|  |
| --- |
| HAN 17/18 |
| ESD Beroepsproduct |
| Technisch Ontwerp |

|  |
| --- |
| Naam: Joris Huinink  Studentnummer: 597240  Klas: I1DA  Docent: M. Verheij  Datum: 4-4-2018 |

Inhoudsopgave

[Inleiding 3](#_Toc510618726)

[State-diagram 4](#_Toc510618727)

[Tabbladendiagram 5](#_Toc510618728)

[Toelichting van gebruikte functies en beschrijving parameters en returnwaarden 6](#_Toc510618729)

[Hardware ontwerp 7](#_Toc510618730)

[Conclusie 8](#_Toc510618731)

# Inleiding

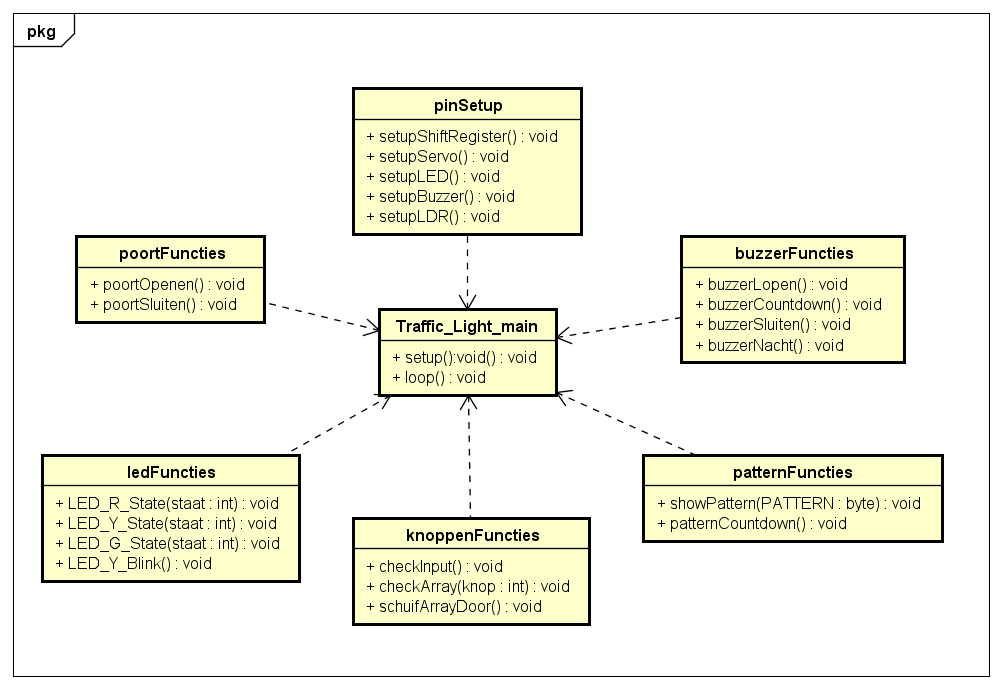
Dit document bevat het technische ontwerp van het te realiseren product. In dit document zijn meerdere diagrammen en ontwerpen opgenomen ,die verdeelt zijn over meerdere hoofdstukken, om de complete werking van het product zo goed mogelijk te tonen.

In het volgende hoofdstuk (“State-diagram”) zijn alle staten ,inclusief entry / do / exit, opgenomen en de bijbehorende overgangsvoorwaarden.

# State-diagram

# Tabbladendiagram

In het hieronder getoonde diagram worden de relaties getoond tussen de verschillende tabbladen.

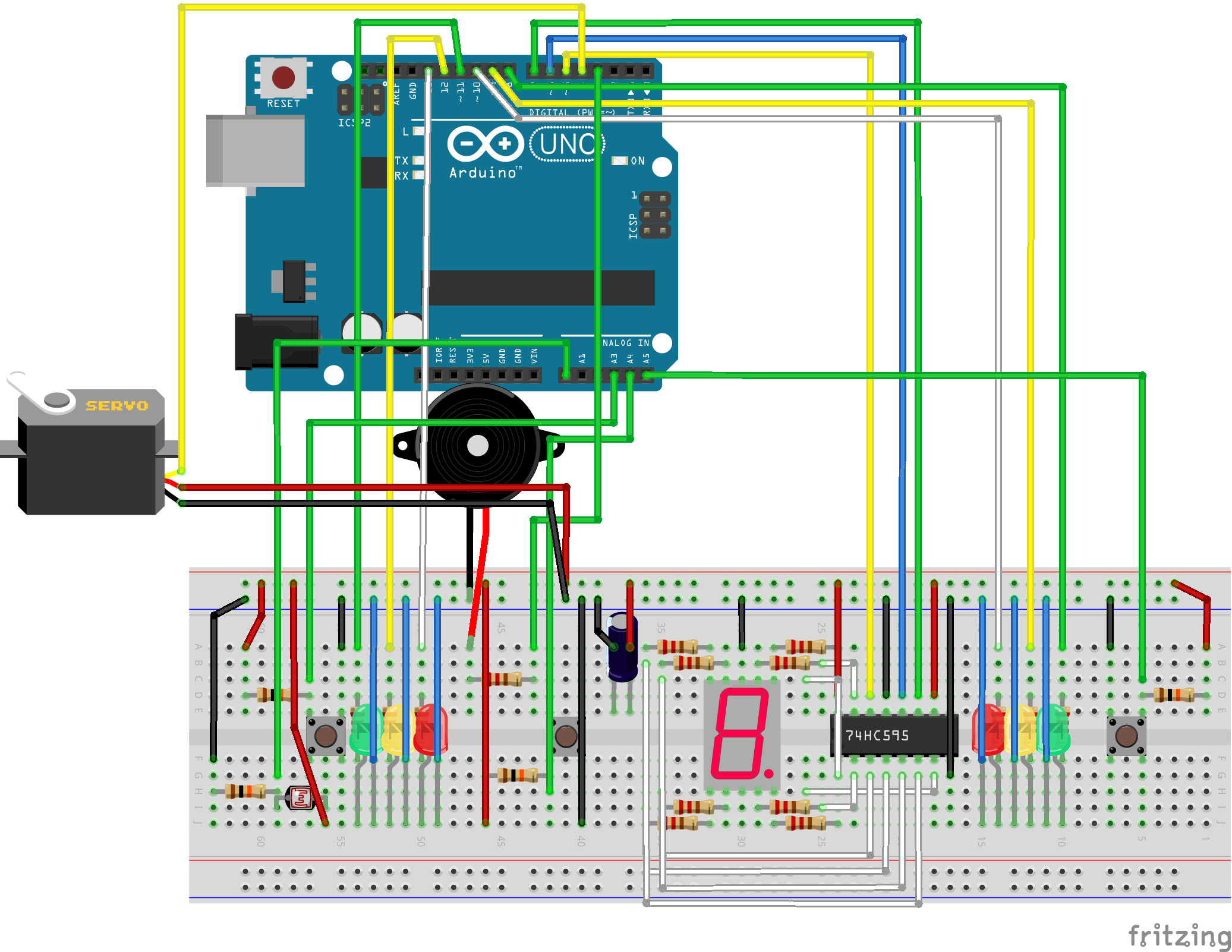
Alle methoden gedefinieerd buiten “Traffic\_Light\_main” worden aangeroepen in de setup() of in de loop() van het tabblad “Traffic\_Light\_main”.

In het volgende hoofdstuk (“Toelichting van gebruikte functies en beschrijving parameters en returnwaarden”) worden de gemaakt functies uitgelegd.

# Toelichting van gebruikte functies en beschrijving parameters en returnwaarden

|  |  |  |
| --- | --- | --- |
| Tabblad | Functies + Parameters | Toelichting |
| Traffic\_Light\_  main | void setup() | Hierin worden alle pinnen gezet en start toestand gezet |
|  | void loop() | Hierin worden alle functies gebruikt die constant moeten worden aangeroepen. |
| pinSetup | void setupShiftRegister() | Zet de pinnen voor het shiftregister. |
|  | void setupServo | Zet de pinnen voor de servo. |
|  | void setupLED | Zet de pinnen voor de leds . |
|  | void setupBuzzer() | Zet de pin voor de buzzer. |
|  | void setupLDR() | Zet de pin voor de LDR. |
| buzzerFuncties | void buzzerLopen() | Laat de buzzer horen voor wanneer voetgangers mogen lopen. |
|  | void buzzerCountdown() | Laat de buzzer horen voor wanneer de timer aan het aftellen is. |
|  | void buzzerSluiten() | Laat de buzzer horen voor wanneer de poort gaat sluiten. |
|  | void buzzerNacht() | Laat de buzzer horen tijdens de nacht. |
| patternFuncties | void showPattern  (byte PATTERN) | Laat een patroon zien op basis van een meegegeven byte. |
|  | void patternCountdown() | Laat een timer van 6 naar 1 zien op het display. |
| knoppenFuncties | void checkInput() | Checkt of er een waarde is binnengekomen van één van de knoppen. |
|  | void checkArray() | Checkt of de binnengekomen waarde al in de array staat. |
|  | void schuifArrayDoor() | Schuift de positie van iedere waarde in de array met 1 positie door richting de [0]. |
| ledFuncties | void LED\_R\_State(int staat) | Zet de waarde van het rode licht. |
|  | void LED\_Y\_State(int staat) | Zet de waarde van het gele/oranje licht. |
|  | void LED\_G\_State(int staat) | Zet de waarde van het groene licht. |
|  | void LED\_Y\_Blink() | Zorgt voor knipperen van het gele/oranje licht. |
| poortFuncties | void poortOpenen() | Zet de poort op de maximale positie in een bepaalde tijd. |
|  | void poortSluiten() | Zet de poort op de minimale positie in een bepaalde tijd. |

# Hardware ontwerp

Dit ontwerp is gemaakt met het programma “Fritzing”.

Alle draden en componenten in dit diagram zijn ook zo aangesloten op het printbord en op de Arduino.

De gebruikte componenten in dit diagram staan gelijk aan de gebruikte componenten in de realisering. Ook is hierbij gelet op de verschillende waarden van de weerstanden.

# Conclusie

Aan de hand van bovenstaande diagrammen, ontwerpen en tabellen kan ik alles samenvoegen en realiseren tot een volwaardig product. Nu kan de code worden geïmplementeerd en de verschillende componenten worden aangesloten tot één werkend geheel.