|  |  |
| --- | --- |
|  | ELEKTRONICA-ICT  Project Onderzoek 2019-2020 |

**MQTT Communicatie met ESP32**

|  |  |
| --- | --- |
| Auteurs  Product Owner | Kieran van Looveren  Vincent Claes |

Abstract

In dit onderzoek wordt er gezocht naar hoe men data kan versturen naar een database met MQTT via een esp 32. Hiermee zouden we een deel van een feedback loop maken in samenwerking met een AI en zo de automatisatie van een productielijn aansturen. Dit is geprogrammeerd met c-code in PlatformIO op Visual Studio Code. In dit onderzoek wordt er gebruik gemaakt van een esp-wrover-kit in combinatie met een temperatuur en vochtigheidsmeter.

Inhoudsopgave

[1 Introductie 2](#_Toc48551888)

[2 Materiaal en methode 2](#_Toc48551889)

[3 Resultaten 3](#_Toc48551890)

[4 Discussie 3](#_Toc48551891)

[5 Conclusie 4](#_Toc48551892)

# Introductie

Overal rondom ons is technologie steeds meer aanwezig. Zowel in wetenschappelijk onderzoek als in het dagelijks leven wordt er steeds meer en meer data verzameld, te veel om handmatige te behandelen. Door de constante groei van data groeit ook de belangrijke rol van artificiële intelligentie. Omdat AI-processen nog steeds intensief kunnen zijn is het niet altijd mogelijk om deze processen door elke chip of instrument uit te laten voeren. Ook wanneer men data van sensoren op verschillende locaties wil combineren is het belangrijk om deze data zo snel mogelijk uit te lezen en door te sturen.

In dit onderzoek wordt er gekeken naar zo een mogelijke situatie. In de opstelling wordt data dat men binnenkrijgt over het internet van andere sensoren naar een getrainde AI gestuurd. De informatie die de AI er dan uithaalt kan men dan terug te sturen zodat de productie mogelijk zich aanpast op problemen. Om dit proces te simuleren werd er gebruik gemaakt van een sensor verbonden aan een esp-32 board. Dit esp-32 board kon deze data doorsturen naar een onlinedatabase.

In het [eerste deel](#_Materiaal_en_methode) van dit project worden de verschillende componenten. Hier wordt ook de gebruikte methode en code besproken. In [Resultaten](#_Resultaten) worden de behaalde resultaten op gelijst, die vervolgens in [het derde deel](#_Discussie) besproken worden. Dit verslag wordt afgerond met een [Conclusie](#_Conclusie) in hoofdstuk 5.

# Materiaal en methode

Dit hoofdstuk wordt opgedeeld in twee delen. Eerst zal kort de gebruikte hardware besproken worden, gevolgd door een beschrijving van de software.

Om metingen te nemen wordt in dit onderzoek gebruik gemaakt van een DHT22 sensor. Deze sensor meet zowel temperatuur als de vochtigheidsgraad van de omgeving. Hier wordt er gebruik gemaakt van de esp-wrover-kit. Er is voor de esp-wrover-kit gekozen omdat deze een esp-32 heeft met de mogelijkheid om een sdkaart voor verschillende wifi connecties te maken. De kit heeft ook een ingebouwd lcd-scherm wat ons de mogelijk biedt om de data te tonen zonder dat men steeds naar een online dashboard moet gaan om de data te zien.

Dit project is geprogrammeerd in Visual Studio Code, gebruik makend van PlaformIO. Een extra extensie is ook toegevoegd aan Visual Studio Code om te werken in de espressif framework.



Fig. 1 Flowchart Programma

Zoals te zien is in de flowchart (Fig. 1) leest de esp-wrover-kit de data uit van de DHT22. Deze data worden dan via MQTT naar een online database gestuurd. Hier wordt deze data dan opgeslagen voor later gebruik. Dit onderdeel zorgt voor het verzamelen van de data in het volledig model, waarvan we de setup in figuur 2 zien. In deze flowchart zien we het systeem met het AI element toegevoegd.

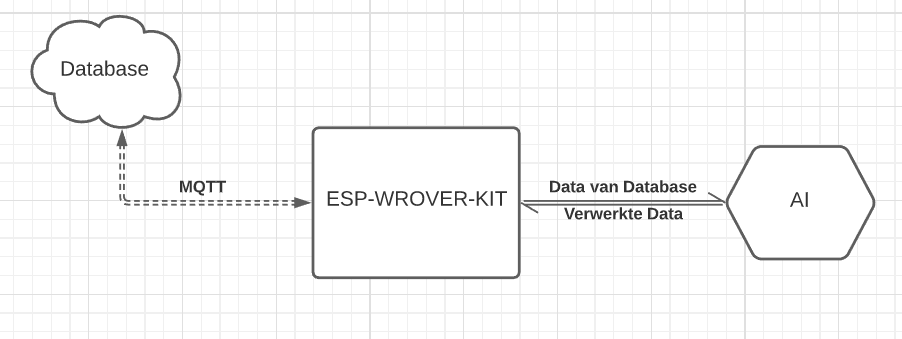


Fig. 2 Flowchart uiteindelijk project

De code geüpload op de esp-wrover-kit is geschreven in c. Deze code leest eerst de data uit van de DHT22. Vervolgens is deze data te zien in een seriële monitor.

# Resultaten

Met de code die we geschreven hebben kunnen we de data uitlezen van de DHT22 en deze ook zien in de seriële monitor berkijken, zoals te zien is in Fig. 3.

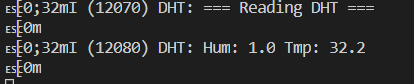


Fig. 3 Serial monitor esp data

# Discussie

Het is ons gelukt om de data van de dht22 uit te lezen. De MQTT communicatie daarentegen moet nog gemaakt worden. Dus is het onderzoek niet volledig gelukt maar er zijn wel enkele vaststellingen opgekomen.

Uit dit onderzoek zijn we ook te weten gekomen dat men niet alles moet installeren dat een ander programma aanwijst als nodig te zijn voor te werken. Tijdens de installatie van PlatformIO werd een basis hello\_world voorbeeld gebruikt. Hier werd er door Visual Studio Code (VSCode) aangewezen dat men de cmake-tools nodig had om de code te compileren en runnen. Maar door deze te installeren werden er bepaalde onderdelen die nodig waren om de espressif framework te gebruiken, veranderd. Enkele voorbeelden hiervan zijn dat in de PATH files dubbelpunten (:) en backslash (\) werden vervangen door puntkomma’s (;) en forwardslash (/). Dit zorgde ervoor dat Windows deze niet meer herkende als file locaties.

Zoals er in Resultaten te zien is, lukt het om de data uit te lezen en deze te tonen. De data zouden dan verwerkt kunnen worden voor verder gebruik en verzending.

De MQTT communicatie in het espressif framework bleek een stuk ingewikkelder dan origineel verwacht. Door de gelimiteerde tijd die ter beschikking was is de MQTT communicatie niet toegepast op dit onderzoek. Indien de tijd niet gelimiteerd was, had het mogelijk geweest om MQTT toe te passen en zo een communicatielijn te openen met de database. Eenmaal deze communicatielijn open zou zijn was het mogelijk geweest om de flowchart die te zien is in Fig. 2 werkelijkheid te maken en zo het grotere project te realiseren.

# Conclusie

Het onderzoek ging over communicatie te openen van een sensor naar een database via de MQTT methode. Deze communicatie ging via een esp-wrover-kit, voor deze te programmeren is er gebruik gemaakt van PlaformIO in de espressif framework. Hier kwamen echter enkele problemen naar boven en na deze problemen waren opgelost is het gelukt om de esp te programmeren en zo data uit te lezen maar het is nog niet gelukt om deze data door te sturen via de MQTT methode.

Uit dit onderzoek hebben we geleerd dat een framework een groot verschil kan maken op hoe bepaalde methodes worden uitgevoerd en geprogrammeerd.

Hieruit hebben we geleerd dat men niet alles moet installeren waarvan gezegd wordt dat die nodig zijn voor de code te doen werken, want deze kunnen er juist voor zorgen dat er bepaalde delen niet meer herkend worden door de OS.