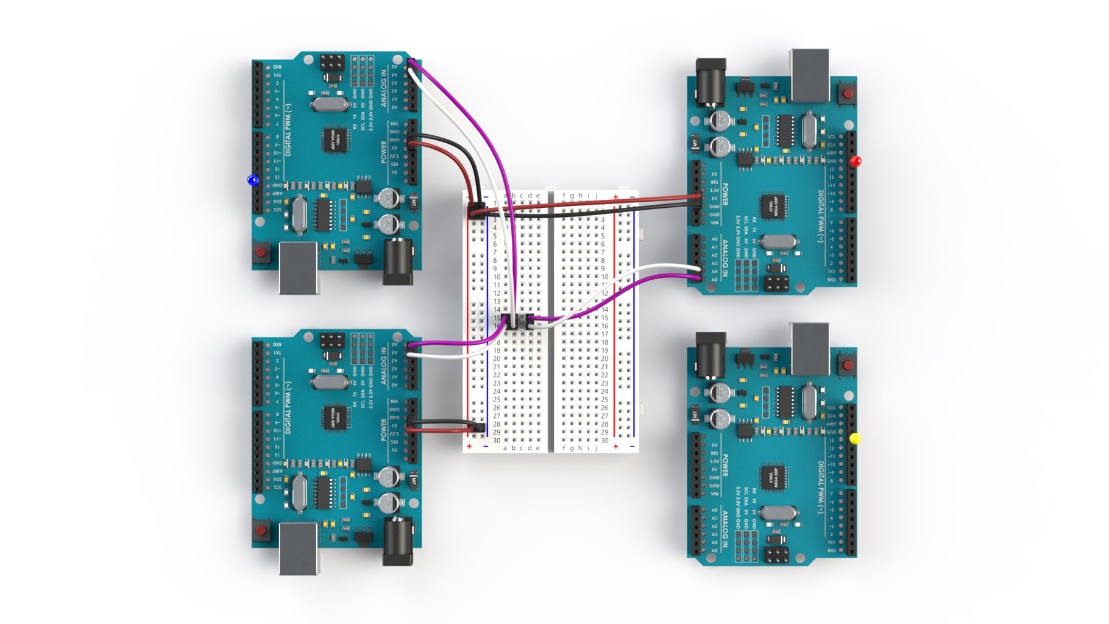
Sharing resources I2C

Tiemon Steeghs

Inhoud

[1. Probleemstelling 3](#_Toc137408685)

[2. Design 4](#_Toc137408686)

[3. Implementatie 5](#_Toc137408687)

[3.1. Versie 1: Status opvragen 5](#_Toc137408688)

[3.2. Versie 2: Lowcode data lezen 5](#_Toc137408689)

[3.3. Versie 3: SDA-lijn direct uitlezen 6](#_Toc137408690)

[4. Bronvermelding 9](#_Toc137408691)

# Probleemstelling

Binnen I2C systemen gaat er veel data heen en weer. Maar wat nou als twee masters tegelijkertijd data willen versturen naar een slave? In deze challenge ga ik kijken hoe je ervoor zorgt dat dit geen problemen oplevert, zodat de communicatie vlekkeloos blijft lopen. Een verdere beschrijving van mijn opstelling is in het volgende kopje beschreven.

# Design

Mijn systeem bestaat uit twee Arduino redboards (masters) en één Qwiic oled display (slave). Mijn systeem gaat als volgt te werk: De twee redboard masters en de Qwiic oled display zitten allemaal aan elkaar aan een I2C bus. Beiden masters kunnen data schrijven naar de display, om zo het gene wat afgebeeld wordt te veranderen. Voordat ze dit doen vragen ze eerst aan de andere master of deze niet al bezig is met het schrijven van data. Op het moment dat de master terugkrijgt van de andere master dat de bus beschikbaar is, zal hij data gaan schrijven. Het is belangrijk om op te merken dat de slave op zichzelf niet geprogrameerd wordt, dus alles wordt geregeld door de twee masters.

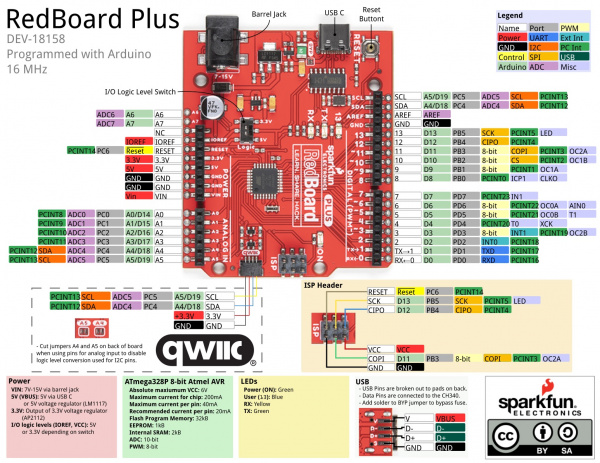
De beschreven opstelling is hieronder ook weergegeven. Bij de pijlen zie je dus ook staan wat de masters opvragen en verkrijgen. De data van de beschikbaarheid gaat overigens via de display omdat die tussen de masters in zit, echter doet de display hier niks mee.

Afbeelding met tekst, lijn, schermopname, diagram

Automatisch gegenereerde beschrijving

Figuur Diagram van het design

Om het systeem werkende te krijgen moeten de twee redboards steeds naar de SDA lijn kijken of deze hoog of laag is. Dit moet overheen komen met de bits die ze willen sturen. Als dit niet overeenkomt weten ze dus dan de ander bezig met schrijven. Om de SDA lijn te kunnen lezen moet ik eerst ontdekken of dit wel kan.



Figuur Pin layout van de Redboard

# Implementatie

Voor de daadwerkelijke implementatie ben ik door verschillende versies gegaan. Dit komt doordat ik erachter kwam dat mijn originele design enkele problemen met zich mee bracht dit wordt hieronder verder beschreven.

## Versie 1: Status opvragen

Mijn eerste oplossing was doormiddel van de masters eerst een bericht naar de ander te laten sturen voor het opvragen van of de bus beschikbaar is.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Echter kwam ik erachter dat als de twee masters precies tegelijk de data zouden opvragen er problemen zullen optreden. Deze implementatie is daarom nog niet voldoende dus ben ik naar andere oplossingen gaan kijken.

## Versie 2: Lowcode data lezen

Hierna ben ik gaan kijken naar een lowcode oplossing om de SDA lijn af te lezen. Echter had deze oplossing ook nog een probleem. Deze methode leest alleen data af die naar het adres van de desbetreffende master wordt gestuurd. Dit betekent dus dat als de master data zou gaan sturen naar de slave op exact hetzelfde moment er nog steeds een probleem optreed.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

## Versie 3: SDA-lijn direct uitlezen

Mijn derde versie van de implementatie lijkt wel een geldige oplossing te zijn. Door direct de SDA-lijn af te lezen kan ik aan de lijn zien of de andere master bezig. Op het moment dat er geen data wordt geschreven word de lijn als hoog afgelezen. Maar op het moment dat dus de andere master data gaat schrijven naar de slave, kan de master dit waarnemen.



Figuur Het lezen van de SDA-lijn

Afbeelding met tekst, schermopname

Automatisch gegenereerde beschrijving

Figuur Het resultaat van de lezing

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijvingVoor mijn kleine POC laat ik de masters een byte aan data sturen en niet een volledige afbeelding omdat dit erg veel bits zijn om te controleren. Voor deze POC gebruik ik daarom de data() methode voor het versturen van data naar de slave. Ik kan hiervoor niet de wire methodes gebruiken omdat deze niet samen gaat met de digitalRead() methode. “You can't use digitalRead() with I2C.” (*Arduino Forums*, 2018b)



Als je nu kijkt wat de andere master op de SDA leest kan hij vrij accuraat zien hoe de data wordt gestuurd. Echter is het nog niet volledig accuraat omdat de timing niet helemaal overheenkomt.

Om de timing te controlleren ben ik gaan kijken naar het I2C signaal met de logic analyzer. Ik stuur nu elke halve seconde data, wat je hieronder ook ziet.

Afbeelding met schermopname, Multimediasoftware, Grafische software

Automatisch gegenereerde beschrijving

Als we verder inzoomen zie je de negen bits die ik aan het sturen ben. Één start bit en acht data bits.

Afbeelding met schermopname, lijn, Rechthoek, Parallel

Automatisch gegenereerde beschrijving

Als we nu kijken naar de tijd tussen elke bit zien we dat dit 30 microseconden is.

Afbeelding met schermopname, Multimediasoftware, software

Automatisch gegenereerde beschrijving

Een volledige bit verzending duurt ongeveer 303 microseconden. Door nu steeds deze tijd te wachten nadat ik elke bit ontvang kan nog accurater de SDA-lijn uitlezen.

Ik kan nu dus redelijk accuraat uitlezen via de SDA-lijn wat de andere master aan het versturen is. Alleen is dit maar de helft van het probleem. De master moet ook instaat zijn om zijn eigen data te lezen op de SDA-lijn. Het is alleen niet zeker of dit wel mogelijk is. Met de digitalRead() methode van de Arduino library kan dit in ieder geval niet omdat hij wat te sloom is. Dit betekent dus dat het probleem maar voor de helft kan worden opgelost.

# Bronvermelding

*Qwiic Micro OLED Hookup Guide - SparkFun Learn*. (z.d.). <https://learn.sparkfun.com/tutorials/qwiic-micro-oled-hookup-guide/all>

*ATmega328P datasheet*

<https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>

*IS THAT ABLE TO DIGITAL READ A I2C BUS SENSOR*. (2018, 24 december). Arduino Forum. <https://forum.arduino.cc/t/is-that-able-to-digital-read-a-i2c-bus-sensor/563734/8>

*Gammon Forum : Electronics : Microprocessors : I2C - Two-Wire Peripheral Interface - for Arduino*. (z.d.). <http://www.gammon.com.au/i2c>