash: "Only Sketch"

Port: "COM4"

Get Board Info

Programmer: "AVRISP mkII"

Burn Bootloader

Arduino Robot Motor

Arduino Gemma

Adafruit Circuit Playground

Arduino Yún Mini

Arduino Industrial 101

Linino One

Arduino Uno WiFi

ESP8266 Modules

Generic ESP8266 Module

Generic ESP8285 Module

ESPduino (ESP-13 Module)

Adafruit Feather HUZZAH ESP8266

ESPRESSO Lite 1.0

ESPRESSO Lite 2.0

Phoenix 1.0

Phoenix 2.0

NodeMCU 0.9 (ESP-12 Module)

NodeMCU 1.0 (ESP-12E Module)

Olimex MOD-WIFI-ESP8266(-DEV)

SparkFun ESP8266 Thing

SparkFun ESP8266 Thing Dev

```

/*****
 * WIFI Settings
 *****/
const char* WIFI_SSID = "HFT";
const char* WIFI_PWD = "13874029516";

/*****
 * Begin DHT11 MQ-2 Settings
 *****/
```

WeatherStationDemo | Arduino 1.8.4

File Edit Sketch Tools Help

WeatherStation

```
#include <ESPWi
#include <ESPHT
#include <JsonL

// time
#include <time.
#include <sys/t
#include <cored

#include "SSD13
#include "OLEDD
#include "Wire.
#include "OpenW
#include "OpenW
#include "Weath
#include "Weath
#include <ESP82
```

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Serial Monitor Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

WiFi101 Firmware Updater

Board: "NodeMCU 1.0 (ESP-12E Module)"

Flash Size: "4M (1M SPIFFS)"

Debug port: "Disabled"

Debug Level: "None"

lwIP Variant: "v2 Lower Memory"

CPU Frequency: "80 MHz"

Upload Speed: "115200"

Erase Flash: "Only Sketch"

Port: "COM4"

Get Board Info

Programmer: "AVRISP mkII"

Burn Bootloader

Serial ports

COM4

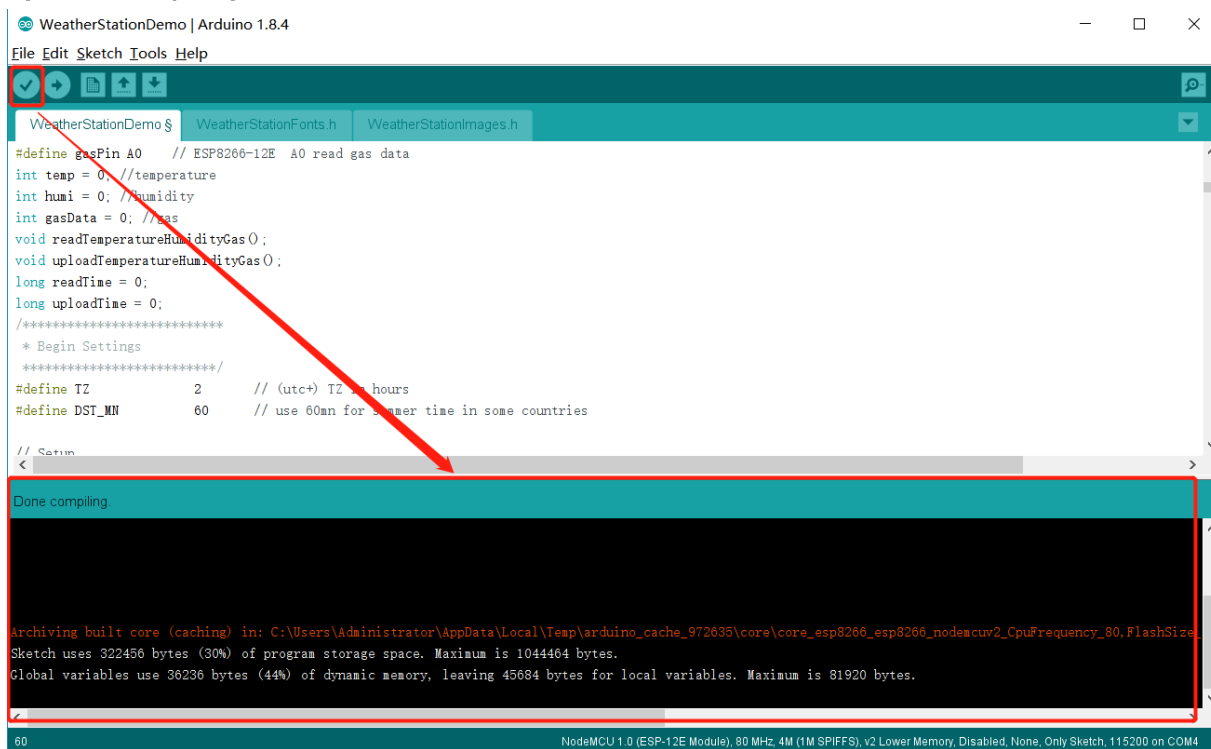
```

/*****
 * WIFI Settings
 *****/
const char* WIFI_SSID = "HFT";
const char* WIFI_PWD = "13874029516";
```

13 NodeMCU 1.0 (ESP-12E Module), 80 MHz, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM4

STEG 10: OVERFØR KODE TIL ESP8266

1) Verifiser prosjektet



```

WeatherStationDemo | Arduino 1.8.4
File Edit Sketch Tools Help
WeatherStationDemo $ WeatherStationFonts.h WeatherStationImages.h
#define gasPin A0 // ESP8266-12E A0 read gas data
int temp = 0; //temperature
int humi = 0; //humidity
int gasData = 0; //gas
void readTemperatureHumidityGas();
void uploadTemperatureHumidityGas();
long readTime = 0;
long uploadTime = 0;
/*****
 * Begin Settings
 *****/
#define TZ 2 // (utc+) TZ in hours
#define DST_MN 60 // use 60mn for summer time in some countries

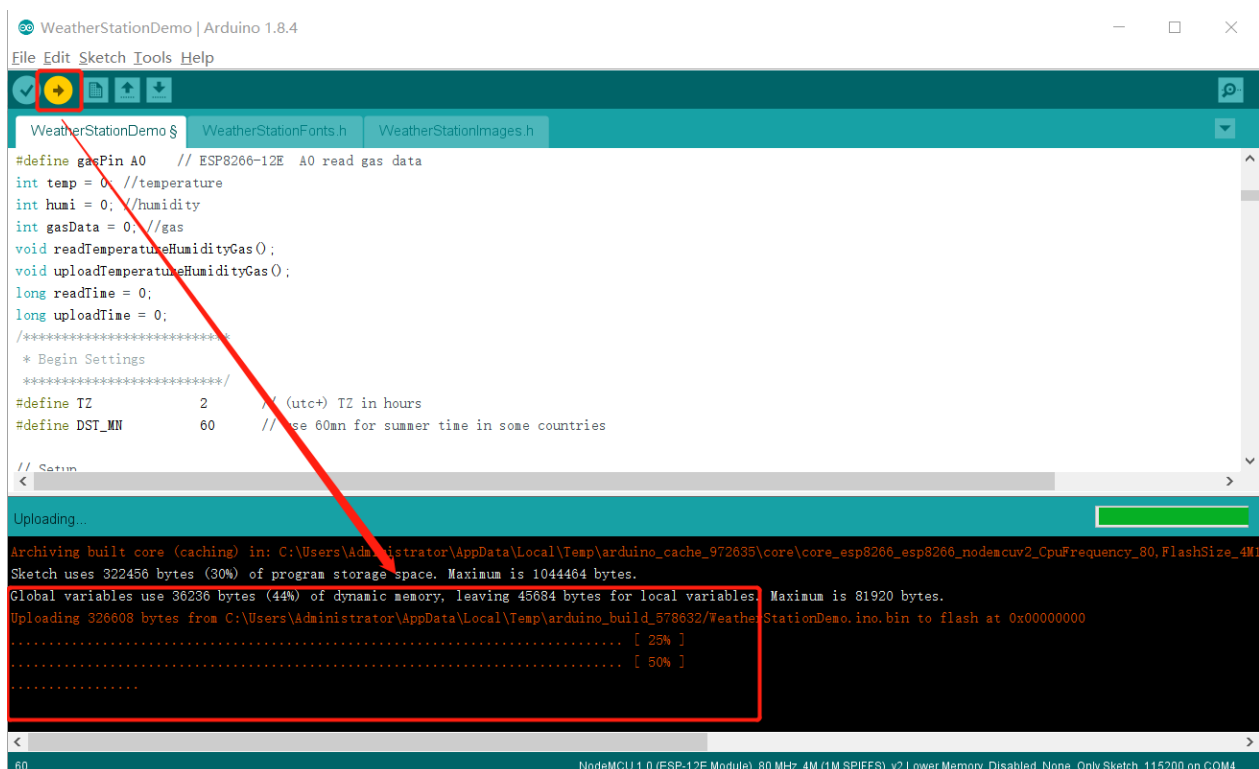
// Setup
<

Done compiling.

Archiving built core (caching) in: C:\Users\Administrator\AppData\Local\Temp\arduino_cache_972635\core\core_esp8266_esp8266_nodecu2_CpuFrequency_80_FlashSize_4M
Sketch uses 322456 bytes (30%) of program storage space. Maximum is 1044464 bytes.
Global variables use 36236 bytes (44%) of dynamic memory, leaving 45684 bytes for local variables. Maximum is 81920 bytes.

60 NodeMCU 1.0 (ESP-12E Module), 80 MHz, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM4
  
```

2) Overfør prosjektet til ESP8266



```

WeatherStationDemo | Arduino 1.8.4
File Edit Sketch Tools Help
WeatherStationDemo $ WeatherStationFonts.h WeatherStationImages.h
#define gasPin A0 // ESP8266-12E A0 read gas data
int temp = 0; //temperature
int humi = 0; //humidity
int gasData = 0; //gas
void readTemperatureHumidityGas();
void uploadTemperatureHumidityGas();
long readTime = 0;
long uploadTime = 0;
/*****
 * Begin Settings
 *****/
#define TZ 2 // (utc+) TZ in hours
#define DST_MN 60 // use 60mn for summer time in some countries

// Setup
<

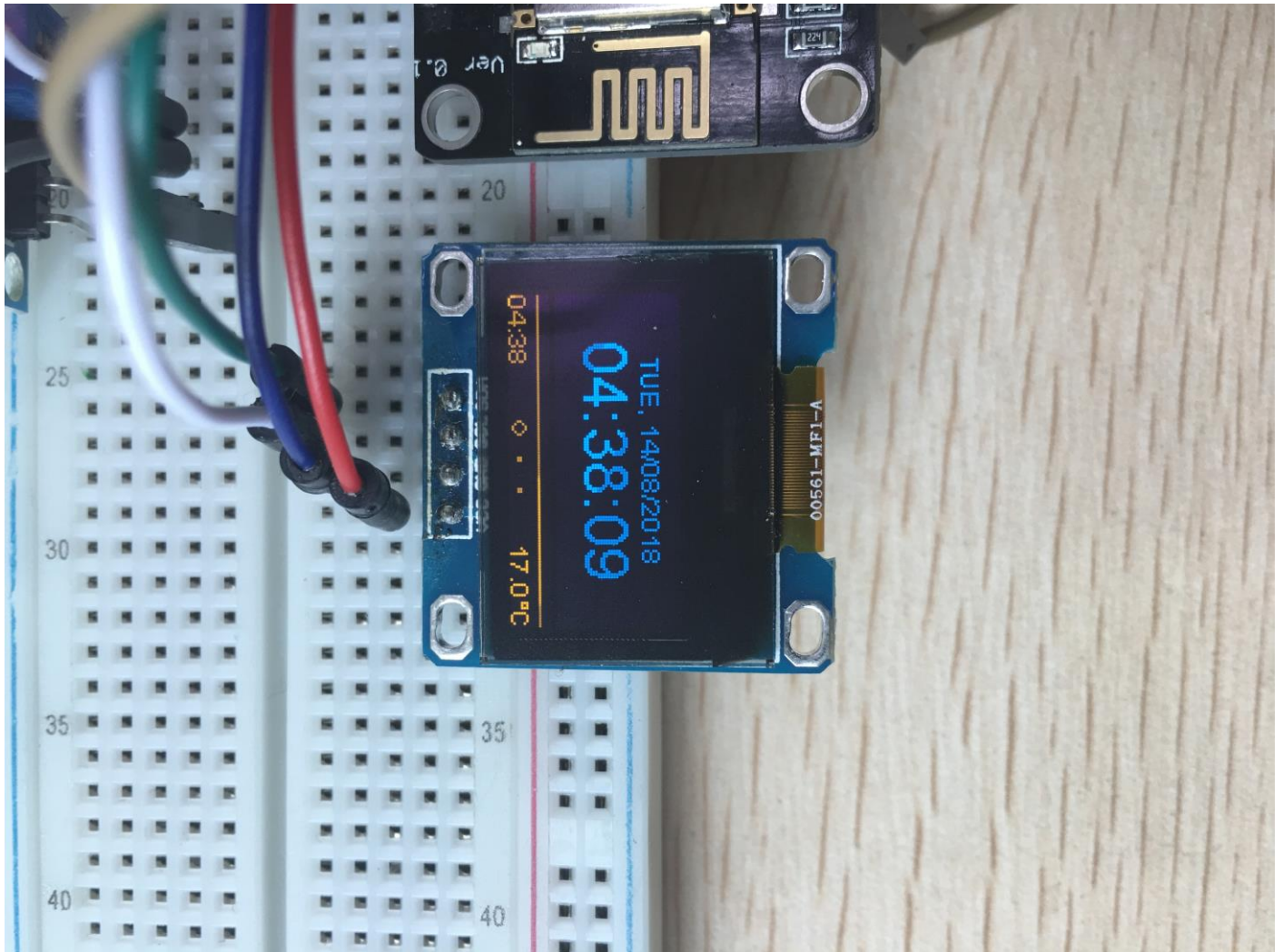
Uploading...

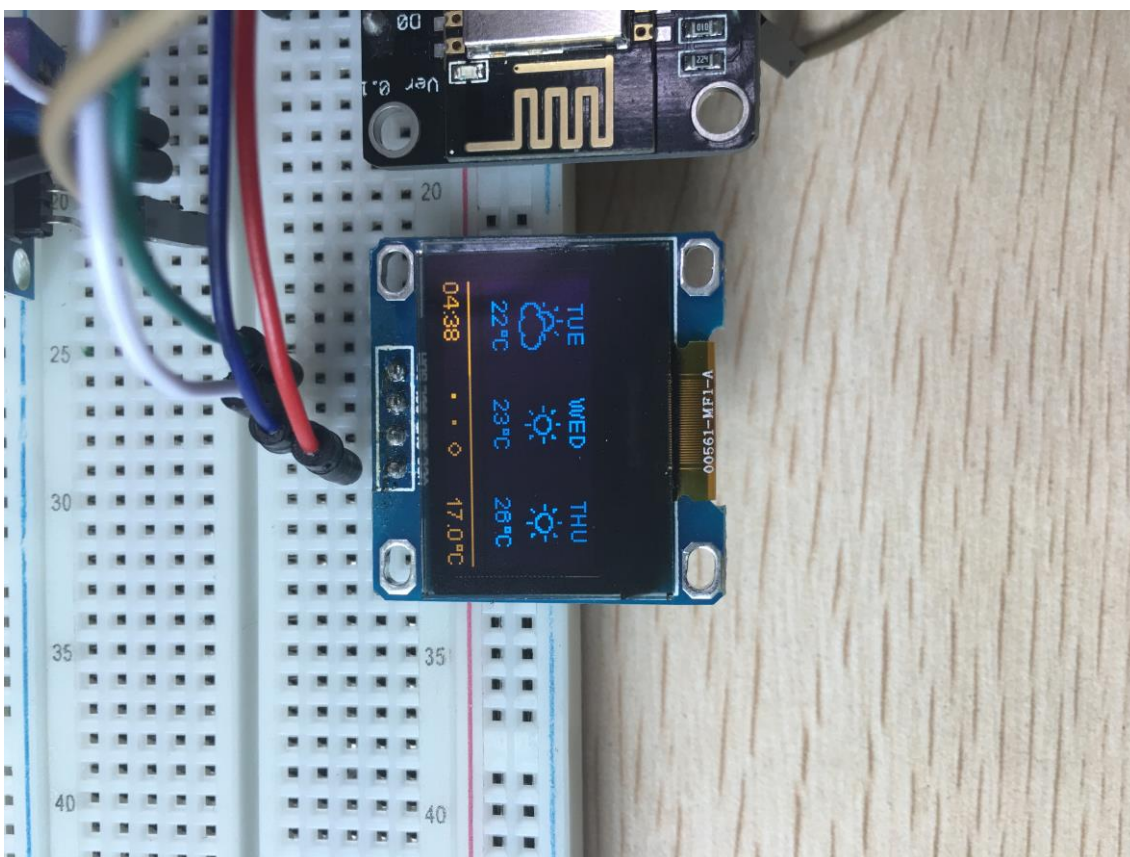
Archiving built core (caching) in: C:\Users\Administrator\AppData\Local\Temp\arduino_cache_972635\core\core_esp8266_esp8266_nodecu2_CpuFrequency_80_FlashSize_4M
Sketch uses 322456 bytes (30%) of program storage space. Maximum is 1044464 bytes.
Global variables use 36236 bytes (44%) of dynamic memory, leaving 45684 bytes for local variables. Maximum is 81920 bytes.
Uploading 326608 bytes from C:\Users\Administrator\AppData\Local\Temp\arduino_build_578632\WeatherStationDemo.ino.bin to flash at 0x00000000
..... [ 25% ]
..... [ 50% ]
.....

60 NodeMCU 1.0 (ESP-12E Module), 80 MHz, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM4
  
```

STEG 11: RESULTAT

1) OLED Display informasjon





2) Serial Monitor Data

WeatherStationDemo | Arduino 1.8.4
File Edit Sketch Tools Help

```

#include <ESPWiFi.h>
#include <ESP8266WiFi.h>
#include <JsonListener.h>

// time
#include <time.h>
#include <sys/time.h>
#include <coredecls.h>

#include "SSD1306Wire.h"
#include "OLEDDisplayUi.h"
#include "Wire.h"
#include "OpenWeatherMapCurrent.h"
#include "OpenWeatherMapForecast.h"
#include "WeatherStationFonts.h"
#include "WeatherStationImages.h"
#include <ESP8266WiFi.h>

#include <Adafruit_BMP085.h>

/*****
 * WIFI Settings
 *****/
const char* WIFI_SSID = "HFT";
  
```

```

light: 106
Pressure = 100567 Pascal
temp:24 humi:42
light: 99
Pressure = 100561 Pascal
temp:24 humi:42
light: 192
Pressure = 100566 Pascal
temp:24 humi:42
light: 195
Pressure = 100564 Pascal
temp:24 humi:42
light: 210
Pressure = 100561 Pascal
temp:24 humi:44
light: 207
Pressure = 100568 Pascal
temp:24 humi:45
light: 205
Pressure = 100569 Pascal
  
```

Done Saving

3) thinkspeak.com Data

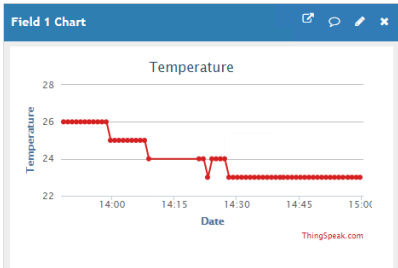
HT11 - Thin x

安全 | https://thingspeak.com/channels/549976/private_show

ThingSpeak™ Channels Apps Community Support Commercial Use How to Buy Account Sign Out

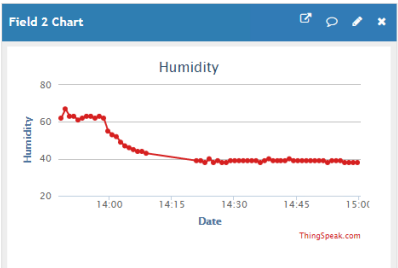
Created: [about a month ago](#)
Updated: [a day ago](#)
Last entry: [31 minutes ago](#)
Entries: 1019

Field 1 Chart



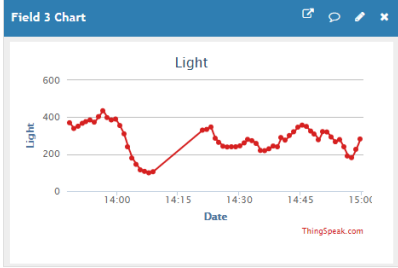
Temperature

Field 2 Chart



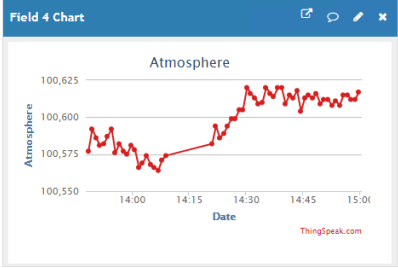
Humidity

Field 3 Chart



Light

Field 4 Chart



Atmosphere