



Individual Project

IT essentials 1ITF

Claes Jonas 1ITF-09
r0841639

CAMPUS

Geel

Technology

Electronics-ICT / Applied informatics

IT essentials

Course unit: IT essentials

Educational activity: IT essentials

First tier



Academiejaar 2020-2021

Inhoudstabel

Content

| | |
|--|-----------|
| Inhoudstabel..... | 3 |
| Lijst met illustraties..... | 4 |
| Lijst met afkortingen en symbolen..... | 5 |
| 1 Overzicht..... | 6 |
| 2 Hardware..... | 7 |
| 2.1 Schema..... | 7 |
| 3 Software..... | 8 |
| 3.1 Structuur..... | 8 |
| 3.2 Werking..... | 9 |
| 4 Conclusie..... | 10 |

Lijst met illustraties

Lijst met afkortingen en symbolen

| |
|--|
| |
|--|

1 Overzicht

Voor dit project heb ik gekozen om zowel mijn code als mijn documentatie op GitHub (https://github.com/jonasclaes/TM_ITEssentials_WeatherStation) te posten. Ik heb dit gedaan om toch wat extra toe te voegen aan mijn project.

Daarnaast heb ik ook gebruik gemaakt van objectgeoriënteerd programmeren. Zo heb ik mijn code in verschillende "bouwstenen" kunnen opdelen, en kan ik heel overzichtelijk programmeren.

Iedere sensor heeft bijvoorbeeld zijn eigen klasse met daarin enkele standaard functies zoals "setup()" en "loop()". Deze 2 functies zijn uiteraard ook gedefinieerd als een abstracte klasse die geïmplementeerd wordt in de sensor klassen.

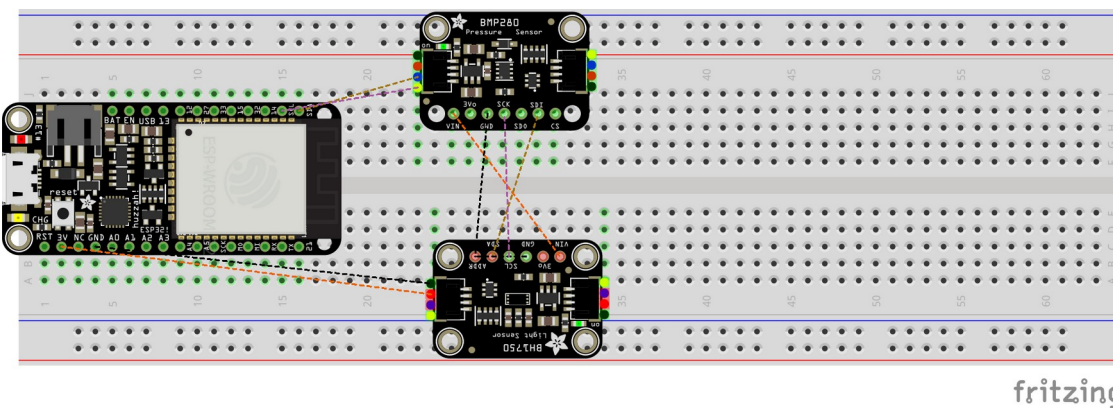
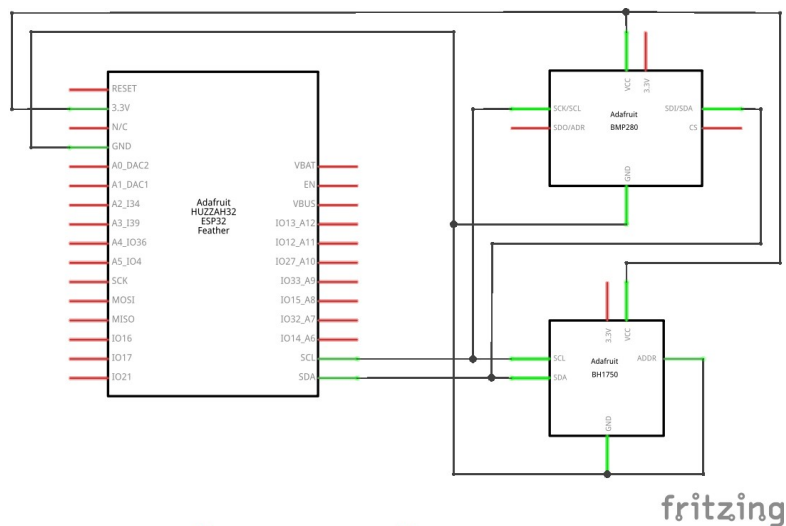
Ook het wireless als het MQTT gedeelte is opgedeeld in klassen, zodat ik uiteindelijk in de "main.cpp" file maar 2 functies moet oproepen.

Verder heb ik ook een configuratie bestand gemaakt, zodat als je een tweede station zou willen maken, je niet overal dingen moet gaan aanpassen, maar gewoon in 1 bestand de gegevens kan invullen van het draadloze netwerk en de ThingSpeak credentials.

2 Hardware

2.1 Schema

Ik heb het schema van de componenten gemaakt met Fritzing. Dit heb ik daarna opgebouwd op een breadboard. Hieronder vindt u het schema terug, alsook een overzicht van het breadboard.



3 Software

3.1 Structuur

Ik zou hieronder kort de opbouw willen toelichten die ik gehanteerd heb.

```
src/  
  main.cpp  
  App.h  
  App.cpp  
  config.h  
  mqtt/  
    WS_MQTT.h  
    WS_MQTT.cpp  
  sensor/  
    Sensor.h  
    BH_1750/  
      WSS_BH1750.h  
      WSS_BH1750.cpp  
    BMP280/  
      WSS_BMP280.h  
      WSS_BMP280.cpp  
  wireless/  
    WS_Wireless.h  
    WS_Wireless.cpp
```

Zoals je kan zien heb ik de software dus in verschillende stukken opgedeeld, elk onderdeel respectievelijk in zijn eigen map en bestand.

Het bestand "main.cpp" wordt altijd door de microcontroller uitgevoerd, hierin heb ik de functies "setup()" en "loop()". Ik doe hier niets anders in dan gewoon de "setup()" en "loop()" functies aanroepen van de App klasse.

In de App klasse gebeurt het echte werk, hier initialiseer ik de sensoren (BH1750 en BMP280), alsook de draadloze verbinding en MQTT.

Voor de klassen van de sensoren maak ik gebruik van "Sensor.h", dit is een abstracte klasse die enkel de functies "setup()" en "loop()" definieert, deze klasse wordt dan geïmplementeerd door de WSS_BH1750- en WSS_BMP280 klasse. Dit maakt dus dat ik simpel beide sensoren kan initialiseren op dezelfde manier en in de App klasse alles er proper en overzichtelijk uit ziet.

3.2 Werking

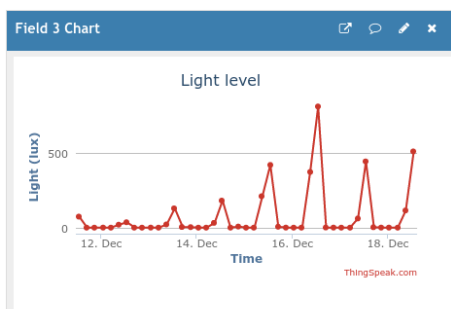
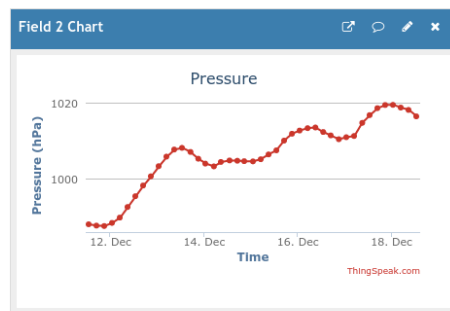
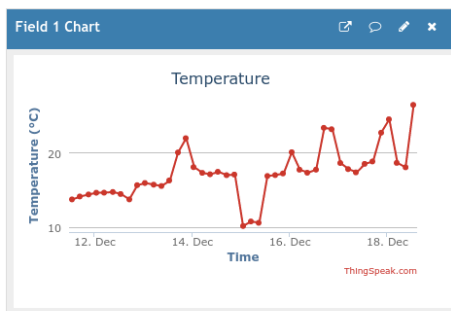
Het is natuurlijk ook geen verrassing dat dit projectje werkt. Zowel het uitlezen van de temperatuur als druk en lichtintensiteit lukt allemaal. Deze worden ook realtime getoond op ThingSpeak via de MQTT verbinding.

Deze data kan ook teruggekeken worden via het ThingSpeak platform.

Hieronder ziet u een overzicht van de voorbije 7 dagen van de metingen die de sensoren hebben gemaakt. De gegevens worden elke 30 minuten doorgestuurd (deepsleep). Om deze grafieken zo te krijgen, heb ik de timescale op 7 dagen gezet, met een interval van 240 minuten en 336 datapunten (ieder halfuur -> 48 metingen/dag -> 336 metingen/week).

Channel Stats

Created: 9 days ago
Last entry: about a minute ago
Entries: 1301



4 Conclusie

Ik vond het een zeer tof project om te doen. Ik heb mezelf kunnen uitdagen door alles objectgeoriënteerd te maken, alsook meteen de stap naar MQTT te doen. Wel had ik eerder al ervaring met de ESP32 doordat ik mijn eigen LED sturingen bouw. Dit heeft alles wel vlotter doen gaan. Ik heb hier alsnog uit geleerd en het is de eerste keer dat ik data visualiseer via een platform zoals ThingSpeak.

5 Bronnen en links

Project, GitHub, 2020-12-18,

(https://github.com/jonasclaes/TM_ITEssentials_WeatherStation)

Data en visualisatie, ThingSpeak, 2020-12-18,

(<https://thingspeak.com/channels/1252241>)

Video, YouTube, 2020-12-18,

(<https://youtu.be/UnOL4vxmAQA>)