Soft UART Challenge

Tiemon Steeghs

Inhoud

[1. Aanleiding 3](#_Toc126916067)

[2. Probleemstelling 3](#_Toc126916068)

[3. Onderzoek 4](#_Toc126916069)

[4. Proof of concept 5](#_Toc126916070)

[5. Testen 6](#_Toc126916071)

[6. Conclusie 7](#_Toc126916072)

[7. Bronvermelding 8](#_Toc126916073)

# Aanleiding

Om UART nog beter te begrijpen ga ik het in deze challenge in een moeilijkere context laten zien. Doormiddel van software matig UART toe te passen hoop ik UART beter te begrijpen en hoop ik dit ook te kunnen laten zien.

# Probleemstelling

UART communicaties maak ik al een lange tijd gebruik van. Maar ik heb eigenlijk al die tijd de standaard functies gebruikt die in de Arduino library worden meegegeven. Ik weet dus eigenlijk nog helemaal niet hoe je dit handmatig zou doen zonder gebruik te maken van libraries.

# Onderzoek

## Hoofdvraag

**Hoe zet je een softwarematige UART communicatie op tussen een Arduino en een Laptop zonder gebruik te maken van bestaande libraries?**

## Deelvraag 1

**Wat is softwarematige UART?**

Software matige UART is een implementatie van UART die embedded systemen zonder een UART module toch de mogelijkheid geeft om een UART communicatie op te zetten. Soft UART simuleert eigenlijk hard UART, doormiddel van de communicatie snelheid, de bit formatie van het bericht en de verwerking van fouten te regelen.

Software matige UART is een goede oplossing voor systemen die niet de hardware hebben die UART ondersteunt. Over het algemeen is software matige UART wel minder snel en ook minder betrouwbaar dan hardware versies, vooral bij de versturing van grote hoeveelheden data wordt dit duidelijk.

## Deelvraag 2

**Wat is het verschil tussen soft UART en hard UART?**

Software UART is een implementatie van UART die volledig software matig is opgesteld. Dit houd in dat er geen speciale hardware module wordt gebruikt, om de UART werkelijkheid te maken. Hardware UART daarentegen, maakt juist gebruik van een module die UART ondersteunt.

De grootste technische verschillen tussen de twee, zijn de betrouwbaarheid en de snelheid. Software UART is over het algemeen minder betrouwbaar en ook minder snel dan hardware UART.

## Deelvraag 3

**Wat is een statemachine?**

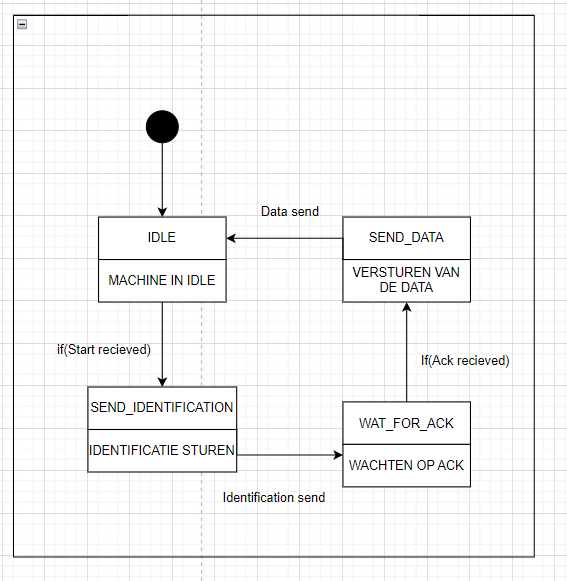
Een statemachine is een model waarin wordt beschreven hoe een machine werkt. Je kan in dit model zien wat voor processen er plaats vinden en hoe deze op elkaar in spelen. Een statemachine bestaat uit een selectie van toestanden waarin de machine zich kan bevinden. Aan de hand van “Events” of gebeurtenissen veranderd de machine van toestand.

## Deelvraag 4

**Waaruit bestaat de statemachine die voor deze challenge van toepassing zou zijn?**

Voor deze challenge maak ik een statemachine die zich in vier toestanden kan bevinden: IDLE, SEND\_IDENTIFICATION, WAIT\_FOR\_ACK en SEND\_DATA. De machine zal aan de hand van commando’s die hij van de laptop ontvangt veranderen van state.

Als eerste start de machine in de “IDLE” toestand. In deze toestand doet de machine nog niet veel, hij wacht namelijk op een start commando van de laptop. Op het moment dat hij het start commando heeft ontvangen stuurt de state machine een identificatie bericht in de SEND\_IDENTIFICATION state. Na het bericht gestuurd te hebben gaat de machine naar de WAIT\_FOR\_ACK toestand. In deze toestand wacht de machine weer op een bericht van de laptop. Op het moment dat hij dit bericht ontvangen heeft zal hij naar de SEND\_DATA toestand gaan en de data sturen. Na de data gestuurd te hebben gaat de machine weer terug naar de IDLE toestand.



## Deelvraag 5

**Hoe zet je software matige UART op?**

Software matige UART kan je opzetten doormiddel van alle functionaliteit van de hardware UART te simuleren. Hierbij beginnen we met het instellen van de baudrate. Nadat deze is ingesteld moeten we ook de bit-time bepalen. Dit is steeds de tijd die de arduino moet wachten om de volgende bit te gaan sturen.

Afbeelding met diagram

Automatisch gegenereerde beschrijving

# Bronvermelding

Software uart:

[**https://doc.riot-os.org/group\_\_drivers\_\_soft\_\_uart.html**](https://doc.riot-os.org/group__drivers__soft__uart.html)

Software matige serial communicatie op de arduino documentatie:

<https://docs.arduino.cc/learn/built-in-libraries/software-serial>

<https://deepbluembedded.com/stm32-external-interrupt-example-lab/>