

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
1 Tabel Leерuitkomsten.....	4
1.1 Oriënterend niveau	4
2 TMC2208 motor driver.....	5
2.1 Challenge Beschrijving.....	5
2.2 Flowchart.....	5
2.3 Aanpak.....	5
2.4 Test resultaat.....	5
2.5 Wat heb ik geleerd?.....	5
3 AS5600 magnetic angle encoder.....	9
3.1 Challenge Beschrijving.....	9
3.2 Flowchart.....	9
3.3 Aanpak.....	9
3.4 Test resultaat.....	9
3.5 Wat heb ik geleerd?.....	9
3.6 Bronnen.....	10
4 Reflectie / evaluatie.....	11
4.1 Waar ben ik trots op?.....	11
4.2 Wat doe ik een volgende keer anders?.....	11
4.3 Welke formatieve indicatie zou ik mezelf geven voor de oriëntatie Technology?.....	11
4.4 Welk verdiepend(e) profiel(en) kies ik en waarom?.....	11

1 Tabel Leeruitkomsten

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

1.1 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	Conditional statements	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

2 TMC2208 motor driver

2.1 Challenge Beschrijving

Een stepper motor aansturen met de TMC2208 motor driver

2.2 Flowchart

Zie: Tekening.vsdx

2.3 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

2.4 Test resultaat

https://youtube.com/shorts/iROxng_Yvek

2.5 Wat heb ik geleerd?

```
#typedef enum { // MS2, MS1
    RST = 0xfe6f, // 1111 1111 00 11 1111    bit mask to reset M1 and M2
    M2 = 0x0040, // 0000 0000 01 00 0000    pin M1 high
    M4 = 0x0080, // 0000 0000 10 00 0000    pin M2 high
    M8 = 0x0000, // 0000 0000 00 00 0000    both pins low
    M16 = 0x00b0 // 0000 0000 11 00 0000    both pins high
} MICRO_STEP;
```

(main.h)

Enum met alle microstep settings

```
typedef struct {
    int64_t steps;
    uint32_t pulse_delay; // delay from 1us -> 4295s (1 is added to delay because 0 us is not valid)
    struct {
        uint8_t micro_step : 2; // 0 - 8 (0: MS2, 1: MS4, 2: MS8, 3: MS16)
        uint8_t spread_mode : 1; // spread mode flag
        uint16_t _ : 13; // reserved
    } settings;
    uint16_t crc; // add crc16?
} MCU_Instruction;
```

(main.h)

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

```
typedef struct {
    int64_t pos;
    int64_t job;
} MCU_State;
```

(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.

```
extern MCU_State state;
extern MCU_Instruction instruction;
extern AS5600_TypeDef* sensor;
```

(main.h)

```
MCU_State state; // structure
MCU_Instruction instruction;
AS5600_TypeDef* sensor;
```

(main.c)

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

In main.c zijn deze variables daadwerkelijk aangemaakt

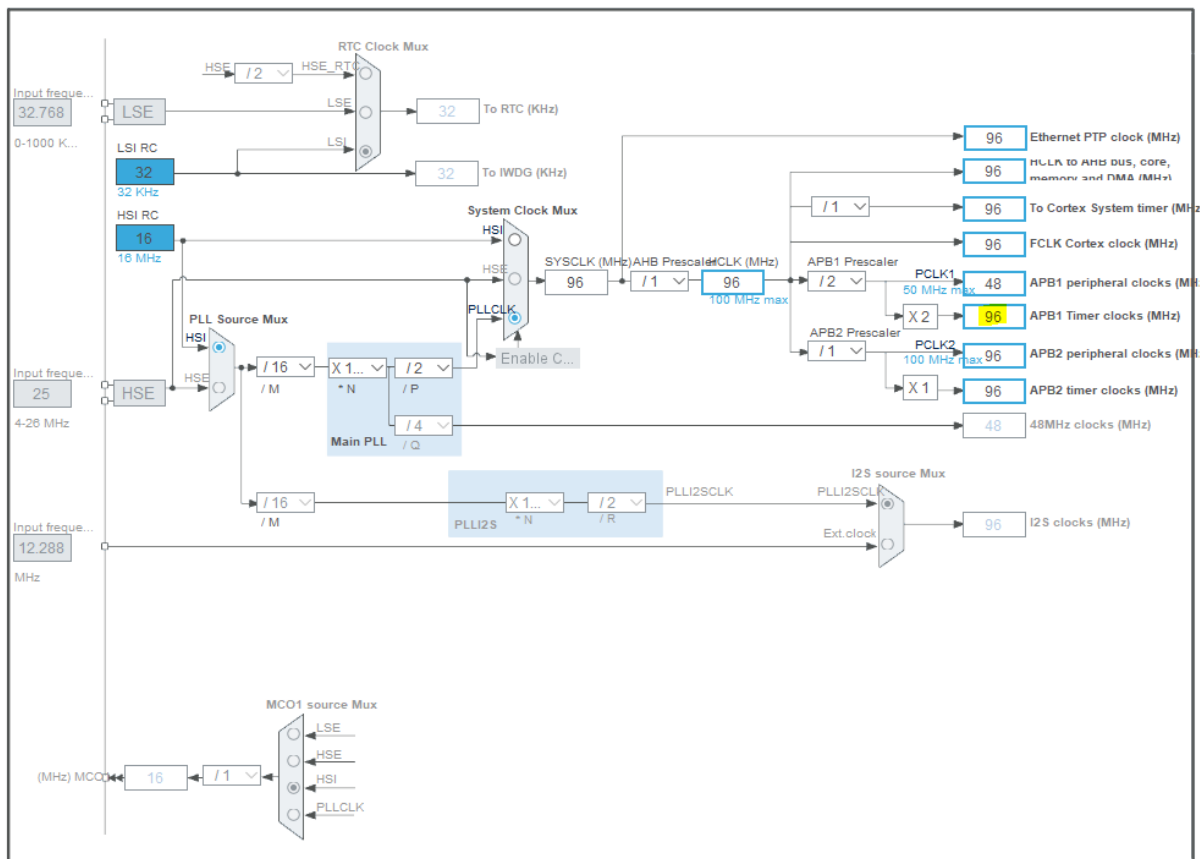
```
void delay_us(uint16_t n) { TIM1->CNT = 0; while(TIM1->CNT < n); }
void init_timer(TIM_TypeDef* instance) {
```

(main.c)

Counter Settings

Prescaler (PSC - 16 bits value)	95
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	0xffff
Internal Clock Division (CKD)	No Division

(config)



(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 $1 \cdot 10^6$ 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.

```
state.pos = 0;
state.job = 0;
instruction.steps = 0; (main.c)
```

Zet alle belangrijke variabelen op 0

```
uint64_t iter = 0;
int8_t mult = 0;
uint64_t pulse_delay_us = 0; (main.c)
```

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

```
instruction.steps = -100000000;
instruction.pulse_delay = 74; // 74; // safe operating range is from 75us and up
instruction.settings.micro_step = 3;
instruction.settings.spread_mode = 0;
```

(main.c)

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

Maak $1 \cdot 10^8$ stappen CCW, delay van 75 us, 16 microsteps, spread_mode = false

```

while (1) {
    if (instruction.steps == 0) { continue; }
    set_motor_setting(&instruction);

    state.job = instruction.steps;
    pulse_delay_us = instruction.pulse_delay + 1;
    instruction.steps = 0;

    HAL_GPIO_WritePin(STEPPER_NEN_GPIO_Port, STEPPER_NEN_Pin, 0);
    HAL_GPIO_WritePin(STEPPER_DIR_GPIO_Port, STEPPER_DIR_Pin, state.job > 0);
    // capture the current job (this can change via interrupt)
    mult = state.job > 0 ? 1 : -1;
    iter = abs_64(state.job);
    for (uint64_t i = 0; i < iter; i++) { // iterate this job for max 4096 iterations at the time before SPI receive
        // update variables first because the SPI DMA transmit has a almost 100% chance to happen during pulse delay
        state.pos += mult;
        state.job -= mult;
        HAL_GPIO_WritePin(STEPPER_STP_GPIO_Port, STEPPER_STP_Pin, 1);
        delay_us(pulse_delay_us);
        HAL_GPIO_WritePin(STEPPER_STP_GPIO_Port, STEPPER_STP_Pin, 0);
        delay_us(pulse_delay_us);
    }
    HAL_GPIO_WritePin(STEPPER_NEN_GPIO_Port, STEPPER_NEN_Pin, 1);
}
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */

```

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

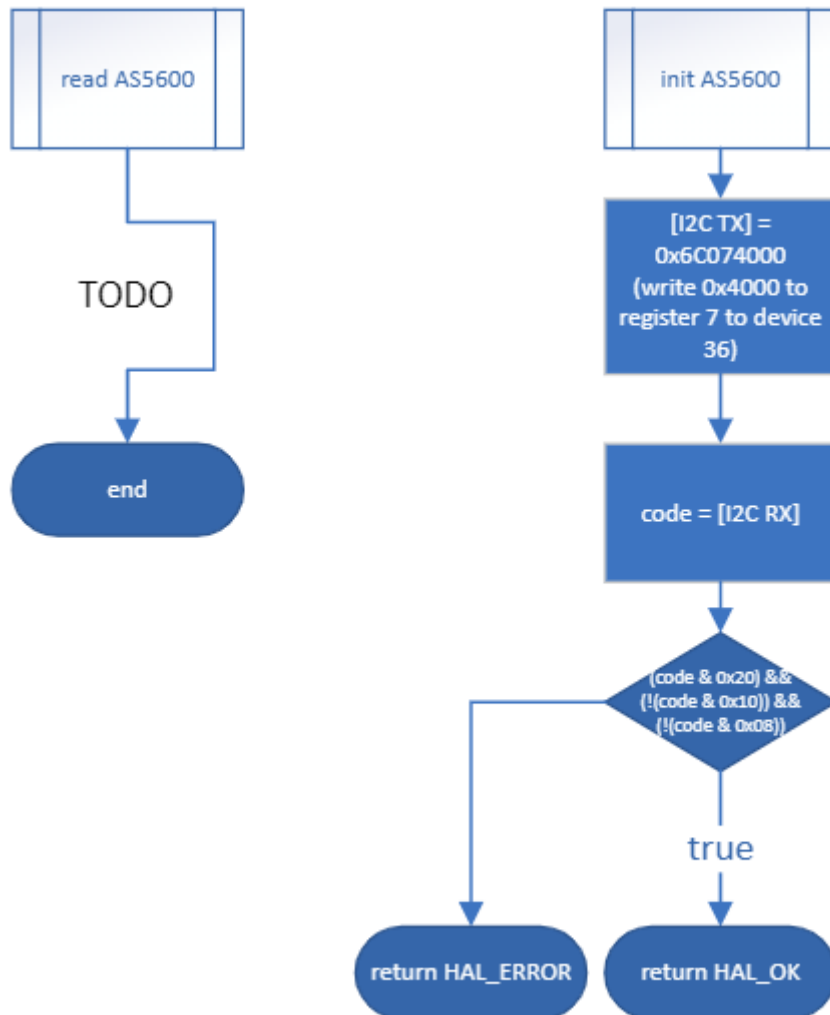
en als het stappen ophoud word de NEN pin weer hoog gezet

3 AS5600 magnetic angle encoder

3.1 Challenge Beschrijving

Het initialiseren en aflezen van de AS5600

3.2 Flowchart



3.3 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

3.4 Test resultaat

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

3.5 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 interrupt 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.

```

if (HAL_I2C_Mem_Write_IT(a->i2cHandle, a->i2cAddr,
                        AS5600_REGISTER_INF_HIGH, I2C_MEMADD_SIZE_8BIT,
                        a->confRegister, 1) != HAL_OK) {
    status = HAL_ERROR;
    return status;
}

HAL_Delay(5);

/* Check magnet status */
if (AS5600_GetMagnetStatus(a, &mag_status) != HAL_OK) {
    status = HAL_ERROR;
    return status;
}

HAL_Delay(5);

if (!(mag_status & AS5600_MAGNET_DETECTED)) {
    /* Magnet not detected */
    status = HAL_ERROR;
    return status;
}

if ((mag_status & AS5600_AGC_MIN_GAIN_OVERFLOW)) {
    /* B-field is too strong */
    status = HAL_ERROR;
    return status;
}

if ((mag_status & AS5600_AGC_MAX_GAIN_OVERFLOW)) {
    /* B-field is too weak */
    status = HAL_ERROR;
    return status;
}

/* Write */
return status;

```

Config 0x07
 0x36 << 1 = 0x6c

delay to stop cpu from checking before I2C completion

reading register 0x11

3.6 Bronnen

https://github.com/MarijnVerschuren/STM32F4_AS5600_library

4 Reflectie / evaluatie

4.1 Waar ben ik trots op?

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

4.2 Wat doe ik een volgende keer anders?

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

4.3 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.	<i>O, Mijn systeem heeft de volgende delen:</i> <i>Componenten:</i> <ul style="list-style-type: none">• <i>Arduino Mega2560 (MC)</i>• <i>STM32F411 (MC)</i>• <i>AS5600 (sensor)</i>• <i>TMC2209 (actuator)</i> <i>Protocollen:</i> <ul style="list-style-type: none">• <i>Uart (USB met eigen protocol)</i>• <i>SPI (Tussen Arduino en STM)</i>• <i>I2C (Tussen STM en AS5600)</i>• <i>PWM (Tussen STM en AS5600)</i>
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.	<i>O, in de code zitten: variabelen, arrays, structures, functions, loops. Conditionals, interrupts (functions)</i>
Sensoren	Je kunt sensoren toepassen, zoals een button, potentiometer, afstandssensor.	<i>G, de AS5600 sensor kan nu ingesteld worden maar nog niet afgelezen worden</i>
Actuatoren	Je kunt actuatoren toepassen, zoals een led, motor, buzzer.	<i>O, ik heb de stepper motor laten draaien met de TMC2209</i>
Verschillende I/O technieken	Je kunt verschillende I/O technieken toepassen, zoals digitale input en output en analoge input.	<i>G, ik heb bijna alle soorten IO technieken gebruikt:</i> <ul style="list-style-type: none">• <i>Digital IO (met en zonder external PU/PD)</i>• <i>PWM read (aflezen van AS5600, nog niet gemaakt)</i>

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

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