Bronbeschrijving

# NFC Module

[https://wiki.dfrobot.com/](https://wiki.dfrobot.com/Gravity:%20I2C%20&%20UART%20NFC%20Module%20SKU:%20DFR0231-H)

Deze bron heb ik gebruikt voor het configureren en lezen van de NFC-module. Eerst moeten we de juiste library downloaden. Vervolgens kunnen we aan de slag.

Het was voor mij de eerste keer dat ik met deze sensor gewerkt heb en heb daarom eerst de voorbeeldcode uitgevoerd. Hierna werd al meteen duidelijk hoe deze sensor en zijn library werkte en kon ik dus op een eenvoudige manier de voorbeeldcode aanpassen naar mijn benodigdheden.

De eerste blok gebruik ik om te bepalen welke blok van de NFC-tag we gaan lezen en de grootte van hoeveel we van deze blok willen lezen.

Afbeelding met tekst, Lettertype, schermopname, nummer

Automatisch gegenereerde beschrijving

Vervolgens gebruiken we de volgende code voor het lezen van de NFC-module en door te sturen naar de MQTT-broker.

Afbeelding met tekst, schermopname, scherm, Lettertype

Automatisch gegenereerde beschrijving

# SCD4X

<https://learn.adafruit.com/adafruit-scd-40-and-scd-41/arduino>

Hier eigenlijk hetzelfde als bij de NFC-module. Het enige verschil bij de SCD4X sensor was dat ik een beetje harder moest zoeken naar juiste documentatie, aangezien er meerdere versies van deze sensor bestaan. Nadat ik de juiste documentatie had gevonden, was het weer zeer simpel om deze om te vormen naar wat ik nodig had.

We moeten voordat we iets doen opnieuw de juiste library downloaden. Hierna kunnen we beginnen doorgebruik te maken van onderstaande code voor het aanmaken van een SCD4X sensor object.



Hierna kunnen we volgende code gebruiken om het object klaar te maken om metingen te maken.

Afbeelding met tekst, schermopname

Automatisch gegenereerde beschrijving

Om dan ten slotte een functie te gebruiken om deze metingen op te vragen

Afbeelding met tekst, schermopname

Automatisch gegenereerde beschrijving

# Servo

<https://github.com/madhephaestus/ESP32Servo>

Het gebruiken van een servo op het esp32 platform is een beetje anders dan op het arduino platform. Hierbij gebruiken we namelijk een andere library. Vervolgens blijft de rest wel gewoon hetzelfde.

We maken een Servo object aan, vervolgens ‘attachen’ we deze op de juiste pin op de mcu om dan ten slotte met de write funtie de servo te laten bewegen.

Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving

# PubSubClient (MQTT)

<https://github.com/knolleary/pubsubclient>

Voor het MQTT-gedeelte wou ik eerst de library gebruiken waarmee we in de les al kennis hebben gemaakt ([async-mqtt-client](https://github.com/marvinroger/async-mqtt-client)). Het probleem hiermee was dat deze niet goed samenwerkte met de NFC library. Daarom heb ik gekozen voor deze library. De documentatie en de voorbeelden maakte het makkelijk om te leren hoe ik deze library moest implementeren.

Om te beginnen moet je altijd een PubSubClient object aanmaken en het WiFiClient object meegeven.



Vervolgens in de setup functie zetten we onze broker en port instellingen juist en stellen we in welke functie moet worden opgeroepen als callback. Deze callback functie zal worden opgeroepen en uitgevoerd, elke keer als er een update wordt gestuurd naar een van de gesubscribeerde topics.

Afbeelding met tekst, Lettertype, schermopname, Graphics

Automatisch gegenereerde beschrijving

Nadien gebruik ik volgende functie om te connecteren op eerder geconfigureerde MQTT-broker, ook subscriben we hier op de juiste topics

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Ten slotte gebruik ik de eerder gekozen callback functie om te bepalen wat er moet gebeuren als er een update naar een eerder gesubscribeerde topic gestuurd wordt.

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving