E. Knaake (598368)

Vak: ESD  
Datum: 4 april 2018  
Versie 1.2

Ontwerp

Eindopdracht verkeerssimulatie

Inhoud

[Inleiding 2](#_Toc510520773)

[Functioneel ontwerp 3](#_Toc510520774)

[Specificaties 3](#_Toc510520775)

[1. Algemeen 3](#_Toc510520776)

[2. Door laten verkeer 4](#_Toc510520777)

[3 Nachtstand 5](#_Toc510520778)

[4 Verkeerslicht 5](#_Toc510520779)

[5 Testen 6](#_Toc510520780)

[MoSCoW analyse 6](#_Toc510520781)

[Technisch ontwerp 9](#_Toc510520782)

[Hardware 9](#_Toc510520783)

[Benodigde hardware 9](#_Toc510520784)

[Schakeling 9](#_Toc510520785)

[Statemachine 10](#_Toc510520786)

[Gebruikte enumeraties 11](#_Toc510520787)

[States 11](#_Toc510520788)

[TikkenStates 11](#_Toc510520789)

[TestFase 11](#_Toc510520790)

[ButtonStates 11](#_Toc510520791)

[Tabbladentabel 11](#_Toc510520792)

[Tabbladendiagram 24](#_Toc510520793)

[Figuren 24](#_Toc510520794)

# Inleiding

De opdrachtgever heeft gevraagd om de voetgangersoversteekplaats in Gendt veiliger te maken. Om dit te bereiken, worden er stoplichten, een slagboom, een ratelapparaat en een voetgangerslicht geïntroduceerd. Omdat veiligheid hier een grote rol speelt, wordt er eerst een functioneel ontwerp gemaakt om te controleren of alle specificaties goed geïnterpreteerd zijn. Vervolgens wordt er een technisch ontwerp gemaakt, zodat er niets vergeten kan worden tijdens het implementeren. Tot slot worden alle specificaties getest en waar nodig worden problemen verholpen.

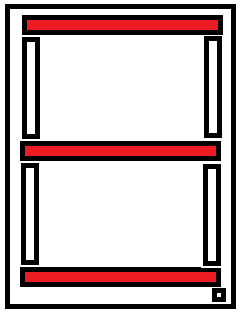
# Functioneel ontwerp

## Specificaties

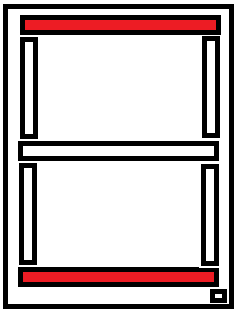
In dit hoofdstuk worden alle specificaties beschreven. Omdat er vrij veel specificaties zijn worden ze opgedeeld in vijf delen. Namelijk: algemeen, door laten verkeer, nachtstand, verkeerslicht en testen.

### [Algemeen](#S1)

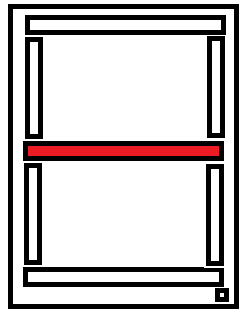
*De volgende specificaties zijn altijd van kracht.*

* 1. Er kan maximaal één richting tegelijk door mogen, onder richtingen wordt verstaan: de beide verkeerslichten en de voetgangersoversteek.
  2. Onder alle richtingen zijn dicht wordt verstaan: de verkeerslichten staan op ‘stoppen’, de slagboom is dicht, het voetgangerslicht toont het teken ‘wachten’ en de buzzer tikt op 2 Hz.
  3. Als een richting sluit (verkeerslicht naar ‘stoppen’ of voetgangersoversteek naar ‘wachten’) is er een dode tijd van drie seconden, in deze tijd kan er geen nieuwe richting opengaan. Deze tijd is er om verkeer de tijd te geven de oversteekplaats te verlaten voordat er nieuw verkeer aan komt.
  4. De slagboom roteert met een snelheid van 15 graden per seconde.
  5. De slagboom heeft een maximale uitslag van 90 graden.
  6. De slagboom heeft een minimale uitslag van 0 graden.
  7. Button 1 hoort bij verkeerslicht 1.
  8. Button 2 hoort bij verkeerslicht 2.
  9. Button 3 hoort bij de voetgangersoversteek.
  10. Button 4 is de testknop.
  11. Knoppen hoeven slechts kort te worden ingedrukt – techniek tegen floating.
  12. Knoppen worden onthouden (om als de huidige handeling voorbij is de volgende knop al te weten).
  13. Met het symbool ‘wachten’ wordt het volgende symbool bedoeld:

Figuur 1

* 1. Onder het symbool ‘let extra op’ wordt het volgende symbool verstaan:

Figuur 2

* 1. Het symbool ‘oversteken’ ziet als volgt uit: 

Figuur 3

*Voor de getallen worden dezelfde symbolen gebruikt als in weekopdracht 3.*

### [Door laten verkeer](#S2)

*De volgende specificaties gaan alleen over het door laten van verkeer, voor het gemak is dit verder onderverdeeld in oversteken voor voetgangers en doorrijd voor auto’s.*

#### 2.1. [Voetgangers](#S2_1)

*De volgende specificaties worden pas uitgevoerd als button 3 (voetgangersknop) is ingedrukt.*

* + 1. De slagboom gaat open naar de maximale uitslag.

*De volgende specificaties worden pas uitgevoerd als de slagboom uit is bewogen.*

* + 1. Het voetgangerslicht laat het teken ‘oversteken’ zien.
    2. De buzzer maakt 4 seconden lang een geluid op 60 Hz.

*De volgende specificaties worden uitgevoerd als de buzzer is uit gepiept.*

* + 1. Het voetgangers licht telt af van 6 naar 0 seconden.
    2. Na elke 3 tikken wacht de buzzer 0,5 seconde.

*De volgende specificaties worden pas uitgevoerd als het voetgangerslicht 0 aangeeft.*

* + 1. De slagboom sluit (naar de minimale uitslag).
    2. De buzzer tikt op 2 Hz.
    3. Het voetgangerslicht laat het teken ‘wacht’ zien.

#### 2.2. [Auto’s](#S2_2)

*De volgende specificaties worden pas uitgevoerd als button 1 of button 2 (auto’s) is ingedrukt.*

* + 1. Verkeerslicht aan de kant behorende bij de knop gaat naar overgang tussen ‘stoppen’ en ‘rij door’. *Deze specificatie wordt alleen uitgevoerd als deze overgang er is.*
    2. Verkeerslicht aan de kant behorende bij de knop gaat naar ‘rij door’.
    3. Verkeerslicht aan de kant behorende bij de knop gaat naar overgang tussen ‘rij door’ en ‘stoppen’.
    4. Verkeerslicht aan de kant behorende bij de knop gaat naar ‘stoppen’.

### [Nachtstand](#S3)

*De volgende specificaties worden uitgevoerd als het donker wordt (LDR-spanningsdeler onder 300/1023). (Naar nachtstand).*

* 1. De slagboom gaat open (naar maximale uitslag).
  2. De buzzer tikt op 2 Hz.
  3. Het voetgangerslicht toont het teken ‘wachten’.

*De volgende specificaties worden uitgevoerd als het systeem in de nachtstand zit.*

* 1. Het voetgangerslicht toont het teken ‘let extra op’.
  2. Beide verkeerslichten knipperen op geel met een frequentie van 2 Hz.
  3. De buzzer staat uit.

*De volgende specificaties worden uitgevoerd als het licht wordt (LDR-spanningsdeler boven 350/1023). (Naar dag stand).*

* 1. Het voetgangerslicht toont het teken ‘wachten’.
  2. De buzzer tikt op 2 Hz.
  3. De verkeerslichten staan op geel.
  4. De slagboom sluit (naar minimale uitslag).

*De volgende specificaties worden uitgevoerd als het systeem in de ruststand staat (het is licht, alle richtingen zijn dicht).*

* 1. Alle richtingen zijn dicht.

## [Verkeerslicht](#S4)

* + 1. Als er een overgang van ‘stoppen’ naar ‘rij door’ is, staat die overgang 0,5 seconde aan.
    2. Een verkeerlicht staat 5 seconden op ‘rij door’.
    3. Een verkeerslicht staat 1 seconde op de overgang van ‘rij door’ naar ‘stoppen’
    4. Er is altijd precies 1 soort verkeerslicht actief.
    5. Het programma begint met een Nederlands verkeerslicht.
    6. Het Nederlandse verkeerslicht wordt geactiveerd na een ‘N’ op de seriële input.
    7. Het Duitse verkeerslicht wordt geactiveerd na een ‘D’ op de seriële input.
    8. Het Oostenrijkse verkeerslicht wordt geactiveerd na een ‘O’.

#### 4.2. [Nederlandse verkeerslicht](#S4_2)

* + 1. Geen overgang van ‘stoppen’ naar ‘rij door’.
    2. Bij ‘stoppen’ staat het rode licht aan.
    3. Bij ‘rij door’ staat het groene licht aan.
    4. Overgang van ‘rij door’ naar ‘stoppen’: gele licht staat aan.

#### [Duitse verkeerslicht](#S4_3)

* + 1. Overgang van ‘stoppen’ naar ‘rij door’: rood en geel staan aan.
    2. Bij ‘stoppen’ staat het rode licht aan.
    3. Bij ‘rij door’ staat het groene licht aan.
    4. Overgang van ‘rij door’ naar ‘stoppen’: gele licht staat aan.

#### [Oostenrijkse verkeerslicht](#S4_4)

* + 1. Overgang van ‘stoppen’ naar ‘rij door’: rood en geel staan aan.
    2. Bij ‘stoppen’ staat het rode licht aan.
    3. Bij ‘rij door’ staat het groene licht aan.
    4. Overgang van ‘rij door’ naar ‘stoppen’:
       1. Groene licht knippert op 2 Hz voor 1 seconde.
       2. Gele licht staat aan. *Deze specificatie wordt uitgevoerd als het groene licht niet meer knippert, deze specificatie telt voor de 1 seconde overgangstijd die eerder is beschreven. De specificatie hierboven telt voor die tijd niet mee.*

## [Testen](#S5)

*De volgende specificaties worden uitgevoerd als button 4 (test knop) is ingedrukt.*

* 1. Verkeerslichten zetten alle lampen een voor een 1 seconde lang aan. Dit gebeurt met een pauze van 0,5 seconde tussen alle lampen.
  2. De buzzer klinkt op 60 Hz voor 2 seconden.
  3. De slagboom gaat 15 graden omhoog.

*De volgende specificatie wordt uitgevoerd als de slagboom op 15 graden staat.*

* 1. De slagboom gaat dicht.

## MoSCoW analyse

Hier volgen de specificaties nogmaals, maar nu volgens het MoSCoW principe. Dit betekent dat de belangrijkste specificaties een M van must krijgen, gevolgd door een S van should, een C van could en een W van would. Deze tabel bevat al ruimte voor de test resultaten.

|  |  |  |  |
| --- | --- | --- | --- |
| Specificatie | Beschrijving | MoSCoW | Test resultaat |
| [1.1](#F1) | Max. een richting door. | M | Geslaagd. |
| 1.2 | Alle richtingen dicht. | M | N.v.t |
| 1.3 | Er is een dode tijd van 3 seconden. | M | Geslaagd. |
| 1.4 | Slagboom draait op 15 graden per seconden. | M | Geslaagd. |
| 1.5 | Maximale uitslag v.d. slagboom is 90 graden. | M | Geslaagd. |
| 1.6 | Minimale uitslag v.d. slagboom is 0 graden. | M | Geslaagd. |
| 1.7 | Button 1 hoort bij verkeerslicht 1. | M | N.v.t |
| 1.8 | Button 2 hoort bij verkeerslicht 2. | M | N.v.t |
| 1.9 | Button 3 hoort bij de voetgangersoversteek. | M | N.v.t |
| 1.10 | Button 4 is de testknop. | M | N.v.t |
| 1.11 | Knoppen hoeven slechts kort ingedrukt te worden. | M | Geslaagd. |
| 1.12 | Ingedrukte knoppen worden onthouden tot verwerking. | M | Geslaagd. |
| 1.13 | Het symbool ‘wachten’. | M | Geslaagd. |
| 1.14 | Het symbool ‘let extra op’. | M | Geslaagd. |
| 1.15 | Het symbool ‘oversteken’. | M | Geslaagd. |
| [2.1.1](#F2_1) | Slagboom gaat open. | M | Geslaagd. |
| 2.1.2 | Symbool ‘oversteken’ wordt getoond. | M | Geslaagd. |
| 2.1.3 | Buzzer maakt geluid van 60 Hz voor 2 seconden. | M | Geslaagd. |
| 2.1.4 | Aftellen voetgangerslicht. | M | Geslaagd. |
| 2.1.6 | Na elke 3 tikken wordt er 0,5 seconden gewacht. | M | Geslaagd. |
| 2.1.7 | Slagboom sluit. | M | Geslaagd. |
| 2.1.8 | Buzzer tikt op 2 Hz. | M | Geslaagd. |
| 2.1.9 | Voetgangerslicht toont ‘wachten’. | M | Geslaagd. |
| [2.2.1](#F2_2) | Toon overgang van ‘rij door’ naar ‘stoppen’ (mits aanwezig) | S | Geslaagd. |
| 2.2.2 | Toon ‘rij door’ | M | Geslaagd. |
| 2.2.3 | Toon overgang van ‘stoppen’ naar ‘rij door’. | M | Geslaagd. |
| 2.2.4 | Toon ‘stoppen’. | M | Geslaagd. |
| [3.1](#F3) | Slagboom gaat open. | M | Geslaagd. |
| 3.2 | Buzzer tikt op 2 Hz. | M | Geslaagd. |
| 3.3 | Voetgangerslicht toont ‘wachten’. | M | Geslaagd. |
| 3.4 | Voetgangerslicht toont ‘Let extra op’. | M | Geslaagd. |
| 3.5 | Verkeerslichten knipperen geel op 2 Hz. | M | Geslaagd. |
| 3.6 | Buzzer staat uit. | M | Geslaagd. |
| 3.7 | Voetgangerslicht toont ‘wachten’. | M | Geslaagd. |
| 3.8 | Buzzer tikt op 2 Hz. | M | Geslaagd. |
| 3.9 | Verkeerslichten staan op geel. | M | Geslaagd. |
| 3.10 | Slagboom sluit. | M | Geslaagd. |
| 3.11 | Alle richtingen zijn dicht. | M | Geslaagd. |
| [4.1.1](#F4) | Overgang ‘stoppen’ naar ‘rij door’ duurt 0,5 seconde. | S | Geslaagd. |
| 4.1.2 | ‘Rij door’ duurt 5 seconden. | M | Geslaagd. |
| 4.1.3 | Overgang van ‘rij door’ naar ‘stoppen’ duurt 2 seconden. | M | Geslaagd. |
| 4.1.4 | Er is altijd 1 soort verkeerslicht actief | M | Geslaagd. |
| 4.1.5 | Het programma begint met het Nederlands licht | M | Geslaagd. |
| 4.1.6 | Bij een ‘N’ op de seriële input wordt het Nederlandse licht actief. | S | Geslaagd. |
| 4.1.7 | Bij een ‘D’ op de seriële input wordt het Duitse licht actief. | S | Geslaagd. |
| 4.1.8 | Bij een ‘O’ op de seriële input wordt het Oostenrijkse licht actief. | S | Geslaagd. |
| [4.2.1](#F4_2) | Geen overgang van ‘rij door’ naar ‘stoppen’. | M | Geslaagd. |
| 4.2.2 | Bij ‘stoppen’ rood licht. | M | Geslaagd. |
| 4.2.3 | Bij ‘rij door’ groen licht. | M | Geslaagd. |
| 4.2.4 | Overgang ‘rij door’ naar ‘stoppen’ geel licht. | M | Geslaagd. |
| [4.3.1](#F4_3) | Overgang van ‘stoppen’ naar ‘rij door’: rood en geel staan aan. | C | Geslaagd. |
| 4.3.2 | Bij ‘stoppen’ staat het rode licht aan. | C | Geslaagd. |
| 4.3.3 | Bij ‘rij door’ staat het groene licht aan. | C | Geslaagd. |
| 4.3.4 | Overgang van ‘rij door’ naar ‘stoppen’: gele licht staat aan. | C | Geslaagd. |
| 4.4.1 | Overgang van ‘stoppen’ naar ‘rij door’: rood en geel staan aan. | C | Geslaagd. |
| [4.4.2](#F4_4) | Bij ‘stoppen’ staat het rode licht aan. | C | Geslaagd. |
| 4.4.3 | Bij ‘rij door’ staat het groene licht aan. | C | Geslaagd. |
| 4.4.4.1 | Overgang van ‘rij door’ naar ‘stoppen’:  Groene licht knippert op 2 Hz voor 1 seconde. | C | Geslaagd. |
| 4.4.4.2 | Het gele licht staat aan. | C | Geslaagd. |
| [5.1](#F5) | Verkeerslichten zetten alle lampen een voor een 1 seconde lang aan. Dit gebeurt met een pauze van 0,5 seconde tussen alle lampen. | S | Geslaagd. |
| 5.2 | De buzzer klinkt op 60 Hz voor 2 seconden. | S | Geslaagd. |
| 5.3 | De slagboom gaat 15 graden omhoog. | S | Geslaagd. |
| 5.4 | De slagboom gaat weer dicht. | S | Geslaagd. |

Tabel 1

# Technisch ontwerp

Om ervoor te zorgen dat de requirements zoals deze zijn beschreven in functioneel ontwerp goed geïmplementeerd worden, volgt hier technisch ontwerp. Omdat er gekozen is om dit product te maken met behulp van een statemachine, wordt in dit ontwerp beschreven welke *states*, *constanten* en *functies* er zijn. Het beschrijven van de *states* gebeurt aan de hand van een statemachine diagram. Terwijl de constanten en functies in een tabbladentabel worden ondergebracht. Om tot deze uitwerkingen te komen, wordt eerst de hardware beschreven.

## Hardware

Om tot goede software te komen moet eerst bekend zijn welke hardware er aangesloten wordt, maar vooral ook hoe deze aangesloten moet worden. Hiervoor is in Fritzing een tekening gemaakt.

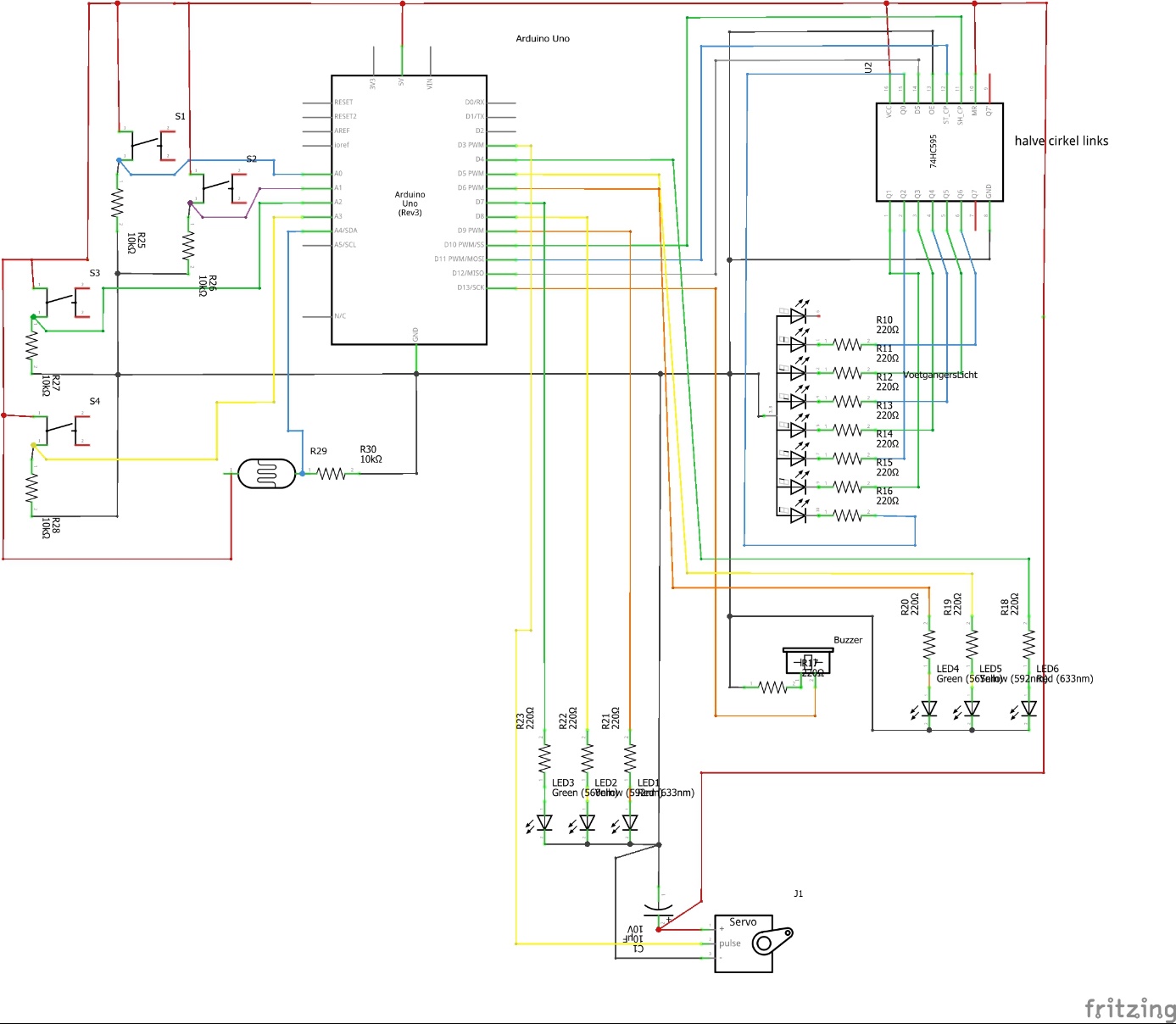
### Benodigde hardware

|  |  |  |
| --- | --- | --- |
| **Component** | **Aantal** | **Benodigd voor:** |
| Rode led | 2 | Verkeerslicht |
| Gele led | 2 | Verkeerslicht |
| Groene led | 2 | Verkeerslicht |
| 7-segmentdisplay | 1 | Voetgangerslicht |
| Register | 1 | Aansturen 7-segmentdisplay |
| Buzzer | 1 | Geluiden afhankelijk van de state. |
| Servomotor | 1 | Slagboom |
| Button | 4 | Events genereren. |
| LDR | 1 | Nacht of dag stand |
| 220 Ω weerstand | 14 | Leds, 7-segmentdisplay, buzzer |
| 10 kΩ weerstand | 5 | Knoppen, LDR |

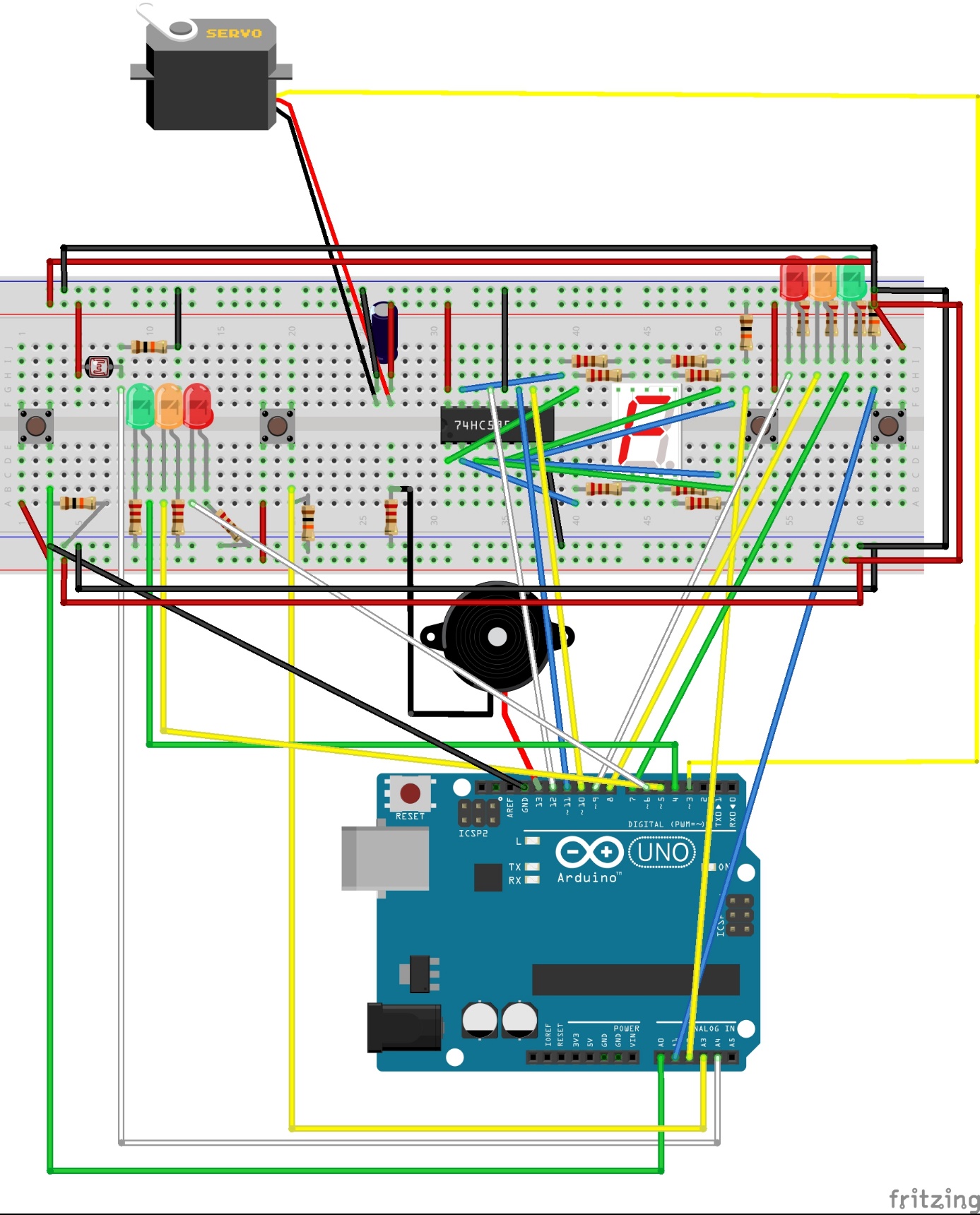
Tabel 2

### Schakeling

In het volgende schema is de aansluiting schematisch weergegeven. Dit schema vindt u ook in de bijlagen.

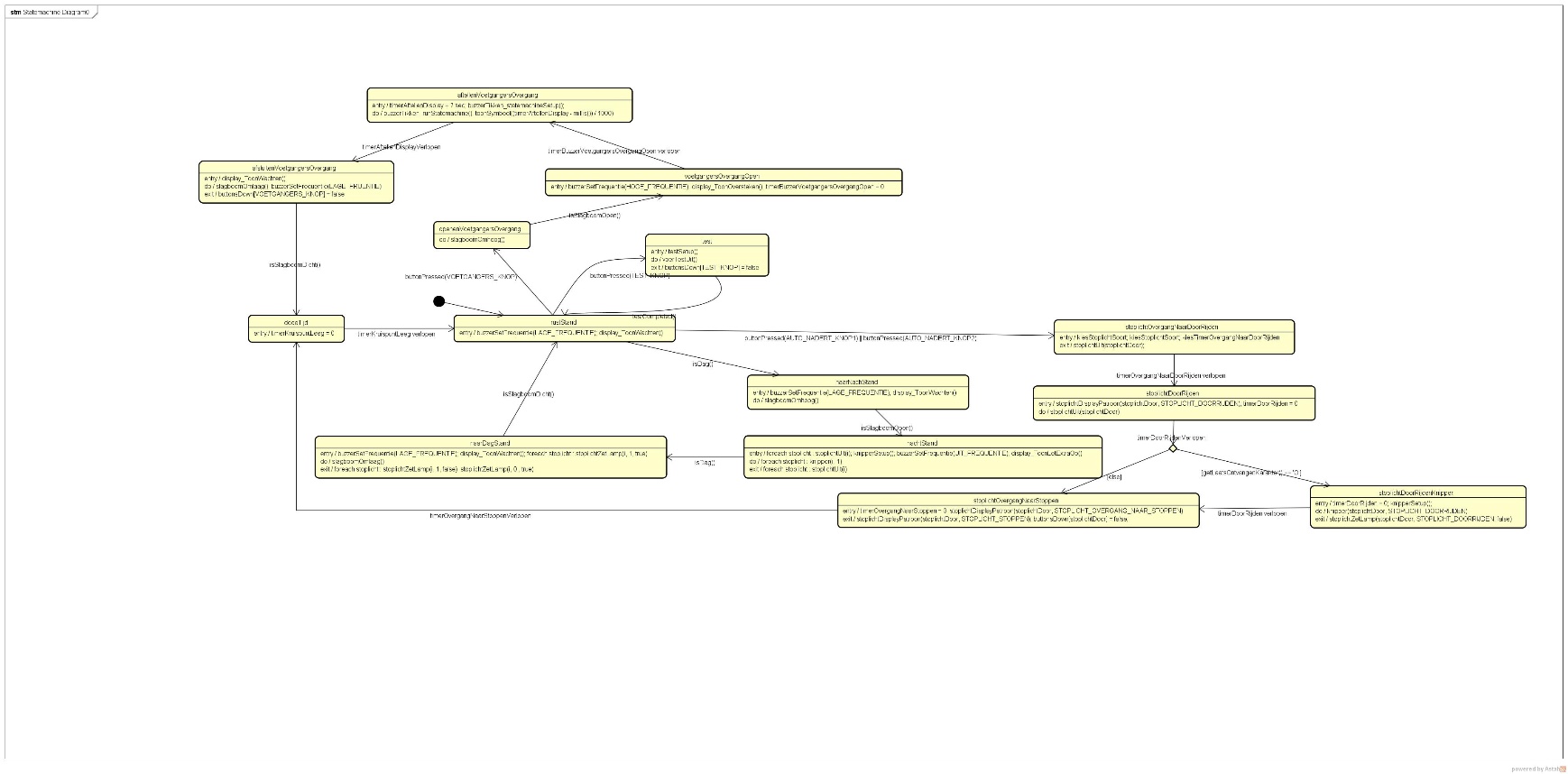


Figuur 4



Figuur 5

## Statemachine

Er is gekozen om de statemachine zo compleet mogelijk te tekenen. Dit zorgt ervoor dat dit diagram alle informatie voor het bouwen van de statemachine geeft. Omdat het moeilijk is om dit diagram weer te geven in Word, zit deze ook in de bijlagen.

Figuur 6

Alle events spreken in dit diagram voor zich. De acties die worden uitgevoerd op de entry, do en exit worden niet beschreven in de tabbladentabel.

## Gebruikte enumeraties

Om de states van de statemachine aan te geven is er gekozen om een aantal enumeraties te gebruiken. Deze enumeraties worden hier kort beschreven.

### States

Bevat alle states die vanaf de hoofdstatemachine bereikt kunnen worden: rustStand, naarNachtstand, nachtStand, naarDagStand, dodeTijd, openenVoetgangersOvergang, voetgangersOvergangOpen, aftellenVoetgangersOvergang, afsluitenVoetgangersOvergang, stoplichtOvergangNaarDoorRijden, stoplichtDoorRijden, stoplichtDoorRijdenKnipper, stoplichtOvergangNaarStoppen en test.

### TikkenStates

Deze enum is bedoeld om bij te houden in welk deel van het drietikken proces het programma is. Deze enum bevat: drieTikken en pauze.

### TestFase

Bevat de verschillende delen van het testproces, hieronder vallen: LAMP1, LAMP2, BUZZER\_FASE en SLAGBOOM.

### ButtonStates

Bevat de verschillende states die een button kan hebben om bouncing/floating te voorkomen, namelijk: RELEASED, CHECK\_PRESSED en PRESSED.

## Tabbladentabel

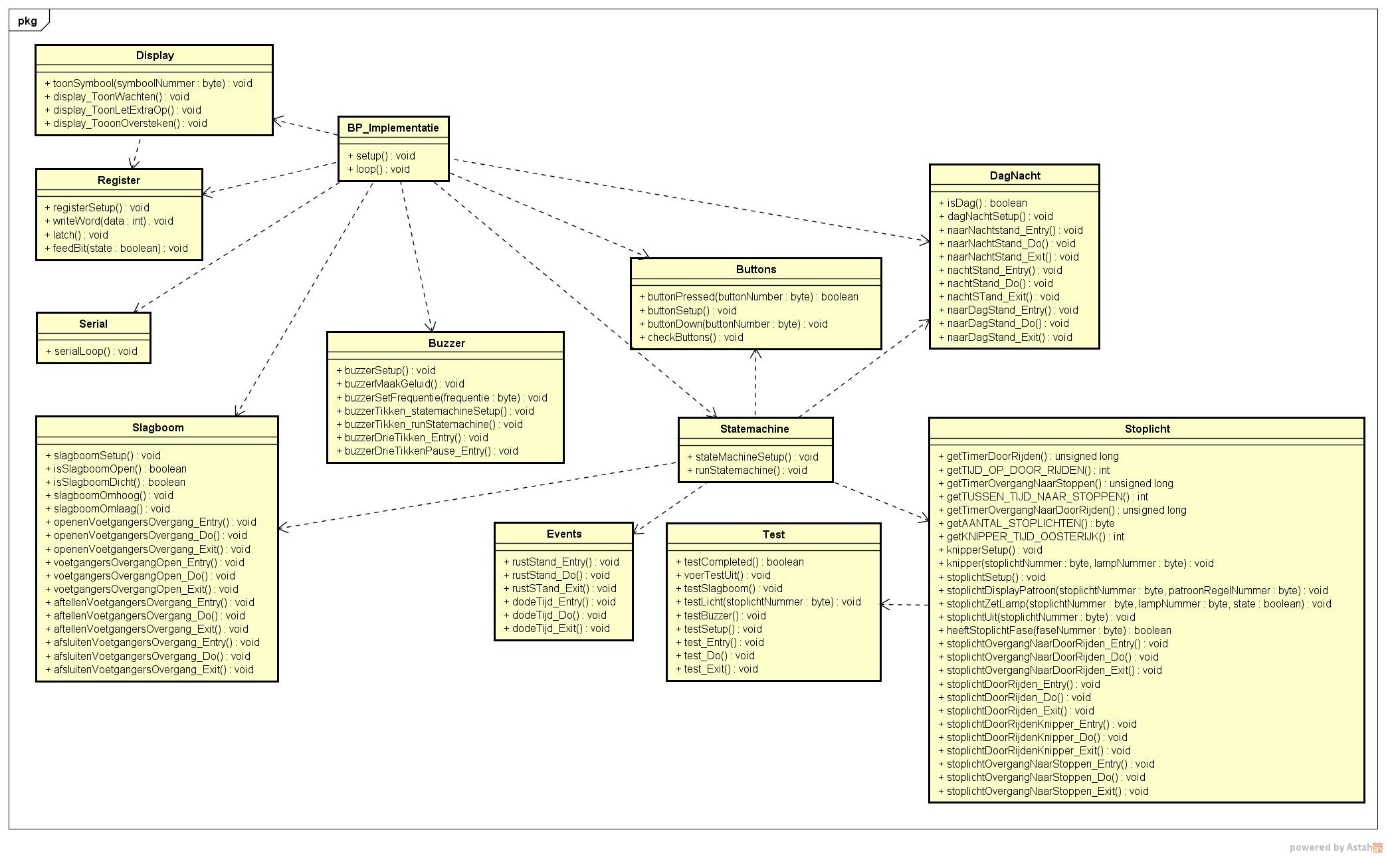
In de volgende tabel wordt telkens een tabblad van code beschreven, met alle variabelen en functies die daarbij horen. Uiteraard worden de functies en variabelen ook toegelicht. Entry, Do en Exit functies zijn hierin niet opgenomen, omdat deze af te leiden zijn uit het statemachinediagram.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabblad** | **Datatype** | **Functie/variabele** | **Uitleg** |
| BP\_Implementatie |  |  | Bevat de standaard Arduino functies setup() en loop() |
|  | void | setup() | Loopt alle andere setup functies af. |
|  | void | loop() | Loop achtereenvolgens checkButtons(), runStateMachine(), serialLoop() en buzzerMaakGeluid() af. |
| Buttons | Const byte | AANTAL\_BUTTONS = 4 | Het aantal aangesloten knoppen. |
|  |  | BUTTONS[AANTAL\_ BUTTONS] = {A0, A1, A2, A3} | Bevat de pinmapping van de buttons |
|  |  | AUTO\_NADERT\_KNOP1 | Index die hoort bij de eerste knop die bij een stoplicht hoort. |
|  |  | AUTO\_NADERT\_KNOP2 | Index die hoort bij de tweede knop die bij een stoplicht hoort. |
|  |  | VOETGANGERS\_KNOP | Index die hoort bij de voetgangers knop. |
|  |  | TEST\_KNOP | Index die hoort bij de testknop. |
|  |  | DEBOUNCE\_TIJD = 10 | Het aantal milliseconde dat een knop ingedrukt moet zijn om geregistreerd te worden als press. |
|  | Unsigned long | previousMillisButtons[ AANTAL\_BUTTONS] | De laatste tijd dat de knoppen zijn gecontroleerd. |
|  | Bool | buttonPressed(byte buttonNumber) | Retourneert of de betreffende knop is ingedrukt (of ingedrukt is geweest maar nog niet afgehandeld is). |
|  |  | buttonDown(byte buttonNumber) | Retourneert of de betreffende knop onder de analoge grens van een buttonpress is. |
|  | ButtonStates | buttonsDown[AANTAL\_ BUTTONS] | Houdt de states van de verschillende knoppen bij. |
|  | Void | checkButtons() | Update de array van buttons presses. |
|  |  | buttonSetup() | Zet de pin modes van de buttons |
| Buzzer | Const byte | BUZZER = 13 | Het pinnummer van de buzzer. |
|  |  | UIT\_FREQUENTIE = 0 | Frequentie om de buzzer uit te zetten. |
|  | Const int | LAGE\_FREQUENTIE = 2 | Laagste van de gebruikte frequenties. |
|  |  | HOGE\_FREQUENTIE = 30 | Hoogste van de gebruikte frequenties. |
|  |  | DRIE\_TIKKEN\_INTERVAL = 500 | Tijd tussen elke drie tikken waar dat nodig is. Tevens de tijd waarin die drie tikken plaats vinden. |
|  |  | DRIE\_TIKKEN\_FREQUENTIE | Uitgerekende frequentie om drie tikken te laten horen binnen de DRIE\_TIKKEN\_INTERVAL |
|  | Unsigned long | previousMillisBuzzer | Tijd sinds de laatste keer dat de buzzer naar hoog is gezet. |
|  |  | timerBuzzerVoetgangers Overgang Open | Timer die bijhoudt of de buzzer weer uit moet na het openen van de voetgangersoversteekplaats. |
|  |  | buzzerTikkenTimer | Timer die bijhoudt of de tijd van de drie tikken is verlopen. |
|  | Int | buzzerFrequentie | Huidige frequentie van de buzzer. |
|  | Const int | TIJD\_BUZZER\_AAN\_ OPENEN\_VOETGANGERS | Tijd dat de buzzer een geluid maakt bij het openen van de voetgangersoversteekplaats. |
|  | TikkenStates | buzzerState | State die wordt gebruikt om te bepalen in welk deel van de drie tikken de buzzer is. |
|  | Void | buzzerZetFrequentie(byte frequentie) | Zet een nieuwe frequentie van de buzzer. |
|  |  | buzzerTikken\_ statemachineSetup() | Initialiseert de statemachine voor de drie tikken. |
|  |  | buzzerTikken\_run Statemachine() | Voert de statemachine van de drie tikken uit |
|  |  | buzzerDrieTikken\_Entry() | Reset de buzzerTikkenTimer en veranderd de frequentie van de buzzer naar de DRIE\_TIKKEN\_FREQUENTIE. |
|  |  | buzzerDrieTikkenPauze\_ Entry() | Reset de buzzerTikkenTimer en zet de buzzer uit. |
| DagNacht | Const byte | LDR = A4 | Pin waarop de LDR-spanningsdeler is aangesloten. |
|  | Const int | LICHTGRENS\_NAAR\_DAG | Waarde die LDR minimaal moet hebben om naar de dag stand te gaan. |
|  |  | LICHTGRENS\_NAAR\_NACHT | Waarde die de LDR maximaal mag hebben om in de nachtstand te komen. |
|  | Bool | Dag | Houdt bij of het op dit moment dag is. |
|  |  | isDag() | Update dag, en retourneert dit. |
|  | Void | dagNachtSetup() | Zet de pin mode van de LDR. |
|  |  |  | *Bevat ook Entry, Do en Exit functies van naarNachtstand, nachtStand en naar dagStand* |
| Display | Const byte | Symbolen[] | Bevat de bit patronen van de symbolen die weergegeven kunnen worden op het 7-segment display. |
|  | Void | toonSymbool(byte symboolNummer) | Functie die het symbool op het betreffende index van SYMBOLEN naar het display schuift. |
|  |  | display\_ToonWachten() | Wrapper van toonSymbool om het symbool wachten te laten zien. |
|  |  | display\_ToonLetExtraOp () | Wrapper van toonSymbool om het symbool LetExtraOp te laten zien. |
|  |  | display\_ToonOversteken() | Wrapper van toonSymbool om het symbool Oversteken te laten zien. |
| Events |  |  | Bevat Entry, Do en Exit functies die niet onder te brengen zijn tot hardware componenten. |
|  | Unsigned long | timerKruispuntLeeg | Timer die gebruikt wordt om te kijken of de dode tijd al verlopen is. |
|  | Const int | DODE\_TIJD = 3000 | Tijd die het kruispunt leeg moet zijn, voordat er een nieuwe richting open kan. |
| Register | Const byte | wordSize = 8 | Het aantal bits dat er in het register past. |
|  |  | dataSerialPin = 12 | DS pin van het register. |
|  |  | shiftClockPulsePin = 10 | SH\_CP pin van het register. |
|  |  | latchPin = 11 | ST\_CP pin van het register |
|  | Void | registerSetup() | Zet de pinmodes van het register. |
|  |  | writeWord(int data) | Schrijft de bits van data naar het register. |
|  |  | Latch() | Latched het register (schuift de bits door naar de output). |
|  |  | feedBit(bool state) | Schuift 1 bit het register in. |
| Serial | char | laatstOntvangenKarakter | Houdt het laatst over de seriële bus verstuurde karakter bij. |
|  |  | getLaatstOntvangenKarater() | Retourneert het laatst ontvangen karakter als hoofdletter. |
| Slagboom | Servo | slagboom | Instantie van de Servo klasse die bij de slagboom hoort. |
|  | Const byte | MAX\_HOEK = 90 | Maximale hoek waar de slagboom naartoe mag draaien. |
|  |  | MIN\_HOEK = 0 | Minimale hoek waarde de slagboom heen mag draaien. |
|  |  | GRANDEN\_PER\_SECONDE = 15 | De snelheid waarop de slagboom draait. |
|  | Unsigned long | slagboomPreviousMillis | Wordt gebruikt om de snelheid van de slagboom te handhaven. |
|  | Void | slagboomSetup() | Maakt de slagboom klaar voor gebruikt door de pin te attachen en de slagboom naar de minimale hoek te draaien. |
|  |  | slagboomOmhoog() | Draait de slagboom een beetje omhoog als de timer dit toelaat. Roep deze functie herhaaldelijk aan om de slagboom geheel te openen. |
|  |  | slagboomOmlaag() | Draait de slagboom een beetje omlaag als de timer dit toelaat. Roep deze functie herhaaldelijk aan om de slagboom geheel te sluiten. |
|  | Bool | isSlagboomOpen() | Retourneert of de slagboom geheel geopend is. |
|  |  | isSlagboomDicht() | Retourneert of de slagboom geheel gesloten is. |
|  |  |  | *Bevat ook Entry, Do en Exit functies van voetgangersOvergangOpen, openenVoetgangersOvergang, aftellenVoetgangersOvergang en afsluitenVoetgangersOvergang.* |
|  | States | mainState | Houdt de hoofdstatus van het programma bij. |
|  | Void | stateMachineSetup() | Initialiseert de statemachine. |
|  |  | runStatemachine() | Voert de statemachine uit. |
| Stoplicht | Unsigned long | timerOvergangNaarDoor Rijden | Timer voor de tijd tussen het doorrijden en het stoppen in (bij het Duitse stoplicht is dit de rood + geel toestand). |
|  |  | timerDoorRijden | Timer voor de tijd dat het stoplicht op doorrijden blijft staan. |
|  |  | previousMillisKnipper | Timer voor het knipperen van een lamp. |
|  |  | timerOvergangNaarStoppen | Timer voor de tijd dat het stoplicht op de overgang naar het stoppen blijft staan (bij Nederlands stoplicht is dit de gele toestand). |
|  |  | getTimerDoorRijden() | Getter voor timerDoorRijden, benodigd voor de statemachine. |
|  |  | getTimerOvergangNaar Stoppen () | Getter voor timerOvergangNaarStoppen, benodigd voor de statemachine. |
|  |  | getTimerOvergangNaar DoorRijden() | Getter voor timerOvergangNaarDoorRijden, benodigd voor de statemachine. |
|  | Const int | OVERGANG\_NAAR\_RIJ\_ DOOR\_TIJD = 800 | Tijd dat het stoplicht op de overgang van stoppen naar doorrijden blijft staan. |
|  |  | TIJD\_OP\_DOOR\_RIJDEN = 5000 | Tijd dat het stoplicht op doorrijden blijft staan. |
|  |  | KNIPPER\_INTERVAL = 300 | Tijd die tussen het schakelen van een lamp zit in de knipper stand. |
|  |  | KNIPPER\_TIJD\_OOSTENRIJK = 2000 | Tijd dat de groene lamp knippert bij het Oostenrijkse stoplicht. |
|  | Int | getOVERGANG\_NAAR\_RIJ\_ DOOR \_TIJD() | Getter voor de overgangstijd naar doorrijden. Benodigd voor de statemachine. |
|  |  | getTIJD\_OP\_DOOR\_RIJDEN() | Getter voor de tijd dat er door gereden mag worden. Benodigd voor de statemachine. |
|  |  | getTUSSEN\_TIJD\_NAAR\_ STOPPEN() | Getter voor de overgangstijd naar het stoppen. Benodigd voor de statemachine. |
|  | Const byte | AANTAL\_STOPLICHTEN = 2 | Het aantal aangesloten stoplichten |
|  |  | AANTAL\_LICHTEN\_PER\_ STOPLICHT = 3 | Het aantal lichten dat elk stoplicht heeft. |
|  |  | AANTAL\_STOPLICHT\_ FASEN = 4 | Het aantal fasen dat een stoplicht kent dat vastgelegd moet worden in de patronen array. |
|  |  | AANTAL\_SOORTEN\_ STOPLICHTEN = 2 | Het aantal stoplichten waarvan de fasen beschreven moeten worden. |
|  |  | NEDERLANDS\_STOPLICHT = 0 | Semantische variabele die de bij de index van een Nederlands stoplicht hoort. |
|  |  | DUITS\_STOPLICHT = 1 | Semantische variabele die de bij de index van een Duits stoplicht hoort. |
|  | Byte | stoplichtSoort | Houdt bij welk soort stoplicht er op dit moment gebruikt wordt. Wordt pas gewisseld als de huidige stoplicht actie voorbij is. |
|  |  | stoplichtDoor | Onthoudt welk stoplicht er op dit moment niet op stoppen staat. |
|  |  | getAANTAL\_STOPLICHTEN() | Getter van het aantal stoplichten. Nodig om over de stoplichten te kunnen itereren in DagNacht. |
|  | Const byte | stoplichten[AANTAL\_ STOPLICHTEN] [AANTAL\_LICHTEN\_PER\_ STOPLICHT] | Mapping van de pinnen die bij de verschillende stoplichten horen. |
|  |  | STOPLICHT\_STOPPEN = 0 | Index die hoort bij het licht dat stoppen betekend in de stoplichtPatronen array. |
|  |  | STOPLICHT\_OVERGANG\_ NAAR\_ DOORRIJDEN = 1 | Index die hoort bij het licht dat de overgang naar doorrijden bevat in de stoplichtPatronen array. |
|  |  | STOPLICHT\_DOORRIJDEN = 2, | Index die hoort bij het licht dat doorrijden betekend in de stoplichtPatronen array. |
|  |  | STOPLICHT\_OVERGANG\_ NAAR\_ STOPPEN = 3; | Index die hoort bij het licht dat de overgang naar stoppen bevat in de stoplichtPatronen array. |
|  | Const bool | stoplichtPatronen[AANTAL\_ SOORTEN\_STOPLICHTEN] [AANTAL\_STOPLICHT\_FASEN] [AANTAL\_LICHTEN\_PER\_ STOPLICHT] | Weergeeft welke lampen er bij welke fase van een stoplicht aan moeten staan. |
|  | Void | stoplichtSetup() | Zet de pinmodes van de stoplichte. Zorgt ervoor dat het standaard licht Nederlands is, zorgt er tevens voor dat stoplichte op de stopstand beginnen. |
|  |  | stoplichtDisplayPatroon(byte stoplichtNummer, byte patroonRegelNummer) | Zet de lampen aan die bij dit stoplicht hoort volgens het geselecteerde patroon en patroonRegelNummer. |
|  |  | stoplichtZetLamp(byte stoplichtNummer, byte lampNummer, bool state) | Zet de gespecificeerde lamp naar de gespecificeerde stand. |
|  |  | stoplichtUit(byte stoplichtNummer) | Zet alle lampen van het gespecificeerde stoplicht uit. |
|  |  | knipperSetup() | Maakt de timer en knipperOn klaar om een lamp te laten knipperen. |
|  |  | Knipper(byte stoplicht Nummer, byte lampNummer) | Kijkt of de betreffende lamp uit of aan moet worde gezet om te knipperen volgens de KNIPPER\_INTERVAL. |
|  | Bool | heeftStoplichtFase(byte faseNummer) | Retourneert of er een lamp in deze fase voor het geselecteerde soort stoplicht aan moet. |
|  |  | knipperOn | Wordt gebruikt in knipper functie om te kijken in welke status de lamp is die moet knipperen. Er wordt ervan uit gegaan dat lampen die knipperen gelijktijdig knipperen. |
|  |  |  | *Bevat ook de Entry, Do en Exit functies van stoplichtOvergangNaarDoorRijden, stoplichtDoorRijden, stoplichtDoorRijdenKnipper en stoplichtOvergangNaarStoppen* |
| Test | Bool | aanHetTesten | Houdt bij of er getest wordt. |
|  |  | testCompleted() | Retourneert of de test klaar is. |
|  |  | faseKlaar | Onthoudt of de huidige testfase is afgerond. |
|  |  | slagboomOmhoogGeweest | Onthoudt of de slagboom omhoog is geweest, zodat deze weer naar beneden kan. |
|  | Unsigned long | previousMillisTest | Wordt gebruikt om te kijken of een deel van de test is voltooid. |
|  | Const int | TIJD\_LED\_AAN\_TIJDENS \_TEST | Tijd dat een lampje aan blijft staan tijdens de test. |
|  |  | TIJD\_TUSSEN\_LED\_ TEST | Tijd dat er tussen elk lampje zit. |
|  |  | TIJD\_BUZZER\_TEST | Hoelang de buzzer wordt getest. |
|  | Const byte | MAXIMALE\_TEST\_UITSLAG = 15 | Hoever de slagboom uitslaat tijden het testen. |
|  | Byte | stoplichtStandDieGetest Wordt | Welk licht er van het stoplicht wordt getest op dit moment. |
|  | TestFase | Houdt bij welk deel van de test bezig is. |  |
|  | Void | voerTestUit() | Voert een gedeelte van de test uit afhankelijk van zijn timer en state. Roep herhaaldelijk aan om de test volledig uit te voeren. |
|  |  | testSlagboom() | Voert de test va de slagboom uit. Roep herhaaldelijk aan om de slagboom volledig te testen. |
|  |  | testLicht(byte stoplichtNummer) | Test de lichten die bij het meegegeven stoplichtnummer horen. |
|  |  | testBuzzer() | Test de buzzer. |
|  |  | testSetup() | Initialiseert de test substatemachine. |
|  |  |  | *Bevat ook Entry, Do en Exit functies van test.* |

Tabel 3

## Tabbladendiagram

Hier vindt u nog een meer visuele versie van de tabbladentabel die u hierboven heeft kunnen lezen. Dit diagram bevat ook de Entry, Do en Exit functies. Ook dit diagram vindt u in de bijlagen.



Figuur 6

# Figuren

[Figuur 1 3](#_Toc510606260)

[Figuur 2 4](#_Toc510606261)

[Figuur 3 4](#_Toc510606262)

[Tabel 1 8](#_Toc510606263)

[Tabel 2 9](#_Toc510606264)

[Figuur 4 10](#_Toc510606265)

[Figuur 5 11](#_Toc510606266)

[Figuur 6 12](#_Toc510606267)

[Tabel 3 18](#_Toc510606268)

[Figuur 6 18](#_Toc510606269)