VAK: EPD

Daniël ROTH (496885)

Eindopdracht Spoorwegovergang

EPD

Klas: ITA-1DB

Leraar: Sander Leer

Inleverdatum: 07-04-2022

2022

Inhoudsopgave

[Inleiding 2](#_Toc100218277)

[1. Functioneel Ontwerp 3](#_Toc100218278)

[1.1 Requirements met MoSCoW prioritering 3](#_Toc100218279)

[2. Technisch Ontwerp 8](#_Toc100218280)

[2.1 Toestands diagram 8](#_Toc100218281)

[2.2 Softwareontwerp (tabbladendiagram) 10](#_Toc100218282)

[2.3 Toelichting bij de tabbladen en functies hierbinnen 11](#_Toc100218283)

[2.3.1 Highlevel-modules 11](#_Toc100218284)

[Spoorwegovergang 11](#_Toc100218285)

[Statemachine 11](#_Toc100218286)

[Treinen 12](#_Toc100218287)

[VerkeersdeelnemerDrukknoppen 12](#_Toc100218288)

[Slagboom 12](#_Toc100218289)

[Stoplicht 13](#_Toc100218290)

[2.3.2 Low-level Modules 13](#_Toc100218291)

[libSerial 14](#_Toc100218292)

[libLed 14](#_Toc100218293)

[libServo 14](#_Toc100218294)

[libPatronen 15](#_Toc100218295)

[libShiftRegister 15](#_Toc100218296)

[libDisplay 15](#_Toc100218297)

[libBuzzer 16](#_Toc100218298)

[libTimer 17](#_Toc100218299)

[libEEPROM 17](#_Toc100218300)

[Button 17](#_Toc100218301)

[ButtonHardware 18](#_Toc100218302)

[2.4 Aansluitschema (Tinkercad) 19](#_Toc100218303)

[Conclusie 20](#_Toc100218304)

[Bijlagen 21](#_Toc100218305)

[A. Zelfbeoordelingsformulier 21](#_Toc100218306)

# Inleiding

Dit is een document geschreven voor de eindopdracht ‘Spoorwegovergang’ voor het vak EPD, het document bevat: een functioneel ontwerp, een technisch ontwerp en een zelfbeoordelingsformulier. Ik heb gedurende de laatste weken behoorlijk wat geoefend met het programmeren met de Arduino dus ik heb een best goed gevoel over het uiteindelijke resultaat.

1. Functioneel Ontwerp

## Requirements met MoSCoW prioritering

**Functionele/non functionele eisen**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Omschrijving | Prioriteit | Opgeleverd | Testresultaat |
| **1.** | Er zijn 2 stoplichten, 1 wijst naar de noord kant, de ander naar de zuid kant. | M | ✅ | Geslaagd |
| **2.** | Een stoplicht bestaat uit 3 LED’s, de kleuren: rood, geel & blauw. | M | ✅ | Geslaagd |
| **3.** | Er kan maar 1 stoplicht tegelijk op groen staan. (groen betekent: u mag passeren) | M | ✅ | Geslaagd |
| **4.** | **4 sec** nadat een stoplicht groen is geworden gaat het stoplicht naar geel, het stoplicht blijft **2 sec** geel, hierna wordt het stoplicht rood. | M | ✅ | Geslaagd |
| **5.** | Nadat een stoplicht op rood is gegaan moet er **2 sec** gewacht worden en blijven beide stoplichten rood voordat het andere stoplicht naar groen mag. (ontruimingstijd) | M | ✅ | Geslaagd |
| **6.** | Een druk van een knop moet altijd onthouden worden totdat de bijbehorende actie verwerkt is. | M | ✅ | Geslaagd |
| **7.** | Wanneer beide stoplichten op rood staan, krijgt de eerste persoon die gedrukt heeft het groen. | S | ✅ | Geslaagd |
| **8.** | Als er een trein aankomt gaat een eventueel groen stoplicht onmiddellijk op geel, het stoplicht blijft **2 sec** geel en vervolgens wordt het rood. | M | ✅ | Geslaagd |
| Nr. | **Omschrijving** | **Prioriteit** | **Opgeleverd** | **Testresultaat** |
| 9. | In het spoor zitten aan 2 kanten van de overgang (oost en west) sensoren, gesimuleerd door 2 drukknoppen, 1 voor elke richting. | M | ✅ | Geslaagd |
| 10. | Als een sensor (oost of west) een trein signaleert gaat het verkeerslicht dat groen is naar geel, na **2 sec** gaat ook het rode licht branden en begint het gele licht te knipperen. | M | ✅ | Geslaagd, lampjes knipperen soms zwak wanneer de servo draait |
| 11. | Nadat het rode stoplicht brand en het gele licht knippert begint de slagboom langzaam te sluiten. | S | ✅ | Geslaagd |
| 12. | De slagboom wordt gesimuleerd door een servomotor. | M | ✅ | Geslaagd |
| 13. | Terwijl de slagboom sluit begint een buzzer een snel tikkend geluid te luiden met een frequentie van **100ms**. | S | ✅ | Geslaagd |
| 14. | Zodra de slagboom gesloten is blijven de stoplichten op rood staan en de gele lichten blijven knipperen. | S | ✅ | Geslaagd, lampjes knipperen soms zwak wanneer de servo draait |
| 15. | Zodra de slagboom gesloten is stopt de buzzer met tikken. | S | ✅ | Geslaagd |
| 16. | Zodra de sensor aan de andere richting dan waar de trein vandaan kwam een trein signaleert opent de slagboom weer. | M | ✅ | Geslaagd |
| 17. | Wanneer de slagboom weer opent begint de buzzer met tikken. | S | ✅ | Geslaagd |
| 18. | Wanneer de slagboom weer opent blijven de stoplichten op rood staan en de gele lichten blijven knipperen. | S | ✅ | Geslaagd, lampjes knipperen soms zwak wanneer de servo draait |

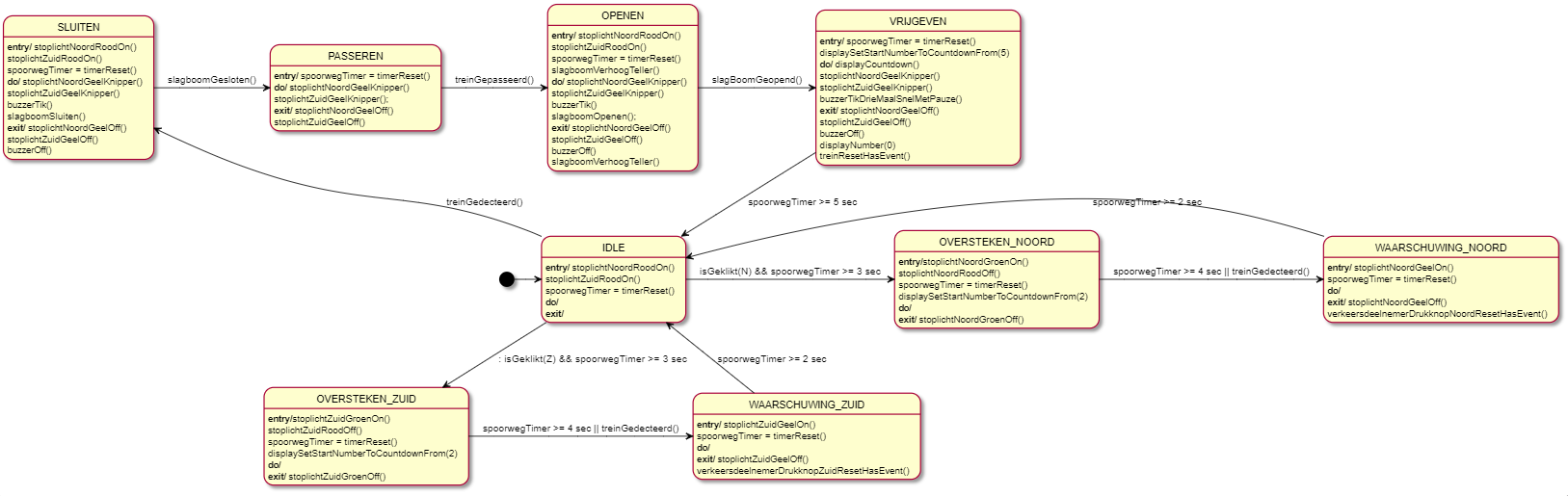
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Omschrijving | Prioriteit | Opgeleverd | Testresultaat |
| 19. | Zodra de slagboom volledig open is, telt het display van 5 naar 0. | S | ✅ | Geslaagd |
| 20. | Terwijl de display van 5 naar 0 telt klinkt uit de buzzer een signaal bestaande uit 3 tikken en dan een korte pauze. | S | ✅ | Geslaagd |
| 21. | Zodra het display op 0 staat stoppen de gele lichten van beide stoplichten met knipperen, de rode lichten blijven branden. | S | ✅ | Geslaagd |
| 22. | Zodra het display op 0 staat moet de buzzer stoppen met tikken. | S | ✅ | Geslaagd |
| 23. | Wanneer het display op 0 staat blijven de stoplichten op rood totdat er weer een verkeersdeelnemer wilt oversteken. | S | ✅ | Geslaagd |
| 24. | Wanneer het systeem in ontruimingstijd is, krijgt de trein voorrang op een verkeersdeelnemer. | M | ✅ | Geslaagd |
| 25. | Als de afhandeling van een knop bezig is en gedurende deze tijd de knop nogmaals ingedrukt wordt dan wordt deze indruk **niet** onthouden. | S | ✅ | Geslaagd |
| 26. | Als tweemaal of vaker achter elkaar een treinsensor geactiveerd wordt zonder dat de andere treinsensor wordt geactiveerd dan hoeft alleen de eerste activatie verwerkt te worden. | S | ✅ | Geslaagd |
| 27. | Bij het opstarten van het systeem zijn beide verkeerslichten op rood en is de slagboom geopend. | S | ✅ | Geslaagd |
| 28. | Er wordt een teller bij gehouden voor het aantal keren dat de slagboom geopend is geweest. | S | ✅ | Geslaagd |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Omschrijving | Prioriteit | Opgeleverd | Testresultaat |
| 29. | Wanneer de teller die het aantal keren dat de slagboom geopend is verhoogd wordt dan wordt de waarde van de teller via de seriële poort geprint. | S | ✅ | Geslaagd |
| 30. | Wanneer de letter ‘r’ via de seriële poort binnenkomt wordt de waarde van de teller die het aantal keren dat de slagboom geopend is bijhoudt weer op 0 gezet. | S | ✅ | Geslaagd |
| 31. | Wanneer tweemaal op een knop gedrukt wordt bij de noord of zuid richting dan blijft het stoplicht aan die kant langer op groen dan normaal. | W | ❌ | Niet geïmplementeerd |
| 32. | Een potmeter bepaalt de aankomende snelheid van een trein, hoe lager de waarde hoe meer tijd tussen het signaleren van een trein en het eventueel naar geel gaan van een groen stoplicht. | W | ❌ | Niet geïmplementeerd |
| 33. | Een thermometer wordt gebruikt om de kwaliteit van het weer te bepalen. | C | ❌ | Niet geïmplementeerd |
| 34. | Wanneer de thermometer een lage waarde aangeeft (slecht weer) dan moet de ‘geel’ stand van een stoplicht niet **2 sec** maar **4 sec** duren. | C | ❌ | Niet geïmplementeerd |
| 35. | Er is een testmode aanwezig, zodra via de seriële poort een ‘T’ gestuurd wordt gaan alle lichten van beide stoplichten een voor een aan en vervolgens weer uit, het slagboom opent en sluit langzaam en de buzzer geeft een pieptoon. | C | ❌ | Niet geïmplementeerd |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Omschrijving | Prioriteit | Opgeleverd | Testresultaat |
| 36. | Het aantal keer dat een slagboom geopend is wordt in een EEPROM opgeslagen, er hoeft niet verder dan 255 geteld te worden, door een ‘r’ via de seriële poort te versturen wordt deze waarde op 0 gezet. | C | ✅ | Geslaagd |
| 37. | De in de EEPROM opgeslagen slagboomteller kan grotere getallen dan 255 opslaan. | C | ❌ | Niet geïmplementeerd |
| 38. | Zodra de waarde van de slagboomteller groter is dan een bepaalde instelbare waarde gaat er een ledje branden zodat een monteur weet dat deze onderhouden moet worden. | C | ❌ | Niet geïmplementeerd |
| 39. | In oost en west richting is er een led die knippert op het moment dat de slagboom weer geopend is. | C | ❌ | Niet geïmplementeerd |
| 40. | Wanneer een stoplicht op geel gaat telt de display af van 2 naar 0. (de tijd dat het stoplicht op geel staat) | C | ✅ | Geslaagd |
| 41. | Wanneer tijdens het passeren van een trein aan beide kanten op een knop gedrukt wordt dan wordt de volgorde waarin deze knoppen gedrukt zijn onthouden en hierna worden de knoppen ook in deze volgorde afgehandeld. | C | ❌ | Niet geïmplementeerd |

1. Technisch Ontwerp

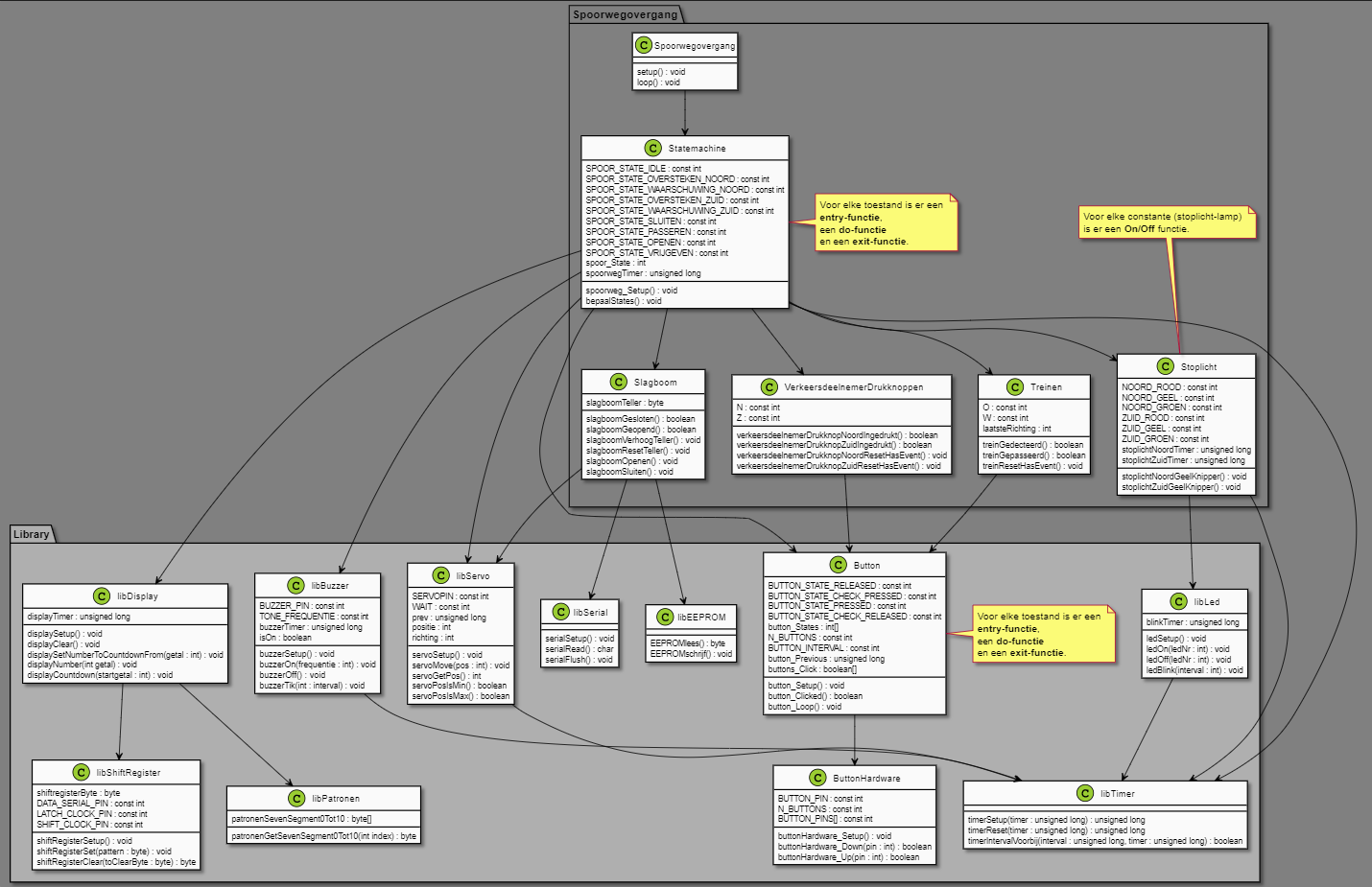
## Toestands diagram



Zoals te zien is heeft het systeem 9 verschillende states:

1. **Idle**: binnen deze state bevind het systeem bij het opstarten, ook wordt deze state elke keer weer bezocht wanneer 1 van de drie mogelijke paden doorgelopen is.
2. **Oversteken\_Noord**: naar deze state wordt gewisseld wanneer een verkeersdeelnemer op een drukknop drukt aan de noordelijke kant.
3. **Waarschuwing\_Noord**: deze state wordt bezocht vanuit de Oversteken\_Noord state wanneer de spoorwegTimer voorbij is of wanneer een trein gedecteerd wordt, na deze state wordt weer de Idle state bezocht.
4. **Oversteken\_Zuid**: naar deze state wordt gewisseld wanneer een verkeersdeelnemer op een drukknop drukt aan de zuidelijke kant.
5. **Waarschuwing\_Zuid**: deze state wordt bezocht vanuit de Oversteken\_Zuid state wanneer de spoorwegTimer voorbij is of wanneer een trein gedecteerd wordt, na deze state wordt weer de Idle state bezocht.
6. **Sluiten**: deze state wordt bezocht wanneer een trein gedecteerd is, het stoplicht wordt rood voor de verkeersdeelnemers en ook gaan de gele lampen van beide stoplichten knipperen.
7. **Passeren**: wanneer de slagboom gesloten is blijft het systeem in deze state totdat de trein weer gepasseerd is.
8. **Openen**: wanneer de trein gepasseerd is opent de slagboom weer, de lampen blijven op rood en de gele lampen blijven knipperen, ook maakt de buzzer een snel tikkend geluid tijdens dit process.
9. **Vrijgeven**: wanneer de slagboom volledig geopend is wordt op de display van 5 naar 0 afgeteld, de buzzer maakt een tikkend geluid met na 3 tikken maken een korte pauze, de lampen blijven op rood en de gele lampen blijven ook knipperen,

## Softwareontwerp (tabbladendiagram)



In dit klasse diagram zijn 2 packages, de “SpoorwegOvergang” package bevat alle modules die gebruik maken van de “Library” modules om het programma zoals hoort te laten functioneren, de “Library” package bevat alle modules die voornamelijk direct communiceren met de hardware en ook de setup functies bevat samen met de pin declaraties van de hardware. In het volgende hoofdstuk wordt elke klasse omschreven en wordt ook uitgelegd wat de functionaliteit is van de verschillende methodes erbinnen.

## Toelichting bij de tabbladen en functies hierbinnen

### 2.3.1 Highlevel-modules

Deze modules bepalen het gedrag van het systeem, de modules maken vooral gebruik van de **low-level** modules om de hardware aan te sturen.

### Spoorwegovergang

Deze module voert de setup functie uit van alle modules vanwaar het nodig is, ook wordt in deze module de hoofd loop lus uitgevoerd.

* **setup()**

Deze functie bevat de setup functie van alle modules binnen het programma.

* **loop()**

Binnen deze loop wordt het programma constant gedraaid.

### Statemachine

Deze module bevat de statemachine van de spoorwegovergang, hierin wordt bepaalt in welke toestand het systeem is op het huidige moment en aan de hand van inkomende events wordt bepaald of het systeem in de huidige toestand blijft of dat het systeem naar een andere toestand moet.

* **spoorweg\_Setup()**

Deze functie zet het systeem in de eerste state “Idle”.

* **bepaalStates()**

In deze functie bevind de statemachine en wordt geschakeld tussen alle mogelijke states, dit wordt bepaald aan de hand van inkomende events.

### Treinen

Deze module bevat verwijst naar 2 drukknoppen die 2 richtingen simuleren waarvan treinen vandaan komen

* **treinGedecteerd()**

Deze functie retourneerd een boolean waarde die bepaald wordt of er een knop op de druk gedecteerd is.

* **treinGepasseerd()**

Wanneer de drukknop tegenover de drukknop waar de trein vandaan kwam gedecteerd wordt dan retourneert deze methode een true terug.

* **treinResetHasEvent()**

Zolang een druk op de knop van een trein niet afgehandeld is dan kan de **treinGedecteerd()** methode geen true teruggeven, deze methode wordt aangeroepen wanneer een event van een drukknop afgehandeld is.

### VerkeersdeelnemerDrukknoppen

Deze module verwijst naar de 2 drukknoppen die tegenover elkaar staan, de drukknoppen worden gebruikt door verkeersdeelnemers, deze module bevat net als de trein modules **getters** om op te vragen of de knop ingedrukt is geweest en ook zijn er reset functies om een event af te handelen wanneer deze voorbij is.

### Slagboom

Deze module maakt gebruik van **libServo** om de slagboom te besturen en om de statemachine informatie terug te geven over de toestand van de slagboom.

* **slagboomGesloten()**

Deze functie retourneert true of false aan de hand van de toestand van de slagboom.

* **slagboomGeopend()**

Deze functie retourneert true of false aan de hand van de toestand van de slagboom.

* **slagboomVerhoogTeller()**

Deze functie verhoogd de teller die bijhoudt hoe vaak de slagboom geopend is sinds het begin dat het systeem is aangezet.

* **slagboomResetTeller()**

Wanneer deze functie aangeroepen wordt, wordt de slagboomteller op 0 gezet en wordt ook het EEPROM geheugenadres ervan naar deze waarde gezet

* **slagboomOpenen()**

Deze functie maakt gebruik van de **servoMove()** functie binnen de **libServo** module, de maximale waarde dat de servomotor kan openen wordt in de servoMove functie gestopt.

* **slagboomSluiten()**

Deze functie doet het zelfde als de **slagboomOpenen()** functie alleen wordt de minimale waarde dat de servomotor kan openen naar de **servoMove()** functie toegestuurd.

### Stoplicht

Deze module bevat constanten die verwijzen naar de pin van de bijbehorende stoplicht LED, in deze module bevinden zich functies die gebruik maken van **libLed** om de stoplicht lampen aan/uit te zetten of te laten knipperen.

*Alle 3 de lampen van beide stoplichten bevatten een on/off functie om de bettreffende lamp te schakelen.*

* **stoplichtNoordGeelKnipper() / stoplichtZuidGeelKnipper()**

Allebei deze functies knipperen de bettreffende gele lamp van de bijbehorende stoplicht.

### 2.3.2 Low-level Modules

Deze modules communiceren voornamelijk direct met de hardware, de **high-level** modules maken gebruik van deze modules om de hardware aan te sturen.

### libSerial

Deze module stuurt en ontvangt informatie via de seriële poort van de Arduino

* **serialSetup()**

Deze functie bepaalt de baud-rate.

* **serialRead()**

Deze functie leest de eerstvolgende ontvangen byte via de seriële poort.

### libLed

Deze module LED lampjes aan/uit en bevat ook een functie om de LED te laten knipperen in een zelfbepaalde interval.

* **ledSetup()**

-

* **ledOn(int ledNr) / ledOff(int Nr)**

Deze functies zetten de led aan/uit aan de hand van de inkomende pin in de parameters.

* **ledBlink(int interval)**

Deze functie zet de betreffende led aan/uit aan de hand van de inkomende interval in de parameters.

### libServo

Deze module bestuurt de servomotor die in deze opdracht een slagboom simuleert.

* **servoSetup()**

Deze functie reset de timer en initialiseert de servomotor.

* **servoMove(int nieuwPositie)**

Deze functie beweegt de servomotor omhoog/omlaag aan de hand van de directie de bepaald wordt aan de hand van de inkomende parameter

* **servoPosIsMin() / servoPosIsMax()**

Deze functies retourneren true of false aan de hand van de positie van de servomotor.

### libPatronen

Deze module bevat de bitmaps die nodig zijn om getallen te tekenen op de 7-segment display.

* **patronenGetSevenSegment0Tot10(int index)**

Deze functie retourneert de juiste byte aan de hand van de inkomende index in de parameters.

### libShiftRegister

Deze module stuurt direct de inkomende bytes naar de shift-register, aan de hand van deze bytes wordt een karakter op de display getoond.

* **shiftRegisterSetup()**

Deze functie wist de byte die nog in de shiftregister kan bevinden en zet deze naar 0.

* **shiftRegisterSet(byte pattern)**

Deze functie ontvangt in de parameters een byte en deze wordt direct verstuurt en geschoven in de shiftregister.

* **shiftRegisterClear()**

Deze functie zet de shiftregisterbyte op 0

### libDisplay

Deze module maakt gebruik van de **libShiftRegister** module om de 7-segment display aan te sturen

* **displaySetup()**

deze functie maakt een call naar **displayClear()** zodat de display niks toont bij het opstarten van het systeem.

* **displayClear()**

Deze functie zet de byte die verstuurd wordt naar de display op 0 om niks te tonen.

* **displayNumber(int getal)**

Deze functie ontvangt een getal in de parameters en door gebruik te maken van **shiftRegisterSet()** en **patronenGetSevenSegment0Tot10** om het juiste getal op de display te tonen.

* **displaySetStartNumberToCountdownFrom(int getal**)

Deze functie zet de startwaarde vanwaar de countdown op de seven-segment display begint.

* **displayCountdown()**

Deze functie start een countdown vanaf een getal wat bepaald wordt door de vorige functie, elke keer wanneer het getal naar het shiftregister gestuurd wordt, wordt de waarde van het getal wat getoond wordt met 1 verlaagd.

### libBuzzer

Deze module bestuurt de buzzer.

* **buzzerSetup()**

Deze functie zet de buzzerpin als output.

* **buzzerOn(int frequentie)**

Deze functie zet de buzzer aan, aan de hand van de frequentie die ontvangen wordt binnen de parameters.

* **buzzerOff()**

Deze functie zet de buzzer uit.

* **buzzerTik(int interval)**

Deze functie laat de buzzer tikken aan de hand van de interval ontvangen binnen de parameters.

### libTimer

Deze module bevat de functionaliteit om een timer te initialiseren, resetten en om op te vragen of een bepaalde timer voorbij is.

* **timerSetup(unsigned long timer)**

Deze functie initialiseert een timer naar 0.

* **timerReset()**

Deze functie retourneert de huidige tijd om een timer mee te resetten.

* **timerIntervalVoorbij(unsigned long interval)**

Binnen deze functie bevind een if/else constructie waarmee bepaald wordt of de timer die ontvangen wordt binnen de parameters voorbij is.

### libEEPROM

Deze module wordt gebruikt om data te schrijven naar een byte EEPROM geheugen binnen de Arduino, in dit geval wordt de slagboomTeller opgeslagen zodat deze wanneer het systeem opstart onthouden wordt.

* **EEPROMlees(int adres)**

Deze functie ontvangt een locatie van de EEPROM waaruit gelezen moet worden en retourneert deze waarde.

* **EEPROMschrijf(int adres, int waarde)**

Deze functie schrijft een waarde naar de toegewezen locatie die bepaalt wordt aan de hand van de ontvangen waarde binnen de parameters.

### Button

Deze module bevat een statemachine die ervoor zorgt dat de knop niet “dendert”. 2 van de knoppen die gebruik maken van deze module simuleren een trein, 2 knoppen simuleren een drukknop voor verkeersgebruikers van de spoorweg.

* **button\_Setup()**

Deze functie zorgt ervoor dat elke knop de start met de toestand **BUTTON\_STATE\_RELEASED.**

* **button\_Clicked()**

Deze functie retourneert true of false aan de hand van of de knop die ontvangen wordt in de parameters ingedrukt is.

* **button\_Loop()**

Deze functie wordt aangeroepen in de **loop()** functie van de hoofdmodule om de statemachine te laten functioneren, deze functie loopt m.b.v. een iterator alle knoppen langs en aan de hand van of een knop ingedrukt is of niet wordt binnen een boolean array vastgehouden of de knop afgehandeld moet worden of niet.

* **buttonRememberedClick(int directie)**

Deze functie retourneert een boolean waarde aan de hand van of een knop ingedrukt is, zolang de knop niet afgehandeld is kan deze waarde niet weer op true gezet worden.

* **buttonResetClickHasEvent(int directie)**

Deze functie is nodig om een event van een drukknop af te handelen, zolang een event niet afgehandeld is kan de staat van de knop niet zomaar op true gezet worden.

### ButtonHardware

Deze module communiceert direct met de drukknoppen verbonden aan de Arduino, deze module wordt gebruik door de **Button** module.

* **buttonHardware\_Setup()**

Deze functie initialiseert alle knoppen als input.

* **button\_Hardware\_Down(int pin) / button\_Hardware\_Up(int pin)**

Deze functies retourneren true of false aan de hand van of de knop ingedrukt is, deze functie wordt gebruikt door de **Button** module binnen de statemachine ervan.

## Aansluitschema (Tinkercad)

Aansluitingen per Arduino pin:

13 Rode LED Stoplicht Noord

12 Gele LED Stoplicht Noord

11 Groene LED Stoplicht Noord

10 Shiftregister Clock

9 Shiftregister Latch

8 Shiftregister Data

7 Rode LED Stoplicht Zuid

6 Rode LED Stoplicht Zuid

5 Rode LED Stoplicht Zuid

4 -

3 Servomotor

2 Buzzer

A0 Button 0 – drukknop Noord

A1 Button 1 – drukknop Zuid

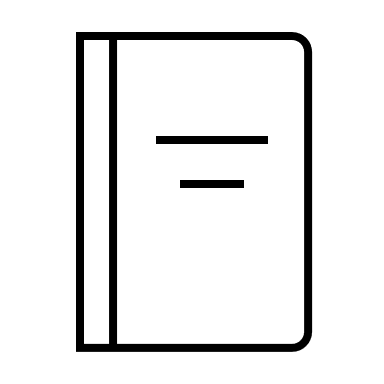
A2 Button 2 – trein Oost

A3 Button 3 – trein West

# Conclusie

Ik vond dit een erg leuke opdracht om te maken, het proces ging behoorlijk soepel. Ik had wel veel moeite met schakelen tussen dit vak en OOPD omdat ik soms graag gebruik had willen maken van OOPD concepten, zoals bij de knoppen die we tijdens deze opdracht gebruikten, ik had het gevoel dat het veel simpeler geweest was als we Button objecten konden instantiëren en elk van deze een eigen state te geven zou hier makkelijker zijn. Verder vond ik deze opdracht erg leerzaam en ik ben best tevreden over wat ik allemaal geleerd heb tijdens het maken ervan.

Bijlagen

A. Zelfbeoordelingsformulier

Algemene info

Dit formulier gaat over het werk van:

Naam: Daniël Roth

Studentnummer: 496885

Klas: ITA-1DB (laat leeg bij herkansing)

Het formulier is ingevuld op datum: 06-04-2022

Knock-out-criteria

* **Delay** Voldoende
* **Uitprogrammeren statemachine** Voldoende
* **Controlekaart documenten** Voldoende
* **Implementeert eisen** Voldoende

Functioneel ontwerp

* **Requirements** Voldoende
* **Prioritering** Voldoende

Technisch ontwerp

* **Tabbladendiagram** Voldoende
* **Toelichting functies en globale variabelen** Voldoende
* **Statemachine-diagram** Voldoende
* **Aansluitschema** Voldoende

Software schrijven en testen

* **Volgt ontwerp** Voldoende
* **Codekwaliteit** Voldoende
* **Testen** Voldoende

Sensoren en actuatoren

* **Aansluiting hardware** Voldoende
* **Traagheid actuatoren** Voldoende