**Startsemester  
Oriëntatieverslag  
Technology**

**Studentnaam: Marijn Verschuren**

**Studentnummer: 510936**

**Klas: PDB-11  
Vakdocent: Rop Pulles  
  
Versie: 1.1  
Datum: 31-8-2022**

**Versiebeheer**

*/\*Haal deze uitleg weg als je snapt wat de bedoeling is.  
Als je een nieuw document in Canvas inlevert, dan hoog je het versienummer op. Je vult de nieuwe datum en je naam in. Bij veranderingen zet je kort neer wat je gewijzigd hebt. Als voorbeeld: “Challenge 8 toegevoegd”.  
Vervolgens wijzig je de datum en het versienummer onderaan de vorige pagina (titelblad),\*/*

|  |  |  |  |
| --- | --- | --- | --- |
| **Versienummer** | **Datum** | **Auteur** | **Veranderingen** |
| *1.0* | *14-09-2022* | *Marijn Verschuren* | *Initiële document gemaakt. + TMC2209* |
| *1.1* | *12-10-2022* | *Marijn Verschuren* | *AS5600 + Reflecie* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Inhoudsopgave

Inhoud

[Inhoudsopgave 3](#_Toc117086126)

[1 Tabel Leeruitkomsten 4](#_Toc117086127)

[1.1 **Oriënterend niveau** 4](#_Toc117086128)

[2 TMC2208 motor driver 5](#_Toc117086129)

[2.1 Challenge Beschrijving 5](#_Toc117086130)

[2.2 Flowchart 5](#_Toc117086131)

[2.3 Aanpak 5](#_Toc117086132)

[2.4 Test resultaat 5](#_Toc117086133)

[2.5 Wat heb ik geleerd? 5](#_Toc117086134)

[3 AS5600 magnetic angle encoder 9](#_Toc117086135)

[3.1 Challenge Beschrijving 9](#_Toc117086136)

[3.2 Flowchart 9](#_Toc117086137)

[3.3 Aanpak 9](#_Toc117086138)

[3.4 Test resultaat 9](#_Toc117086139)

[3.5 Wat heb ik geleerd? 9](#_Toc117086140)

[3.6 Bronnen 10](#_Toc117086141)

[4 Reflectie / evaluatie 11](#_Toc117086142)

[4.1 Waar ben ik trots op? 11](#_Toc117086143)

[4.2 Wat doe ik een volgende keer anders? 11](#_Toc117086144)

[4.3 Welke formatieve indicatie zou ik mezelf geven voor de oriëntatie Technology? 11](#_Toc117086145)

[4.4 Welk verdiepend(e) profiel(en) kies ik en waarom? 11](#_Toc117086146)

# Tabel Leeruitkomsten

Je ontwikkelt en programmeert interactieve embedded systemen, waarbij sensoren en actuatoren toegepast worden, die verschillende I/O technieken gebruiken.

## **Oriënterend niveau**

* **Interactieve embedded systemen**  
  Je stelt een systeem samen welke bestaat uit verschillende componenten. Hiervoor maak je gebruik van een microcontroller board, sensoren en actuatoren.
* Je systeem kan communiceren met een ander systeem volgens een eigen gedefinieerd protocol.
* **Programmeren**  
  Je begrijpt en past de volgende programmeerconcepten toe: variables, conditional statements, loops, functions, arrays.
* De focus hierbij is op het werkend krijgen van het product.
* **Sensoren**  
  Je kunt sensoren toepassen, zoals een button, potentiometer, afstandssensor.
* **Actuatoren**  
  Je kunt actuatoren toepassen, zoals een led, motor, buzzer.
* **Verschillende I/O technieken**  
  Je kunt verschillende I/O technieken toepassen, zoals digitale input en output en analoge input.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Actuatoren** | **sensoren** | **Programmeren** | Variabelen | Condional staements | loops | functions | Arrays | **Digitale Input en Output** | **Analoge Input** | **Comment** |
| **Challenge** | **x** |  | **x** | **x** | **x** | **x** | **x** |  | **x** |  | TMC2008 aangestuurd |
| **Challenge** |  | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | x | AS5600 aangestuurd |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

# TMC2208 motor driver

## Challenge Beschrijving

Een stepper motor aansturen met de TMC2208 motor driver

## Flowchart

Zie: Tekening.vsdx

## Aanpak

Ik heb door middel van 6 pins de TMC2208 aangestuurd

PA9 -> MS1

PA8 -> MS2

(

MS1 0, MS2, 0 -> 8 microsteps

MS1 1, MS2 0 -> 2 microsteps

MS1 0, MS2 1 -> 4 microsteps

MS1 1, MS2 1 -> 16 microsteps

)

PB15 -> SRD // spread mode

PB-14 -> DIR // direction (0: CW, 1: CCW)

PB-13 -> STP // step pulse

PB-12 -> NEN // not enable

## Test resultaat

<https://youtube.com/shorts/iROxng_Yvek>

## Wat heb ik geleerd?

Text

Description automatically generated(main.h)

Enum met alle microstep settings

Text

Description automatically generated(main.h)

Structure die de motor instructie opslaat (word in een volgende versie via SPI ontvangen van daar de CRC16)

Text

Description automatically generated(main.h)

Structure die de positie (in stappen) en de stappen die nog genomen moeten worden bij houd

Deze structure moet nog veranderd worden zodat het de actuele posite in graden of radialen opslaat samen met het aantal graden of radialen dat nog gedraaid gaat worden omdat het aantal stappen meer of minder meetelt als je een andere microstep setting gebruikt.

Graphical user interface, text

Description automatically generated (main.h)

(main.c)

Deze variablen zijn als extern in main.h gedefineerd zodat andere .h en .c files deze kunnen gebruiken

In main.c zijn deze variables daadwerkelijk aangemaakt

(main.c)

Graphical user interface, text, application

Description automatically generated (config)

Diagram

Description automatically generated(config)

Functie die n microsecondes wacht doormiddel van een timer deze timer is aangesloten aan de internal clock die op 96Mhz draait via APB1 (gemarkeerd) de timer heeft een prescaler van 95 dat betekend dat de counter word opgehoogd elke 95 + 1 clockcycles dus deze counter word opgehoogd met een frequentie van 1Mhz of wel 1\*10^6 keer per seconden of wel een keer per microseconde dus als je wacht totdat de counter (start op 0) gelijk is aan n heb je n microseconden gewacht.

(main.c)

Zet alle belangrijke variabelen op 0

Text

Description automatically generated (main.c)

Creeer de variabelen die in de loop gebruikt gaan worden en zet ze op 0

Text

Description automatically generated(main.c)

Maak een instructie om de motor te testen in dit geval:

Maak 1\*10^8 stappen CCW, delay van 75 us, 16 microsteps, spread\_mode = false

Text

Description automatically generated

In de main loop word gekeken of er stappen gezet moeten worden volgens de instructie

dan worden de DIR pin op de juiste waarde gezet en de NEN pin laag gezet

dan word eer een iteratie count uitgerekend (max 4096, deze maximale waarde bestaat omdat als de instructie aangepast word via de DMA moet dit invloed hebben op de motor (dit gaat waarschijnlijk veranderen of weg))

dan worden er stappen gezet

en als het stappen ophoud word de NEN pin weer hoog gezet

# AS5600 magnetic angle encoder

## Challenge Beschrijving

Het initialiseren en aflezen van de AS5600

## Flowchart

Diagram

Description automatically generated

## Aanpak

De initialisatie functie schrijft eerst een configuratie naar het register 0x07 en leest dan het register 0x11 af waardoor je een code krijgt deze code moeten de bitjes voor 0x20 aan staan en 0x10, 0x08 uit staan

## Test resultaat

<https://youtube.com/shorts/G3Ni2nRQHQA>

## Wat heb ik geleerd?

* Dat de I2C bus hoog staat als deze idle is
* Dat het hooghouden van de I2C bus meestal gedaan word door de “slave”
* Dat I2C fast mode niet goed werkt op de STM32F4 nucleo
* Dat de maker van de AS5600 library I2C interupt calls heeft gebruikt in plaats van normale I2C calls waardoor de value nog niet binnen is na de call wat samen met mijn clock speed instelling ervoor zorgt dat de value al gecheckt word voordat de value binnen is.

Text

Description automatically generated

## Bronnen

<https://github.com/MarijnVerschuren/STM32F4_AS5600_library>

# Reflectie / evaluatie

## Waar ben ik trots op?

Dat ik de I2C bus tussen de STM32F411 en de AS5600 werkend heb kunnen maken

## Wat doe ik een volgende keer anders?

Eerst de datasheet lezen en wat meer onderzoek doen naar concepten zoals I2C

## Welke formatieve indicatie zou ik mezelf geven voor de oriëntatie Technology?

|  |  |  |
| --- | --- | --- |
| Leeruitkomst oriëntatie Technology | | |
| Onderdeel | Criterium | Rating |
| Interactieve embedded systemen | Je stelt een systeem samen welke bestaat uit verschillende componenten. Hiervoor maak je gebruik van een microcontroller board, sensoren en actuatoren. Je product kan communiceren met een ander systeem volgens een eigen gedefinieerd protocol. | *O, Mijn systeem heeft de volgende delen:*  *Componenten:*   * *Arduino Mega2560 (MC)* * *STM32F411 (MC)* * *AS5600 (sensor)* * *TMC2209 (actuator)*   *Protocollen:*   * *Uart (USB met eigen protocol)* * *SPI (Tussen Arduino en STM)* * *I2C (Tussen STM en AS5600)* * *PWM (Tussen STM en AS5600)* |
| Programmeren | Je begrijpt en past de volgende programmeerconcepten toe: variables, conditional statements, loops, functions, arrays. De focus hierbij is op het werkend krijgen van het product. | *O, in de code zitten: variabelen, arrays, structures, functions, loops. Conditionals, interupts (functions)* |
| Sensoren | Je kunt sensoren toepassen, zoals een button, potentiometer, afstandssensor. | G, de AS5600 sensor kan nu ingesteld worden maar nog niet afgelezen worden |
| Actuatoren | Je kunt actuatoren toepassen, zoals een led, motor, buzzer. | *O, ik heb de stepper motor laten draaien met de TMC2209* |
| Verschillende I/O technieken | Je kunt verschillende I/O technieken toepassen, zoals digitale input en output en analoge input. | *G, ik heb bijna alle soorten IO technieken gebruikt:*   * *Digitial IO (met en zonder external PU/PD)* * *PWM read (aflezen van AS5600, nog niet gemaakt)* |

## Welk verdiepend(e) profiel(en) kies ik en waarom?

*Tech omdat ik het leuk vind om echt iets fysieks te maken.*