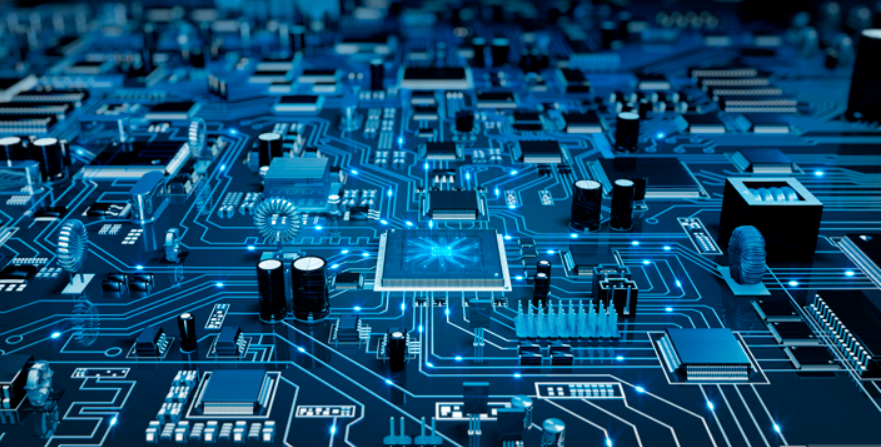


Auteurs: Stephan Hilberts, Gijs Entius & Dagmar Hoogendorp

Project Embedded Systems

Documentatie van team !C

# Projectdocumentatie



De projectdocumentatie van Team !C

Opdrachtgever: Hanzehogeschool Groningen

Klas: ITV2F

Onder begeleiding van: Ramazan Kirmali

Documentdatum: 11 november 2017

RES INHOUD

# Inleiding

Tegenwoordig is smart technologie enorm populair. Er worden steeds meer dingen, die men in het dagelijks leven gebruikt, slimmer gemaakt. Zo zijn er tegenwoordig automatische deuropeners, slimme verlichting, maar ook slimme rolluiken, die vooraf op een simpele manier door de gebruiker programmeerbaar zijn. De grootste ontwikkelaar van deze technologie is Zeng ltd.

Zeng ltd. is gevestigd in Hong Kong met drie grote fabrieken in China. Deze fabrieken hebben elk 2800 werknemers. Hiermee is Zeng ltd. de grootste ontwikkelaar van domotica-apparatuur. Dit is slimme, vaak geautomatiseerde apparatuur zoals de slimme, zelfwerkende rolluiken.

Dit rapport is geschreven naar aanleiding van een probleem dat Zeng ltd. al geruime tijd probeert op te lossen. De omzet van Zeng ltd. neemt af omdat concurrentie de producten van Zeng ltd. kopieert en verkoopt. Om de omzet te doen stijgen, wil Zeng ltd. het grootste marktaandeel hebben in Europa.

Zeng ltd. heeft een ICT-bureau genaamd Team !C in de arm genomen om onderzoek te doen naar mogelijkheden om de omzet te doen stijgen. Team !C heeft de opdracht gekregen om een innovatieve domotica-oplossing voor zonneschermen en rolluiken te realiseren voor Zeng ltd. Dit rapport doet verslag van het ontwikkelingsproces van deze oplossing. De doelstelling van de opdracht is om voor 16 november 2017, een innovatieve oplossing te hebben voor de zonneschermen en rolluiken, waardoor het marktaandeel van Zeng ltd. met 10% stijgt.

Dit rapport bestaat uit genummerde hoofdstukken en begint bij hoofdstuk 1. In hoofdstuk 1 wordt eerst het plan van aanpak met alle onderdelen besproken. Vervolgens wordt in hoofdstuk 2 het functioneel ontwerp besproken. De verschillende onderdelen en protocollen worden hierin uitvoering besproken in meerdere tabellen. Hierna wordt in hoofdstuk 3 bij het technische ontwerp stilgestaan. Tot slot wordt besproken wat het project opgeleverd heeft met daaropvolgend de bijlages voor het ontwerp.

# Hoofdstuk 1.1 – Plan van aanpak, projectinhoud

## Inleiding

In dit plan van aanpak wordt er beschreven hoe Team !C het project gaat aanpakken. Voor dit project is het de bedoeling dat rolluiken geautomatiseerd worden. Dit wordt gedaan door verschillende Arduino’s in te zetten als besturingseenheden. Deze Arduino’s worden door middel van de taal C geprogrammeerd om de temperatuur, lichtintensiteit en afstand te meten en deze door te sturen naar de centrale. Naast de besturingseenheden, wordt er ook een centrale gerealiseerd. Deze centrale ontvangt de gemeten data van de besturingseenheden en geeft deze data weer in een GUI. Daarnaast moet het mogelijk zijn om via de GUI waarden in te stellen waarop de rolluiken in- en uitrollen. De centrale zal worden geprogrammeerd met Python 3.5.

## – Doel van het plan van aanpak

Het doel van deze plan van aanpak is om een gedetailleerd plan te weergeven over de aanpak van het project genaamd “Embedded Systems”.

## – Doelgroep

Het plan van aanpak is bestemd voor:

- De Hanzehogeschool Groningen

- Stephan Hilberts, Gijs Entius & Dagmar Hoogendorp

- Directie Zeng ltd. & Projectmanager Zeng ltd.

## – Toepassingsgebied van het plan van aanpak

Dit project is de basis voor een systeem dat een zonnescherm kan besturen. Er wordt gewerkt met de programmeertalen C en Python. Daarnaast zal er een duidelijke documentatie zijn en worden de voltooide taken bijgehouden.

## – Opzet en structuur van dit plan van aanpak

Het plan van aanpak beslaat de volgende onderwerpen:

* **Projectinhoud** – Hierin wordt beschreven wat er in het plan van aanpak wordt beschreven.
* **Projectdefinitie** – Wat is de doelstelling van het project, de probleemstelling en risico’s?
* **Projectaanpak** – Wat zijn de gestelde mijlpalen en de gewenste resultaten?
* **Projectorganisatie** – Welke middelen worden er gebruikt en hoe worden de uren georganiseerd?
* **Projectplanning** – Wat zijn de mijlpalen en zijn deze behaald?

## – Algemene definities

De term “domotica” staat voor huisautomatisering. De term “agile” staat voor het iteratief aanpakken van projecten. Wanneer er gebruik gemaakt wordt van de agile werkmethode, kan het product snel aangepast worden aan nieuwe wensen.

# Hoofdstuk 1.2 – Projectdefinitie

## 1.2.1 – Projectopdracht

De opdracht voor dit project is door Zeng ltd. opgebouwd, waarbij het de bedoeling is dat er een domotica oplossing wordt gemaakt voor een zonnescherm.

## 1.2.2 – Probleemstelling

Zeng ltd. stelt de volgende eisen aan het eindproduct:

* Een centrale besturingseenheid voor de eindgebruiker met de volgende eigenschappen.
  + Kan gegevens ontvangen van individuele besturingseenheden.
  + Kan visualisaties geven van de gegevens die worden ontvangen.
  + Heeft een graphical user interface.
  + Kan de volgende opdrachten naar de besturingseenheden sturen.
    - In- uitrollen van de zonneschermen.
    - Instellingen van de besturingseenheden aanpassen.
* Individueel werkende besturingseenheden welke de volgende eigenschappen hebben.
  + Mogelijkheid om temperatuur, licht en afstand te meten met behulp van sensoren.
  + Stuurt gegevens naar de centrale wanneer deze is aangesloten.
  + Mogelijkheid om variabelen in te stellen.
  + De individueel werkende besturingseenheden moeten aangestuurd kunnen worden door de centrale.

## 1.2.3 – Doelstelling

Dit plan heeft alleen betrekking op het ontwikkelen van een prototype eindproduct. Er zal een product gemaakt voor een thuis waarin een zonnescherm automatisch in- en uitgerold kan worden, afhankelijk van de weersomstandigheden.

Wanneer dit doel bereikt is, dan is het project succesvol geweest.

## 1.2.4 – Definitie projecteinde

Het project is af wanneer de individuele besturingseenheden metingen uit kunnen voeren en deze metingen door kunnen sturen naar een centrale. Dit wordt gemeten aan de hand van tests waarbij de functionaliteit van de besturingseenheden onder de loep genomen wordt. Wanneer alle test succesvol blijken te zijn, dan wordt het project dusdanig als af bestempeld. De criteria waar het project dan aan voldoet komen overeen met het functioneel dan wel het technisch ontwerp.

## 1.2.5 – Risico’s

Bij het project moet rekening gehouden worden met de volgende activiteiten:

* Ziekte van werknemers
* Het uitvallen van apparatuur zoals laptops
* Bugs in de software

## 1.2.6 – Kwaliteitsborging

De kwaliteit van projectuitvoering en de projectresultaten wordt geborgd door de volgende zaken:

* Met de projectmedewerkers zullen wij elke week een ‘codereview’ houden. Hierin kijken we naar elkaars code en kijken we wat er verbeterd kan worden.
* We houden producten en processen nauw in de gaten. Dit betekent dat iedereen weet wat door wie gedaan wordt.
* Elke week kijken we naar de doelen en of deze behaald zijn.

# Hoofdstuk 1.3 – Projectaanpak

## 1.3.1 – Benaderingswijze

Het project wordt opgezet middels een Agile aanpak. Hiervoor wordt er eerst met het team om tafel gezeten om een plan van aanpak te maken. Wanneer het team overeenstemming heeft bereikt wat betreft de plan van aanpak, wordt er gekeken welke functies noodzakelijk zijn voor het eindproduct. Deze functies worden vanuit de gebruiker beschreven door middel van een use case diagram, in een functioneel ontwerp. Naast het functioneel ontwerp, worden de technische eisen van elk onderdeel van het project, beschreven in een technisch ontwerp. Dit wordt gedaan door middel van een UML diagram. Wanneer zowel het functioneel als het technische ontwerp af zijn, worden de taken verdeeld over de teamleden.

## 1.3.2 – Fasering en mijlpalen

Het project is opgedeeld in vijf fasen:

**Initiatiefase**

Hierin wordt er, door het team, het idee doorgenomen en wordt er gebrainstormd over eventuele aanvullingen.

**Definitiefase**

Hierin worden de eisen en wensen die aan het project gesteld worden zo goed mogelijk bepaald. Het is belangrijk dat de verwachtingen van de opdrachtgever zo goed mogelijk worden bevredigd.

**Ontwerpfase**

In de ontwerpfase worden het functioneel en het technisch ontwerp gemaakt. Het is belangrijk dat duidelijk is hoe de verschillende functies precies werken.

**Realisatiefase**

In deze fase wordt het project daadwerkelijk uitgevoerd. De verschillende delen waaruit het eindproduct bestaat, worden gebouwd en getest. Aan het einde van de realisatiefase wordt het resultaat gecontroleerd door middel van de eisen en wensen uit de definitiefase.

Aangezien er gebruik gemaakt wordt van de Agile werkmethode, wordt er met sprints gewerkt. Elke sprint duurt 2 dagen. Voor het managen van deze sprints, worden de use cases op Trello, in de backlog geplaatst. Aan het begin van elke sprint worden de use cases in de backlog verdeeld over de teamleden.

**Nazorgfase**

Hierin wordt teruggekeken op het project. Een goede manier hiervoor is om als team en individueel te reflecteren.

## 1.3.3 – Projectresultaten

Voor dit project leveren we allereerst een functioneel ontwerp en een technisch ontwerp op. In het functionele ontwerp worden de wensen en eisen voor het product vanuit de gebruiker duidelijk gemaakt. In het technische ontwerp worden de technische eisen en de “hoe” duidelijk. Op basis van deze ontwerpen leveren we drie besturingseenheden op. Dit zijn Arduino UNO’s die geprogrammeerd zijn om licht, temperatuur of afstand te meten en deze in een gegeven interval door te sturen naar de centrale.

Verder wordt er een Python GUI opgeleverd voor op de laptop. Deze Python GUI functioneert als een centrale. Hierin worden de gemeten waarden in grafieken weergegeven. De gebruiker moet limieten voor de waarden kunnen instellen waarop de rolluiken oprollen of uitrollen. Ook moet de gebruiker de mogelijkheid hebben om het rolluik te kunnen oprollen en uitrollen door middel van een knop.

# Hoofdstuk 1.4 – Projectorganisatie

## 1.4.1 – Organisatie

De navolgende personen hebben een rol in het project:

|  |  |  |
| --- | --- | --- |
| **Persoon** | **Rol** | **Taak** |
| Zeng ltd | Opdrachtgever | Neemt besluiten, bepaalt eisen |
| Dagmar Hoogendorp | Teamlid | voert projectactiviteiten uit |
| Gijs Entius | Teamlid | voert projectactiviteiten uit |
| Stephan Hilberts | Teamlid | voert projectactiviteiten uit |

## 1.4.2 – Aanschaf tooling

De volgende software moet beschikbaar worden gesteld voor de teamleden ten behoeve van het voltooien van het project:

|  |  |
| --- | --- |
| **Software** | **Bijzonderheden** |
| Git |  |
| Python | Versie 3.5 of 3.6 |
| Atmel Studio 7 | Versie 6 of 7 zijn allebei mogelijk |
| Avrdude |  |

De volgende hardware items zijn nodig in het project:

|  |  |
| --- | --- |
| **Item** | **Bijzonderheden** |
| Arduino Uno |  |
| USB kabel | Voorkeur voor 1 meter lang |
| LED & Key TM1638 module |  |
| LED rood 5mm |  |
| LED groen 5mm |  |
| LED geel 5mm |  |
| Weerstand 680 ohm 1/8W |  |
| Jumperwire M-M 15cm |  |
| Jumperwire M-F 15cm |  |
| Breadboard 400-punt |  |
| Ultrasoonsensor HC-SR04 |  |
| LDR |  |
| Weerstand 10k ohm 1/8W |  |
| Temperatuursensor analoog | bv. TMP36 |
| Pushbutton |  |

## 1.4.3 – Rapportage

Aan het einde van het project volgt een rapport met de volgende onderdelen:

* Functioneel ontwerp
* Technisch ontwerp
* Bijlagen

o verwijzing naar repository

o schermontwerpen (van functioneel ontwerp

Naast het eindrapport worden ook de volgende items aangeleverd:

* Broncode software centrale
* Broncode software besturingseenheden

Aan het einde van het project volgt een presentatie/demonstratie van de eindproducten.

## 1.4.4 – Documentatie

Bij het project worden de volgende documenten opgeleverd:

* Plan van aanpak
* Functioneel ontwerp
* Technisch ontwerp
* Persoonlijk reflectieverslag
* Link naar de gebruikte Github
* Presentatie van eindproduct

## 1.4.5 – Communicatie

Binnen het team wordt gecommuniceerd via verschillende digitale kanalen. Voor dagelijkse, praktische communicatie wordt gebruik gemaakt van een Whatsapp groep. Communicatie binnen deze groep dient beperkt te blijven tot het organiseren van meetings en werksessies. Projectinhoudelijke communicatie verloopt via Slack. Het Scrum board wordt beheerd met Trello. Tot slot zal de code en bijbehorende documentatie beheerd en gedeeld worden via Github.

# Hoofdstuk 1.5 - Projectplanning

## 1.5.1 – Mijlpalen

|  |  |
| --- | --- |
| Arduino verbonden met centrale | X |
| Data verzenden naar centrale | X |
| Instructies verzenden naar besturingseenheid | X |
| GUI front-end wordt weergegeven | X |
| GUI back-end is werkend | X |

# Hoofdstuk 2.1 – Functioneel ontwerp, De centrale

## 2.1.1 – Data display

|  |  |
| --- | --- |
| Use Case | Data display |
| Pre-condition | De computer staat aan. Er zijn één of meerdere besturingseenheden aangesloten. |
| Standard path | s1 - Check welke besturingseenheden er gekoppeld zijn en om welke besturingseenheid het gaat.  s2 - Kijken of er data verstuurd wordt door een besturingseenheid.  s3 - Wanneer er data verstuurd wordt, wordt dit weergegeven in de Centrale. |
| Post condition | Besturingseenheden hebben een ID en er komt data binnen. |
| Alternate path | a1@s1 - besturingseenheid wordt niet herkend. Centrale kent een ID toe aan de nieuwe besturingseenheid.  a2@s2 -  Er is geen data. Dit laat de besturingseenheid je weten d.m.v. een melding. |

## 2.1.2 – GUI

Onze gebruikersinterface, ook wel GUI genoemd, laat alleen waarden zien en vraagt om input. Er zit geen functionaliteit in de GUI. De functionaliteit zit erachter in de configuratie. Die handelt de gebruikers input af. De GUI toont alleen maar waar de gebruiker de input kan invoeren.

|  |  |
| --- | --- |
| Use Case | GUI |
| Pre-condition | De computer staat aan. Er hoeft nog niet een besturingseenheid gekoppeld te zijn. De grafieken zijn leeg. |
| Standard path | s1 - Laat zien welke besturingseenheden gekoppeld zijn  s2 - Lees de data display uit, alleen positieve waarden worden doorgegeven.  s3 - Gebruiker kiest welke waarden zij/ hij wil zien.  s4 - Grafiek toont de betreffende waarden.  s5 - Gebruiker kan de temperatuur instellen.  s6 - Gebruiker kan de lichtintensiteit instellen.  s7 - Gebruiker kan de maximale uitrolstand instellen.  s8 - Gebruiker kan het scherm uit of inrollen.  s9 - Gebruiker kan besturingseenheid ‘aanvinken’. Hiermee wordt de desbetreffende besturingseenheid wel of niet getoond.  s10 - Status rolluik wordt weergegeven. |
| Post condition | Configuratie reageert op de invoer van de gebruiker. |
| Alternate path | a1@s1 - Er zijn geen besturingseenheden gekoppeld, configuratie geeft een foutmelding.  a2@s2 - Er zijn negatieve waarden, configuratie geeft een foutmelding.  a3@s3 -  Er zijn nog niet genoeg waarden gemeten. configuratie geeft een foutmelding.  a4@s5 - Invoer temperatuur valt buiten de minimale en maximale waarden. configuratie geeft een foutmelding.  a5@s6 - Invoer niet correct/ volgens verwachting, configuratie geeft een foutmelding.  a6@s7 - Het scherm is al uit of ingerold op maximale of minimale stand. configuratie geeft een foutmelding. |

## 2.1.3 – Configuratie

|  |  |
| --- | --- |
| Use Case | Configuratie |
| Pre-condition | Centrale en besturingseenheden zijn ingeschakeld en hebben onderling verbinding gemaakt. |
| Standard path | s1 - Functie die reageert op gebruiker input voor de temperatuur.  s2 - Functie die reageert op de gebruiker input voor de lichtintensiteit.  s3 - Functie die reageert op de maximale uitrolstand.  s4 - Functie rol scherm uit of in.  s5 - Toon besturingseenheid.  s6 - Instellen welke data wel en niet wordt getoond. |
| Post condition | Besturingseenheid ontvangt waarden van de configuratie. |
| Alternate path | a1@s1,s2,s3 - Wanneer de ingevoerde waarde buiten de min of max waarde valt, wordt er een error gegeven.  a2@s4 - Wanneer de opdracht hetzelfde is als de huidige stand, gebeurd er niks en wordt er een melding gegeven. |

## 2.1.4 – Sensoren uitlezen en opdrachten versturen naar besturingseenheid

|  |  |
| --- | --- |
| Use Case | Uitlezen sensoren en opdracht geven aan rolluik |
| Pre-condition | Besturingseenheden zijn verbonden en doen metingen. Centrale kan opdrachten versturen naar besturingseenheden. |
| Standard path | s1 - Lichtsensor wordt uitgelezen.  s2 - Temperatuursensor wordt uitgelezen.  s3 - Ultrasoonsensor wordt uitgelezen. Deze wordt gebruikt voor de status in de GUI.  s4 - Geef opdracht om het scherm op of uit te rollen  s5 - Test of opdracht wordt uitgevoerd |
| Post condition | Waarden worden uitgelezen en gebruikt in de Centrale. |
| Alternate path | a1@a1,s2,s3 - Zijn er geen waarden die uitgelezen kunnen worden? Exception: Sensor geeft geen waarden.  a2@s4 - wordt de opdracht niet uitgevoerd? Exception: opdracht is niet goed aangekomen.  a3@s5 - wanneer opdracht niet goed is uitgevoerd, geef een melding. |

# Hoofdstuk 2.2 – De besturingseenheden

## 2.2.1 – Temperatuureenheid

|  |  |
| --- | --- |
| Use Case | Uitlezen en versturen temperatuursensor |
| Pre-condition | Besturingseenheid is ingeschakeld. |
| Standard path | s1 - Temperatuursensor leest om de 40 seconden de temperatuur uit met een waarde in Voltage. Hoe hoger het Voltage, hoe hoger de temperatuur.  s2 - Functie zet gemeten Voltage om in temperatuur in Celsius.  s3 - Functie stuurt de gemeten waarde door naar de centrale.  s4 - Scheduler regelt dat om de 60 seconden de verstuur functie aangeroepen wordt en dus de waarde verstuurd wordt. |
| Post condition | Temperatuursensor meet en verstuurd waarden in een tijdsinterval. |
| Alternate path | Voor het uitlezen en versturen van de temperatuursensor wordt er geen gebruik gemaakt van alternate paths. |

## 2.2.2 – Lichteenheid

|  |  |
| --- | --- |
| Use Case | Uitlezen en versturen lichtsensor |
| Pre-condition | Besturingseenheid is ingeschakeld. |
| Standard path | s1 - Lichtsensor leest om de 30 seconden de lichtintensiteit uit.  s2 - Functie zet de gemeten waarden om in een 10 bits getal.  s3 - Functie stuurt gemeten waarde door naar de centrale.  s4 - Scheduler regelt dat de gemeten waarde om de 60 seconden verstuurd wordt naar de centrale. |
| Post condition | Lichtsensor meet en verstuurd waarden in een tijdsinterval |
| Alternate path | Voor het uitlezen en versturen van de lichtsensor wordt er geen gebruik gemaakt van alternate paths. |

## 2.2.3 – In/uitrollen van zonnescherm

|  |  |
| --- | --- |
| Use Case | In/uitrollen van rolluik |
| Pre-condition | Besturingseenheid is ingeschakeld en verbonden met de centrale. |
| Standard path | s1 - Status van het rolluik wordt door een functie gecontroleerd.  s2 - Wanneer de knop uitrollen wordt ingedrukt, wordt de rolluik uitgerold.  s3 - Wanneer de knop inrollen wordt ingedrukt, wordt de rolluik ingerold. |
| Post condition | Status van rolluik kan worden aangepast. |
| Alternate path | a1@s1 - Deze status van het rolluik wordt in een variabele gezet.  a2@s2 - Wanneer het rolluik de status “uitgerold” heeft, krijgt de gebruiker een melding.  a3@s3 - Wanneer het rolluik de status “ingerold” heeft, krijgt de gebruiker een melding. |

## 2.2.4 – Controle status zonnescherm

|  |  |
| --- | --- |
| Use Case | Controleren wat de status is van het rolluik. |
| Pre-condition | Besturingseenheid is ingeschakeld en verbonden met de centrale. |
| Standard path | s1 - De status van het rolluik wordt bij het eerst inschakelen “ingerold”.  s2 - Wanneer het rolluik van status veranderd, wordt dit bijgehouden in een variabele die reversed. |
| Post condition | Status van het rolluik wordt bijgehouden. |
| Alternate path | Bij het controleren wat de status is van het rolluik wordt geen gebruik gemaakt van alternate paths. |

## 2.2.5 – Ultrasoonsensor

|  |  |
| --- | --- |
| Use Case | Meten en versturen ultrasoonsensor |
| Pre-condition | Besturingseenheid is ingeschakeld en verbonden met de centrale. |
| Standard path | s1 - De ultrasoonsensor meet constant de afstand van het rolluik uit.  s2 - Wanneer een vooraf ingestelde afstand wordt overschreden, wordt de status van het rolluik aangepast.  s3 - Functie stuurt de gemeten waarde door naar de centrale.  s4 - Scheduler regelt dat de gemeten waarde om de seconde wordt gestuurd naar de centrale. |
| Post condition | De afstand van het rolluik kan worden gemeten en verstuurd naar de centrale. Hierdoor wordt de status van het rolluik bepaald. |
| Alternate path | Bij het meten en versturen van data met de ultrasoonsensor wordt geen gebruik gemaakt van alternate paths. |

# Hoofdstuk 2.3 – Beschrijving communicatieprotocol

## 2.3.1 – Data polling procedure (ontvangen van data)

Beschrijving alternatieve methode voor het verzamelen van de data die weergegeven moet worden op de display in de GUI.

* De centrale doet verzoeken aan de aangesloten besturingseenheden om de data te versturen.
* De besturingseenheden updaten de data in bepaalde variabelen die opgestuurd worden bij een data request van de centrale.
* De data die in de variabelen staat wordt geüpdatet aan de hand van een ISR veroorzaakt door de timer in de besturingseenheid zelf.

|  |  |
| --- | --- |
| Use Case | Data polling tussen de GUI en de besturingseenheden (ontvangen) |
| Pre-condition | Besturingseenheden zijn verbonden met de centrale |
| Standard path | s1 - De centrale controleert de pool met besturingseenheden.  s2 - Elke aangesloten besturingseenheid krijgt een identificatienummer van de centrale.  s3 - De centrale polled de seriële verbinding om te kijken of een besturingseenheid data doorgestuurd heeft.  s4 - De centrale koppelt de ontvangen data via een identificatienummer aan een besturingseenheid.  s5 - De centrale stuurt de data van de besturingseenheden als output naar het scherm (GUI). |
| Post condition | Er vind communicatie plaats tussen een besturingseenheid en de centrale. |
| Alternate path | a1@s1 - De pool van de besturingseenheden is leeg.  Exception: centrale breekt het communicatieproces af.  a2@s2 - Als een besturingseenheid al een identificatienummer heeft krijgt deze geen nieuwe.  a3@s2 - Als het identificatienummer al voorkomt moet er een nieuwe gegenereerd worden.  a4@s3 - Geen data verstuurd via de seriële verbinding  a5@s4 - Het identificatie nummer gekregen via de seriële verbinding bestaat niet binnen de besturingseenheden pool.  Exception: Het identificatienummer wordt als nieuwe besturingseenheid in de pool gezet en alle besturingseenheden worden opnieuw op connectiviteit gecontroleerd. |

## 2.3.2 – Connectieverzoek van de Arduino aan de centrale

|  |  |
| --- | --- |
| Use Case | Connectiviteit bepalen tussen besturingseenheid en de centrale |
| Pre-condition | Er wordt een besturingseenheid aangesloten, of er is een exceptie opgetreden waarbij de connectiviteit opnieuw gecontroleerd wordt. |
| Standard path | s1 - De centrale controleert de pool met besturingseenheden.  s2 - De centrale voegt de besturingseenheid toe aan de pool met besturingseenheden en stuurt een opdracht naar de besturingseenheid waarin een identificatie aan de besturingseenheid wordt gekoppeld. |
| Post condition | Besturingseenheid is gekoppeld aan de centrale. |
| Alternate path | a1@s2 - Als de pool niet leeg is worden alle besturingseenheden op connectiviteit gecontroleerd d.m.v. het opvragen van het identificatienummer. Als er een nieuwe besturingseenheid wordt toegevoegd, wordt stap s2 gewoon gedaan voor die besturingseenheid  Exception: Een identificatienummer wordt 2x gevonden. Dan wordt een van de besturingseenheden uit de pool verwijderd. |

## 2.3.3 – Het versturen van opdrachten naar de besturingseenheden

|  |  |
| --- | --- |
| Use Case | Opdrachten vanaf de centrale naar de besturingseenheden (verzenden) |
| Pre-condition | Besturingseenheden zijn verbonden met de centrale. |
| Standard path | s1 - Centrale stuurt een opdracht naar een besturingseenheid.  s2 - De besturingseenheid ontvangt de opdracht en voert deze uit.  s3 - besturingseenheid stuurt een bericht naar de centrale dat de opdracht succesvol uitgevoerd is. |
| Post condition | Er is een opdracht uitgevoerd door een besturingseenheid. |
| Alternate path | a1@s3 - Er wordt geen succes bericht verstuurd maar een negatief of een exception.  Exception: De fout wordt gelogd. |

# Conclusie

Voor Zeng ltd. moesten wij een domotica-oplossing uitwerken die het marktaandeel van Zeng ltd. zou vergroten. Om dit doel te bereiken, hebben wij een domotica-oplossing gemaakt die gebruikt kan worden voor zonneschermen en rolluiken.

Eerst zijn wij bezig gegaan met het maken van het functioneel ontwerp. Vervolgens zijn we begonnen met het programmeren van de besturingseenheden in C. Eerst hebben we de seriële verbinding voor de besturingseenheden gemaakt in C. Vervolgens hebben we de licht en temperatuursensoren geprogrammeerd om data te meten en om dit vervolgens naar de centrale door te sturen. Toen dit werkende was, hebben we de ultrasoonsensor zo goed mogelijk proberen te programmeren, zodat deze nauwkeurig de afstand kan meten en kan doorsturen. Het werkend krijgen van de sensoren was lastig maar is uiteindelijk wel gelukt. Helaas kregen we problemen met de sensoren wanneer deze in de scheduler gezet werden. Door de scheduler kregen wij valse waarden van de sensoren. Dit hebben wij weten op te lossen.

Naast de besturingseenheden, hebben we een centrale gebouwd met behulp van Python. In eerste instantie hebben wij een GUI gebouwd met behulp van tkinter, maar deze GUI vonden wij niet mooi genoeg. Hierop zijn we overgegaan op PyQT om een mooiere GUI te maken voor de centrale. Naast de GUI, hebben we met behulp van pySerial een seriële verbinding weten op te zetten met de besturingseenheden. Het was de bedoeling om commands naar de besturingseenheden te kunnen sturen vanuit de centrale. Hiervoor hebben we gekozen om de verschillende commands, zowel in de besturingseenheden als in de centrale, te koppelen aan waarden zoals 1, 2 & 3 etc. Hierdoor kunnen er commands zoals het oprollen van een zonnescherm, naar de besturingseenheden gestuurd en ook daadwerkelijk uitgevoerd worden.

Wij hebben het op een paar momenten, redelijk lastig gehad tijdens dit project. Gelukkig hebben wij dankzij goed samenwerken en doorzetten, deze problemen weten op te lossen en een goed eindproduct weten op te leveren. Wij hebben dankzij dit project, enorm veel geleerd van C, Python en het opzetten van een seriële verbinding tussen een besturingseenheid en een webapplicatie. Het was een lastig project maar wij hebben er individueel en als groepje, veel van geleerd.

# Bijlage 1 – Eerste ontwerp GUI

# 