



Rapport project 2.1

FUNTIONEEL ONTWERP, TECHNISCH ONTWERP

ARJAN BAKEMA, ANTON BONDER, RUBEN EEKHOF, TIM BUNK



**Hanzehogeschool
Groningen**

University of Applied Sciences

Rapport project 2.1

Namen:

Tim Bunk - 409678
Ruben Eekhof - 405033
Anton Bonder - 409075
Arjan Bakema – 384728

Leerjaar:

2020 – 2021

Vakdocent:

Arne Padmos

Opleiding:

HBO-ICT Voltijd jaar 2

Instituut:

Hanze Hogeschool Groningen

Bron afbeelding omslag:

[Afbeelding](#)

Inhoud

Hoofdstuk 1	Inleiding.....	3
Hoofdstuk 2	Functioneel ontwerp.....	4
2.1	Doelgroep.....	4
2.2	Eisen en wensen (Moscow analyse)	4
2.2.1	Must have.....	4
2.2.2	Should have	5
2.2.3	Could have.....	5
2.2.4	Won't have	5
2.3	Interface.....	6
2.3.1	Drop down menu's	6
2.3.2	Functionaliteiten.....	7
2.4	Block diagram.....	8
2.5	Communicatieprotocol	8
2.5.1	Instellingen versturen besturingseenheid naar centrale	8
2.5.2	Live	9
2.5.3	Data versturen van centrale naar besturingseenheid	10
Hoofdstuk 3	Technisch ontwerp.....	11
3.1	Aansluitingen Arduino Uno.....	11
3.1.1	I/O Config	11
3.1.2	Aansluitschema.....	11
3.2	Bestanden	11
3.2.1	Bestand hiërarchie	11
3.3	Klassendiagram	12
3.3.1	Centrale (Python).....	12
3.3.2	Besturingseenheid (C).....	14
3.4	Flowcharts.....	16

Hoofdstuk 1 Inleiding

Wij hebben de opdracht gekregen van Zeng Ltd om een systeem te bouwen voor een automatisch zonnescherf. Dit systeem bestaat uit een besturingseenheid en een centrale (Python applicatie) waarin de gebruiker data kan inzien van zijn zonnescherf en eventueel instellingen kan aanpassen. De besturingseenheid bevat een temperatuursensor, afstandsmeter en een lichtsensor. Zeng Ltd wil dit product komend jaar op de markt brengen. In dit rapport vindt u een functioneel ontwerp en een technisch ontwerp van de uitvoering van bovenstaande.

Hoofdstuk 2 Functioneel ontwerp

2.1 Doelgroep

- Volwassen particulieren.
- Eigenaren van een huis met zonneschermen.

Waarom particulieren? Dit maakt hun leven makkelijker. Ze hoeven namelijk niet zelf meer de zonneschermen te besturen. Ook als hun even weg zijn en de zon begint volop te schijnen gaan automatisch het zonnenscherm naar beneden.

2.2 Eisen en wensen (Moscow analyse)

2.2.1 Must have

Dit zijn punten die de applicatie MOET hebben, zonder deze punten kan de applicatie niet goedgekeurd worden voor gebruik/innamen.

2.2.1.1 *Ultrasonorsensor*

- Meet in hoeverre een zonnenscherm is uitgerold of opgerold in centimeters. Bij afwijkende data (als de afstand van 2 naar 100 centimeter gaat) dan wordt de data niet meegerekend.
- Elke halve seconde meet de ultrasonorsensor de afstand en wordt doorgegeven aan de besturingseenheid.

2.2.1.2 *Temperatuursensor:*

- Deze module meet de buitentemperatuur (in °C) met een maximum van 40 en een minimum van -10.
- Om de seconde wordt de buitentemperatuur opgemeten en vervolgens wordt er elke 60 seconden een gemiddelde temperatuur weergegeven.
- De gemiddelde temperatuur wordt om de 60 seconden aan de centrale doorgegeven als deze is aangesloten.

2.2.1.3 *Lichtsensoren:*

- Deze module meet de lichtintensiteit (in een range van 1 tot en met 5 doormiddel van voltage, 1 = geen licht en 5 = veel licht).
- Om de seconde wordt de lichtintensiteit opgemeten en vervolgens wordt er elke 60 seconden een gemiddelde lichtintensiteit weergegeven.
- De gemiddelde lichtintensiteit wordt om de 60 seconden aan de centrale doorgegeven als deze is aangesloten.

2.2.1.4 *Besturingseenheid*

- Kan het zonnenscherm uitrollen of binnenhalen als de temperatuur een bepaalde boven-of ondergrens overschrijdt.
- Kan het zonnenscherm uitrollen of binnenhalen als de lichtintensiteit in een bepaald gebied zit (in een range van 1 tot en met 5 doormiddel van voltage, 1 = geen licht en 5 = veel licht).
- De besturingseenheid kan data versturen en ontvangen naar en van de centrale doormiddel van een seriële verbinding.

2.2.1.5 Centrale

- Maximale uitrollen/oprollen van zonnescerm is instelbaar.
- Gebruiker kan via een computersysteem (centrale) de zonneschermen/rolluiken bedienen.
- Kan door middel van grafiekjes de data van de sensoren kan bekijken.
- De centrale kan data versturen en ontvangen naar en van de besturingseenheid doormiddel van een seriële verbinding.
- Het is mogelijk de maximale uitrolstand of oprolstand in te stellen in centimeters.
- Ook kan men dit via knoppen in de interface aangeven of het zonnescerm op-of uitgerold moeten worden.
- Als een besturingseenheid niet is aangesloten dan zal er ook geen informatie worden vertoond m.b.t. die eenheid.

2.2.2 Should have

Dit zijn punten die erin moet komen maar als ze er niet in zitten kan de applicatie nog wel in gebruik genomen worden.

- Dag/Tijdschakelaar, dit kan ingesteld worden op datum en tijd.
- Knop bij het zonnescerm op hem open/dicht te doen.

2.2.3 Could have

Dit zijn punten die er alleen in hoeven te komen als alle must's en should's erin zitten en er tijd over is.

- Website waar het dashboard op staat.
- Het personaliseren van het dashboard.
- Kunnen wisselen tussen de meeteenheden Celsius en Fahrenheit.

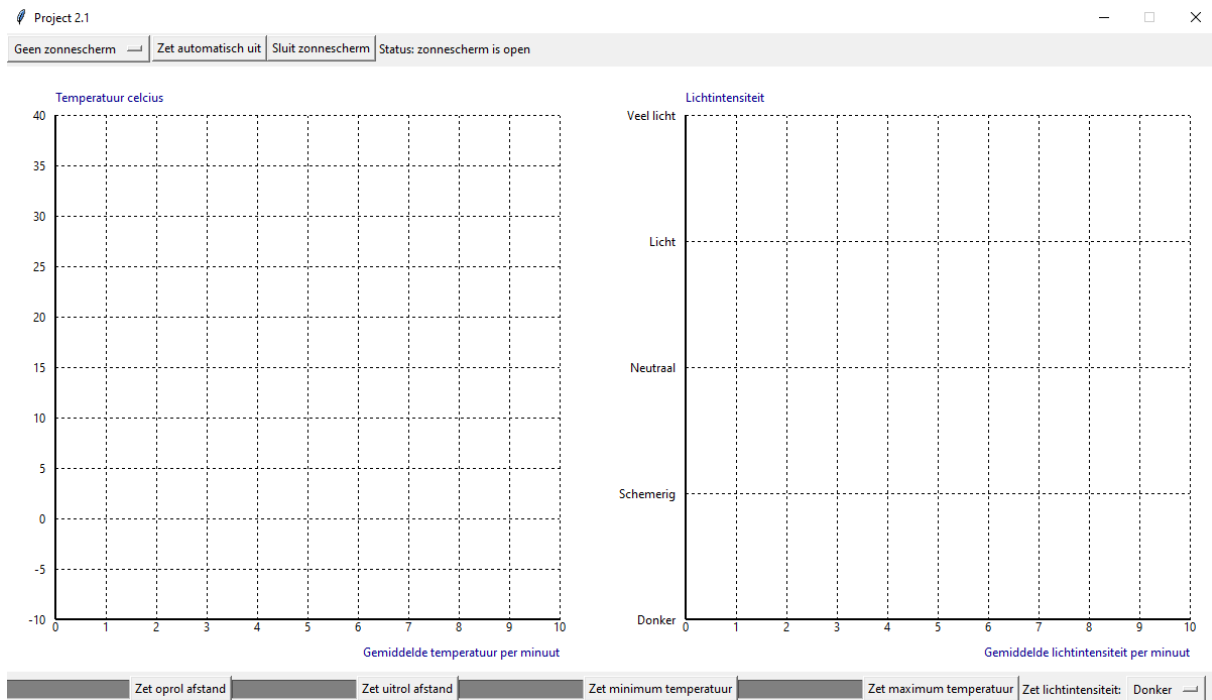
2.2.4 Won't have

Dit zijn de punten die niet voor zullen komen in deze applicatie maar, die in vervolg applicaties wel interessant kunnen zijn.

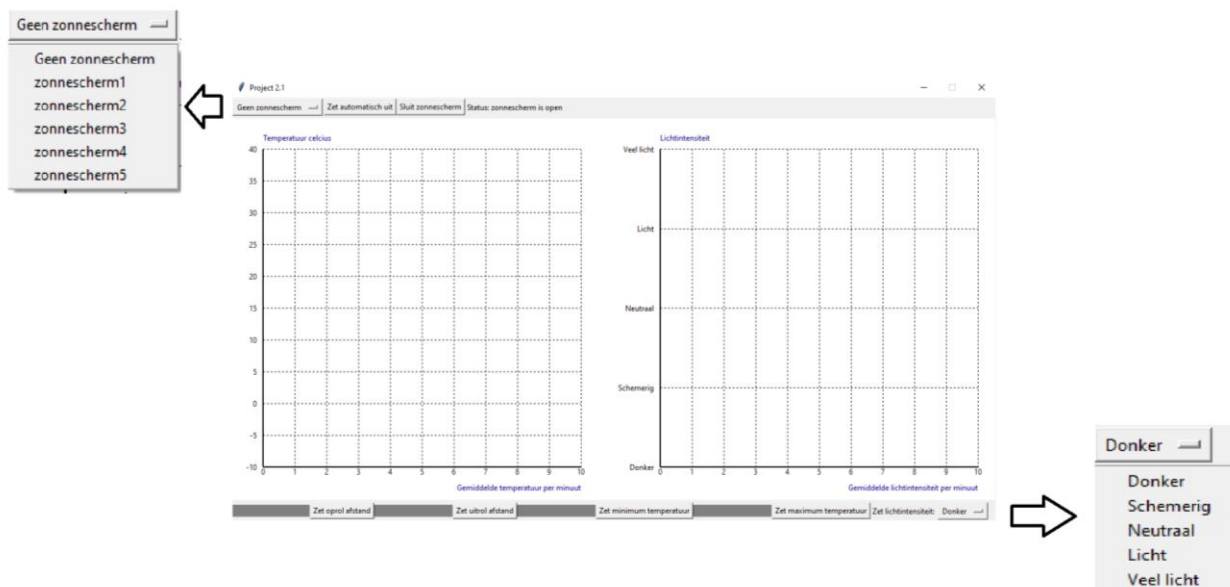
- Bediening via een mobiele applicatie.

2.3 Interface

Wij hebben gekozen voor het onderstaande model om te gebruiken voor het project. Het model is duidelijk en simpel. Wij hebben gekozen voor een duidelijk en simpel design zodat het voor de gebruiker erg toegankelijk is.

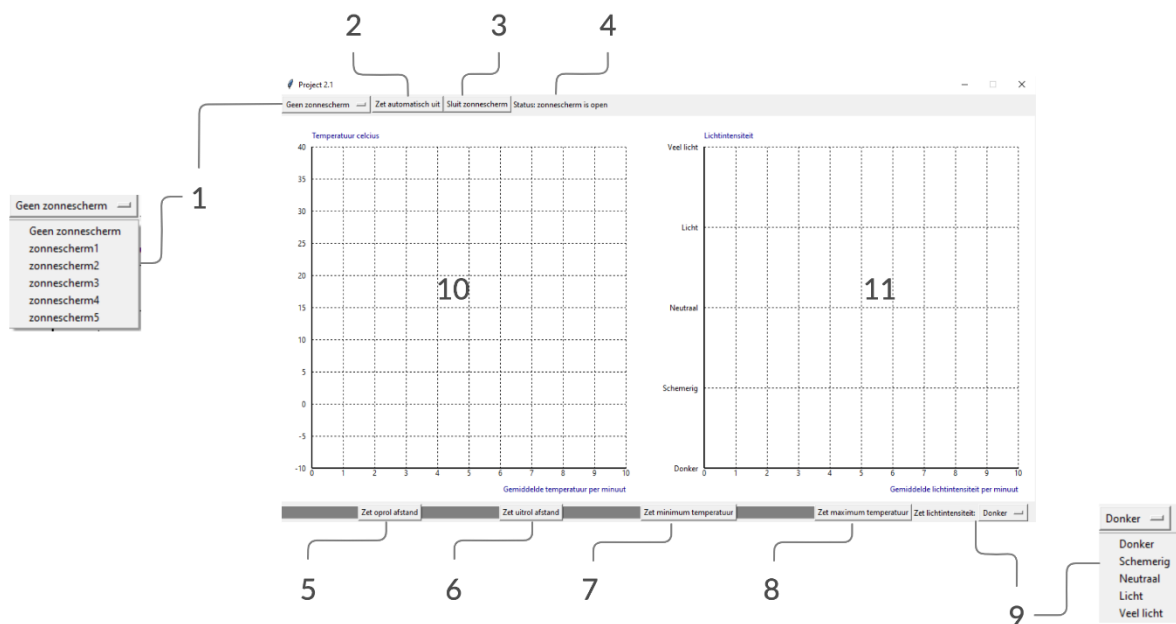


2.3.1 Drop down menu's



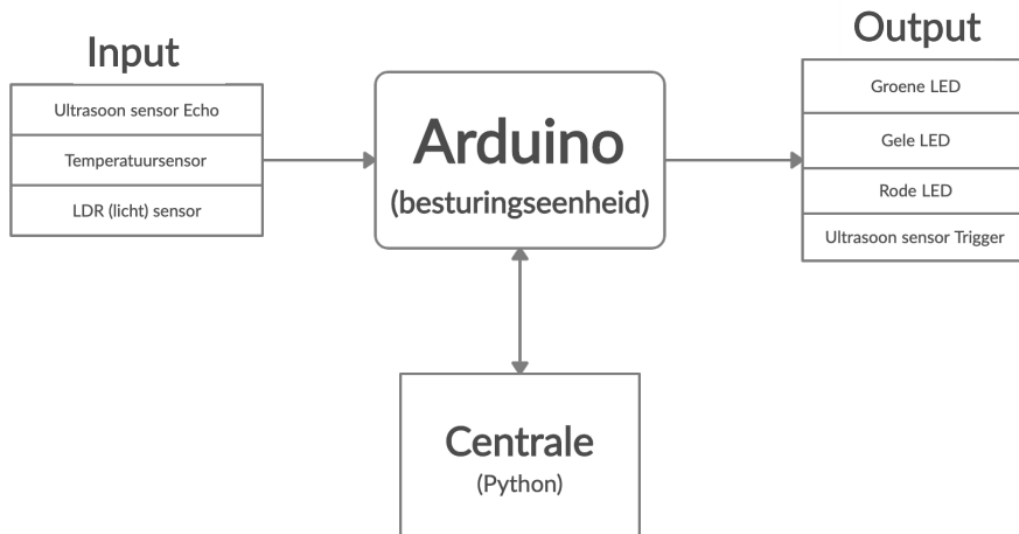
2.3.2 Functionaliteiten

Om alle functionaliteiten van ons model toe te lichten hebben we een legenda gemaakt waarin we alle functionaliteiten kort beschrijven:



- 1) **Zonnescherm selecteren:** met deze knop kan er gekozen worden tussen verschillende zonneschermen. Als er een zonnescherm wordt geselecteerd wordt het scherm geüpdatet met het desbetreffende zonnescherm.
- 2) **Zet automatisch aan/uit:** met de auto functie kan je wisselen tussen automatisch het zonnescherm naar beneden en omhoog te doen, en handmatig het zonnescherm naar beneden en omhoog doen.
- 3) **Sluit/open zonnescherm:** doormiddel van deze knop kan je het zonnescherm handmatig openen of sluiten. Als het zonnescherm open is, staat er "Sluit zonnescherm". Als het zonnescherm gesloten is, staat er "Open zonnescherm".
- 4) **Status zonnescherm:** in de statusbalk wordt weergegeven of het zonnescherm open of dicht is.
- 5) **Zet oprol afstand:** met deze knop en invoerveld kan de gebruiker een afstand (in cm) invoeren tot hoever het zonnescherm moet oprollen.
- 6) **Zet uitrol afstand:** met deze knop en invoerveld kan de gebruiker een afstand (in cm) invoeren tot hoever het zonnescherm moet uitrollen.
- 7) **Zet minimumtemperatuur:** met deze knop en invoerveld kan de gebruiker een minimale temperatuur (in graden Celsius) invoeren.
- 8) **Zet maximumtemperatuur:** met deze knop en invoerveld kan de gebruiker een maximale temperatuur (in graden Celsius) invoeren.
- 9) **Lichtintensiteit selecteren:** de gebruiker kan de lichtintensiteit selecteren waarop het zonnescherm dicht gaat. Bij een hogere lichtintensiteit dan dat er geselecteerd is, gaat het zonnescherm ook dicht.
- 10) **Grafiek temperatuur:** hier wordt een grafiek weergegeven van de gemiddelde temperatuur per minuut.
- 11) **Grafiek lichtintensiteit:** hier wordt een grafiek weergegeven van de gemiddelde lichtintensiteit per minuut.

2.4 Block diagram



2.5 Communicatieprotocol

2.5.1 Instellingen versturen van de besturingseenheid naar de centrale

Als de besturingseenheid wordt aangesloten vraagt de centrale om de huidige informatie die op de besturingseenheid staat als volgt:

Stap 1. De centrale stuurt een signaal naar de besturingseenheid om aan te geven dat we de huidige instellingen willen ophalen. Het signaal '1' wordt gestuurd om aan te geven dat we instellingen willen ontvangen.

Stap 2. Er worden pakketjes verstuurd van de besturingseenheid naar de centrale als volgt:

Pakket 1 geeft aan:

- Of de automatische modus aan of uit staat.
- Of het zonnescerm open of dicht is.
- Wat de minimaal lichtintensiteit is.

Dit wordt als volgt gedaan:

- Bit 1 = Als die 0 is dan staat automatische modus uit. Als die 1 is dan staat automatische modus aan.
- Bit 2 = Als die 0 is dan is het zonnescerm dicht. Als die 1 is dan is het zonnescerm open.
- Bit 3 tot en met 7 = Geeft aan welke minimum lichtintensiteit wordt gebruikt.

Pakket 2 geeft aan:

- De gemiddelde lichtintensiteiten van een minuut wordt verstuurd.
- Dit pakket wordt 4 keer verstuurd.

Dit wordt als volgt gedaan:

- Er worden 3 bits gebruikt om een gemiddelde lichtintensiteit te versturen.
- Een pakketje bestaat uit 8 bits en aangezien we de gemiddelde lichtintensiteiten van de laatste 10 minuten versturen is de berekening: $(3 * 10 / 8) = 4$ pakketjes.

Pakket 3 geeft aan:

- De gemiddelde temperatuur van een minuut wordt verstuurd.
- Dit pakket wordt 8 keer verstuurd.

Dit wordt als volgt gedaan:

- Er worden 6 bits gebruikt om een gemiddelde temperatuur te versturen.
- Een pakketje bestaat uit 8 bits en aangezien we de gemiddelde temperaturen van de laatste 10 minuten versturen is de berekening: $(6 * 10 / 8) = 8$ pakketjes.

Pakket 4 geeft aan:

- De maximum oprolafstand

Dit wordt als volgt gedaan:

- We gebruiken alle 8 bits om een getal te versturen die de maximum oprolafstand weergeeft.

Pakket 5 geeft aan:

- De maximum uitrolafstand

Dit wordt als volgt gedaan:

- We gebruiken alle 8 bits om een getal te versturen die de maximum uitrolafstand weergeeft.

Pakket 6 geeft aan:

- De minimumtemperatuur

Dit wordt als volgt gedaan:

- We gebruiken 6 bits om een getal te versturen van 0 tot en met 50 die de minimumtemperatuur weergeeft.

Pakket 7 geeft aan:

- De maximumtemperatuur

Dit wordt als volgt gedaan:

- We gebruiken 6 bits om een getal te versturen van 0 tot en met 50 die de maximumtemperatuur weergeeft.

2.5.2 Live

Terwijl de besturingseenheid is aangesloten wordt er ook nog data verstuurd namelijk:

- Gemiddelde temperatuur dat gemeten is over het afgelopen minuut.
- De lichtintensiteit die gemeten is over het afgelopen minuut.
- Of het zonnenscherm open of dicht is.

De besturingseenheid kan maar een van deze 3 data tegelijk opsturen en dat wordt als volgt gedaan:

- Bij een waarde van 0 tot en met 1 gaat het om de status van het zonnenscherm. 0 is dicht en 1 is open.
- Bij een waarde van 2 tot en met 7 gaat het om de gemiddelde lichtintensiteit die de afgelopen minuut is gemeten.
- Bij een waarde van 8 tot en met 57 gaat het om de gemiddelde temperatuur die de afgelopen minuut is gemeten.

2.5.3 Data versturen van de centrale naar de besturingseenheid

Een pakketje met de waarde '1' wordt gebruikt om aan te geven dat we alle instellingen willen ontvangen van deze besturingseenheid.

Om data te versturen naar een besturingseenheid moeten we 2 pakketjes versturen. De eerste geeft aan wat we opsturen en de tweede geeft de waarde aan. Dus als volgt:

Auto

Pakketje1: 2

Pakketje2: bevat 0 of 1. 0 geeft aan dat automatisch uit moet en 1 geeft aan data automatisch aan moet.

Open/sluit zonnescerm

Pakketje1: 3

Pakketje2: bevat 0 of 1. 0 geeft aan dat het zonnescerm moet opengaan en 1 geeft aan dat het zonnescerm moet sluiten.

Oprol afstand

Pakketje1: 4

Pakketje2: Bevat de maximale oprol afstand (0-255)

Uitrol afstand

Pakketje1: 5

Pakketje2: Bevat de maximale uitrol afstand (0-255)

Minimumtemperatuur

Pakketje1: 6

Pakketje2: Bevat de minimumtemperatuur (0-50)

Maximumtemperatuur

Pakketje1: 7

Pakketje2: Bevat de maximumtemperatuur (0-50)

Lichtintensiteit

Pakketje1: 8

Pakketje2: Bevat de lichtintensiteit (0-4)

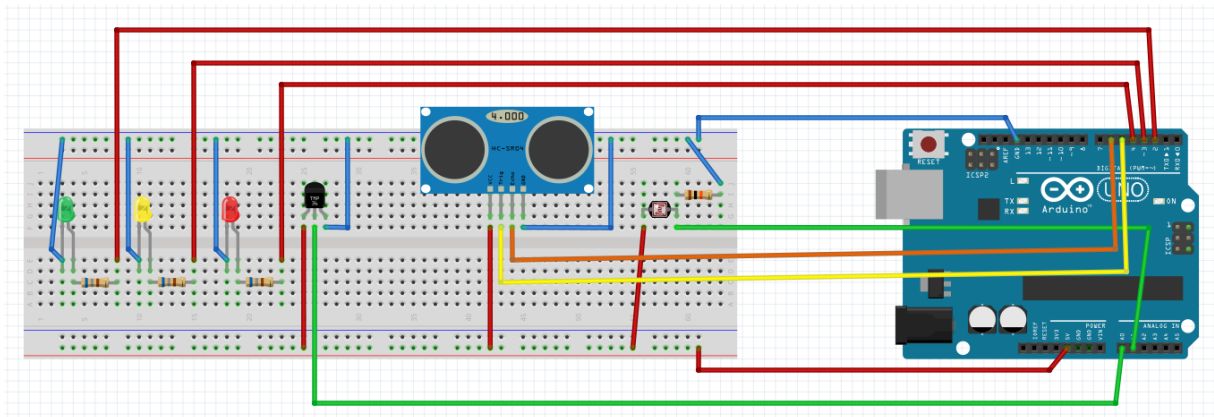
Hoofdstuk 3 Technisch ontwerp

3.1 Aansluitingen Arduino Uno

3.1.1 I/O Config

LED1 (groen)	Digitaal pin 2	Output
LED2 (geel)	Digitaal pin 3	Output
LED3 (rood)	Digitaal pin 4	Output
Ultrasoonsensor Trigger	Digitaal pin 5	Output
Ultrasoonsensor Echo	Digitaal pin 6	Input
Temperatuursensor	Analoog pin A0	Input
LDR-sensor	Analoog pin A1	Input

3.1.2 Aansluitschema



3.2 Bestanden

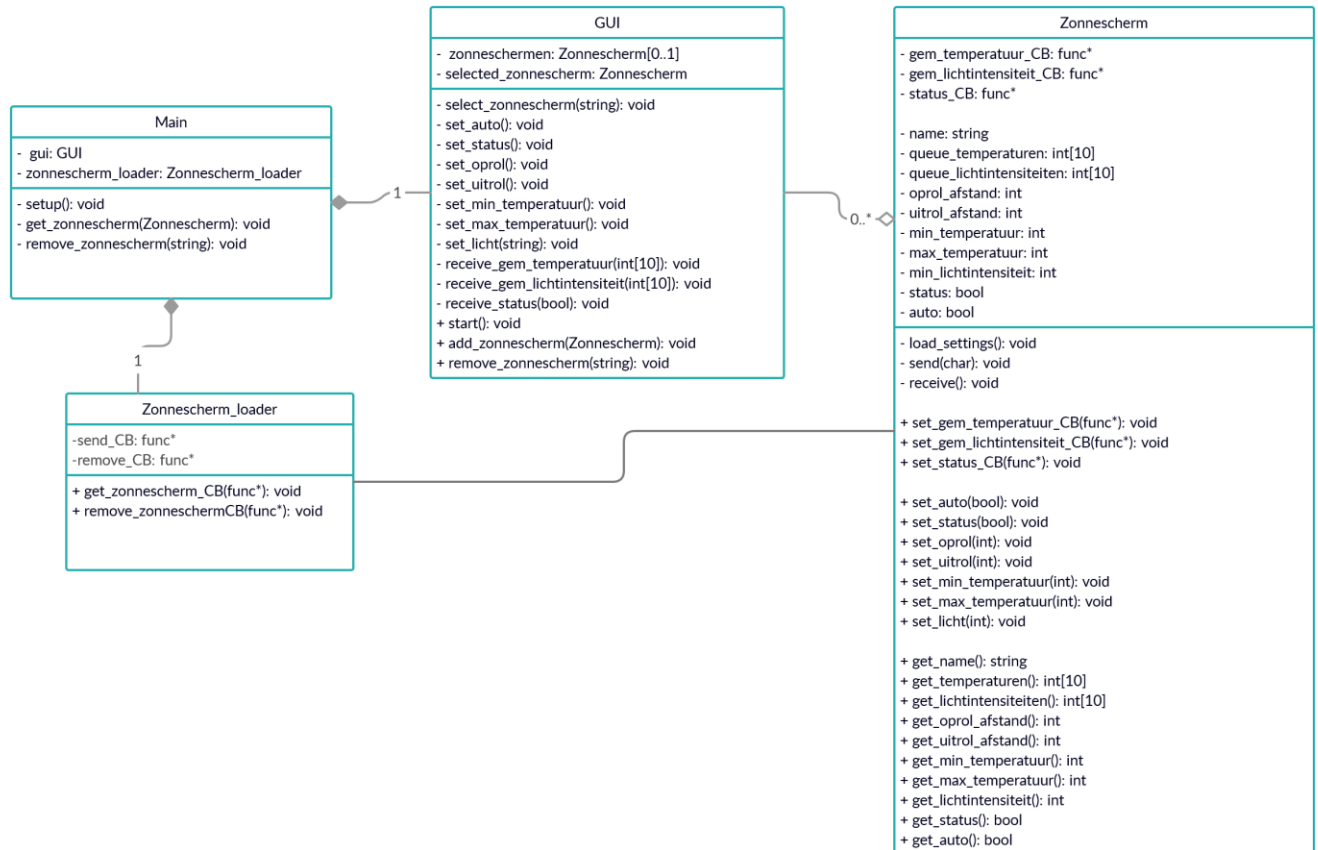
3.2.1 Bestand hiërarchie

- Rolluik/
 - o Py/
 - __main__.py
 - GUI.py
 - Zonnescherm_loader.py
 - Zonnescherm.py
 - o C/
 - AVR_TTC_scheduler.c
 - AVR_TTC_scheduler.h
 - Leds.c
 - Leds.h
 - Ultrasonic_sensor.c
 - Ultrasonic_sensor.h
 - Temperatuur_sensor.c
 - Temperatuur_sensor.h
 - LDR_sensor.c
 - LDR_sensor.h
 - Main.c

3.3 Klassendiagram

3.3.1 Centrale (Python)

Dit is de class diagram die hoort bij de Python code



Er zijn 3 classes **Zonnescherm_loader**, **GUI** en **Zonnescherm**. **Main** is dus geen class maar een python bestand waarin het geheel van de applicatie wordt gevormd.

Zonnescherm:

In de **Zonnescherm** class is alle informatie aanwezig die bij een besturingseenheid hoort. Wanneer de class wordt gemaakt wordt de functie `load_settings()` aangeroepen deze zorgt er voor dat alle data van de besturingseenheid in deze class wordt gezet.

De name van het **Zonnescherm** is de COM port waar die op aangesloten is.

Daarnaast heb je de functies `send(char)` en `receive()`. Deze zorgen ervoor dat je data kan versturen naar de besturingseenheid en data kan ontvangen.

Wanneer data wordt ontvangen wordt er gekeken naar wat de data inhoudt (zie hiervoor het communicatieprotocol). Daarna wordt de juiste CB aangeroepen dus `gem_temperatuur_CB`, `gem_lichtintensiteit_CB` of `status_CB`. Doormiddel van deze callbacks wordt de data van het zonnescherm verstuurd naar de GUI. Hiervoor moet de GUI dus wel eerst de callback functies doorgeven aan het **Zonnescherm** doormiddel van de functies `set_gem_temperatuur_CB(func*)`, `set_gem_lichtintensiteit_CB(func*)` of `set_status_CB(func*)`.

Ook heeft het zonnescerm een aantal getters en setters. Wanneer een setter wordt aangeroepen moet de gegeven data ook worden doorgegeven aan de besturingseenheid doormiddel van de `send(char)` functie.

Zonnescerm_loader

De class `Zonnescerm_loader` heeft maar twee taken namelijk kijken wanneer er een besturingseenheid wordt aangesloten of verwijderd. Wanneer er een wordt aangesloten of verwijderd geeft de `Zonnescerm_loader` dit door aan de Main doormiddel van callbacks.

Dus wanneer er een wordt aangesloten wordt er een nieuwe instantie gemaakt van `Zonnescerm` en wordt de `send_CB` aangeroepen en wordt de `Zonnescerm` meegegeven.

En wanneer er een wordt verwijderd wordt de naam (COM-poort) doorgegeven aan de `remove_CB`.

Main

De main heeft een instantie van `Zonnescerm_loader` en een instantie van GUI. Wanneer de Main een `Zonnescerm` ontvangt geeft hij deze door aan de GUI en wanneer de Main een signaal krijgt dat er een `Zonnescerm` verwijderd moet worden geeft hij dit ook door aan de GUI.

GUI

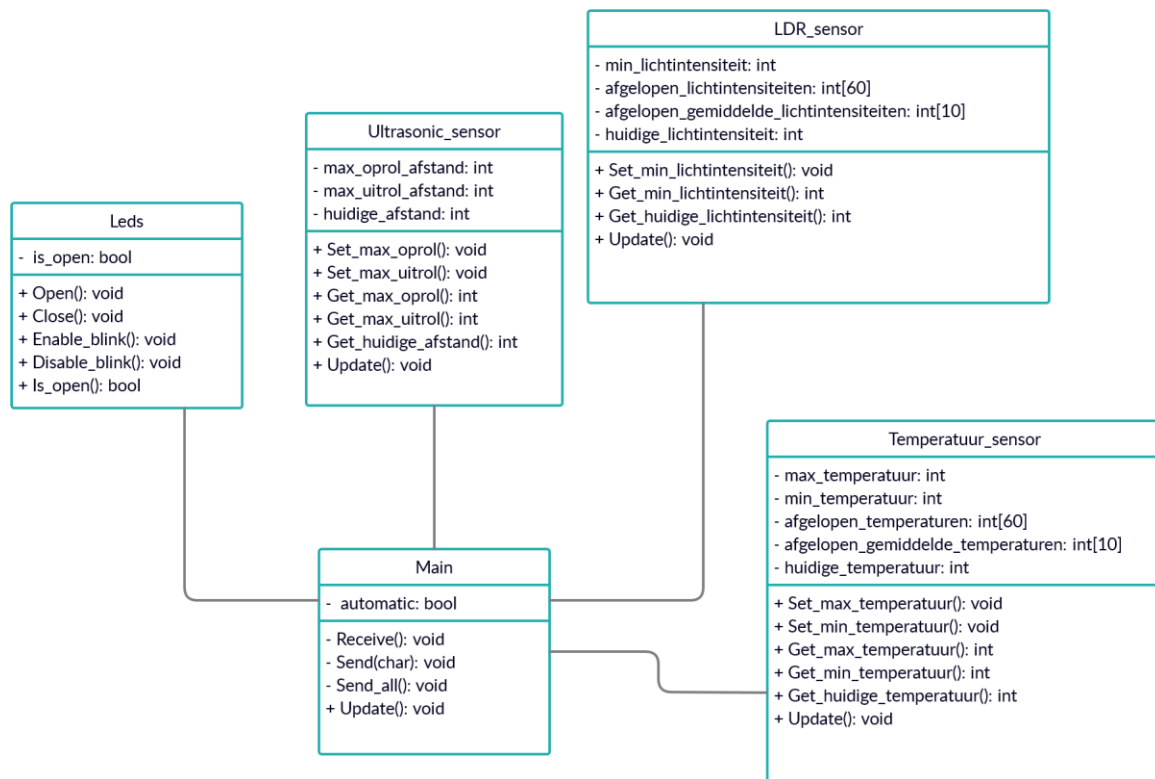
De GUI bevat een lijstje met alle `Zonneschermen` die op het moment verbonden zijn. Ook wordt er bijgehouden welke `Zonnescerm` erop het moment geselecteerd is door de gebruiker. Dit wordt aangegeven door de variabele `selected_zonnescerm` (deze kan `None` zijn als er niks is geselecteerd).

De GUI bevat allemaal knoppen en tekstvelden en zodra er ergens op een knop wordt geklikt dan wordt de juiste functie bij die knop aangeroepen. Vervolgens wordt de wijziging die plaats vindt door die knop doorgegeven aan de `selected_zonnescerm`.

Wanneer `select_zonnescerm(string)` wordt aangeroepen dan betekent dat, dat de gebruiker een ander zonnescerm wil bekijken. Dus dan moeten alle instellingen die op die zonnescerm staan worden geladen in de GUI.

3.3.2 Besturingseenheid (C)

Dit is de class diagram voor de C-code. Het zijn uiteraard geen classes maar het geeft wel goed aan hoe de code gaat werken.



Leds

De leds houden bij of het zonnescerm is geopend of gesloten. Je kan met de functies Open() en Close() aangeven of je het zonnescerm wil openen of sluiten. Daarnaast kan je ook een gele led laten knipperen om aan te geven dat hij open of dicht gaat met de functies Enable_blink() en Disable_blink().

Ultrasonic_sensor

De Ultrasoonsensor houdt de afstand bij van het zonnescerm in centimeters. Deze wordt opgeslagen in de variabele huidige_afstand. Daarnaast zijn er nog twee variabelen die de maximale oprol en uitrol afstanden bijhouden. Deze kunnen opgevraagd worden met getters of bijgewerkt worden met setters. De updatefunctie wordt elke 0.5 seconden aangeroepen om te checken wat de afstand van het zonnescerm in centimeters nu is.

LDR_sensor

De LDR_sensor houdt de min_lichtintensiteit bij deze kan worden opgehaald of bijgewerkt worden met getters en setters. Elke seconde wordt Update() aangeroepen. Deze haalt de huidige lichtintensiteit op en naar 60 seconden wordt er een gemiddelde berekend die wordt opgeslagen in afgelopen_gemiddelde_lichtintensiteiten.

Als Get_huidige_lichtintensiteit() wordt aangeroepen geeft deze de laatste gemiddelde lichtintensiteit terug die was toegevoegd.

Temperatuur_sensor

De temperatuur_sensor houdt minimale en maximale temperatuur bij en deze kunnen worden bijgewerkt met de getters en setters. Elke seconde wordt Update() aangeroepen. Deze haalt de huidige temperatuur op en na 60 seconden wordt er een gemiddelde berekend die wordt opgeslagen in afgelopen_gemiddelde_temperatures.

Als Get_huidige_temperatuur() wordt aangeroepen geeft deze de laatste gemiddelde temperatuur terug die was toegevoegd.

Main

De main heeft een belangrijk doel en dat is om de data van alle sensoren te gebruiken om te bepalen of het zonnenscherm opengedaan moet worden of niet. Dit wordt gedaan in de Updatefunctie elke 0.5 seconden.

Ook heeft de main een variabele auto waarin wordt bijgehouden of het zonnenscherm automatisch open en dicht gedaan mag worden.

De main heeft ook nog 2 belangrijke functies namelijk Send(char) en Receive() deze worden gebruikt om te communiceren met de centrale (Python software).

Als Send_all wordt aangeroepen wordt alle data die op de besturingseenheid staat verstuurt naar de centrale.

3.4 Flowcharts

