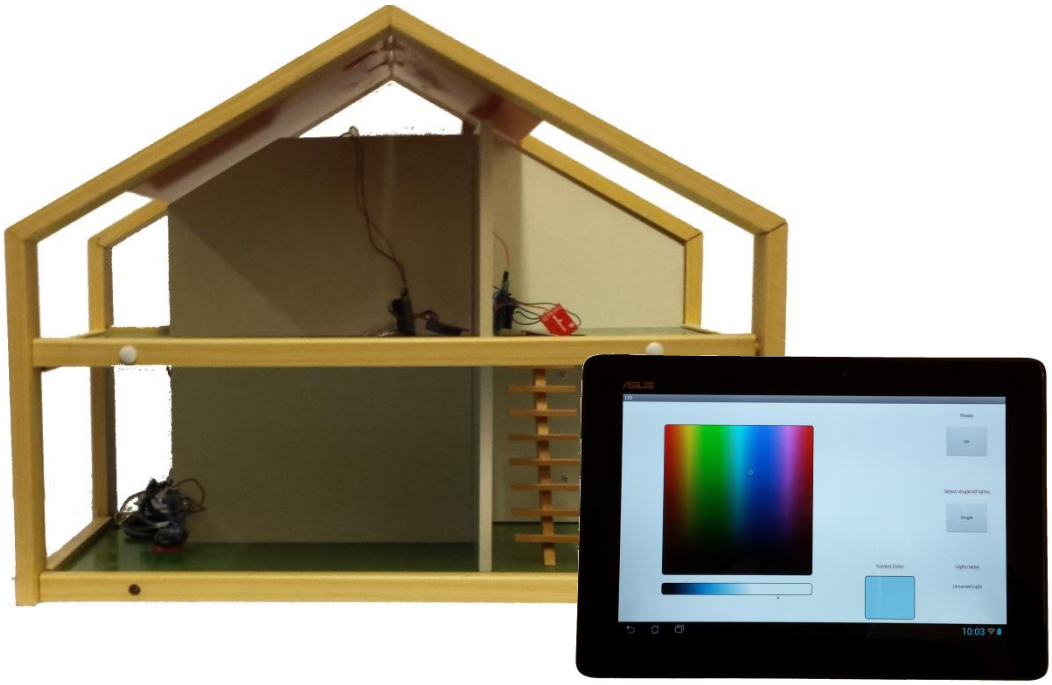
Domotica

Afstudeerverslag



ICT Automatisering

DE HAAGSE HOGESCHOOL

Datum:	6/4/2014
Bedrijf:	ICT Automatisering B.V.
Auteur:	David Ren
Begeleiders:	Khalid el Mrabet, Billy van der Burgt
Schoolbegeleider:	Paul Witte

Voorwoord

Op de opleiding Elektrotechniek van de Haagse Hogeschool wordt van iedere student verwacht om zijn/haar opleiding af te ronden met een afstudeeropdracht van 17 weken. Dit heb ik mogen uitvoeren bij ICT Automatisering te Son met Khalid el Mrabet en Billy van der Burgt als begeleiders.

Als onderdeel van de opdracht is uiteindelijk dit document tot stand gekomen, hiermee wil ik graag de lezer informeren over hoe de opdracht uitgevoerd is, de problemen die voor gekomen zijn, hoe deze opgelost zijn en het uiteindelijke resultaat.

Tijdens de afstudeeropdracht heb ik veel kunnen leren over het werken in een bedrijf en ik wil graag ICT Automatisering B.V. bedanken voor het bieden van deze mogelijkheid. Pok wil ik Khalid el Mrabet en Billy van der Burgt bedanken voor de ondersteuning die ik heb gekregen en tot slot wil ik graag Paul Witte bedanken voor zijn begeleiding als afstudeercoach.

Verklarende woordenlijst

Android	Besturingssysteem dat vooral gebruikt wordt voor tablets/smartphones.
Arduino	Een opensource-computerplatform, het is opgebouwd rond een Atmega microncontroller.
Backend	Het achterliggende systeem, de gebruiker kan hier niet direct bij.
Frontend	User Interface (HMI) voor de gebruiker om de Backend te bereiken
ETC	Estimated Time to Complete, geschatte tijd dat nog nodig is om een taak af te ronden.
GUI	Graphical User Interface, een grafische weergave van een interface (applicatie) die het voor de gebruiker gemakkelijker maakt om te gebruiken.
HMI	Human-Machine Interface, maakt interactie tussen mens en machine mogelijk.
HTTP	Hypertext Transfer Protocol, protocol voor de communicatie tussen een webclient en een webserver.
HSL	Hue, saturation en lightness, wordt gebruikt om kleuren te defineren, kan omgezet worden naar/van RGB-kleuren.
iOS	Besturingsysteem dat gebruikt wordt voor producten gemaakt door het bedrijf Apple Inc.
LED	Light-emitting Diode, een halfgeleider die licht geeft wanneer er in de juiste richting stroom doorheen gaat.
Library	Gevalideerde functies die gebruikt kunnen worden, hierdoor hoeft de programmeur niet op laag niveau te programmeren.
Linkedlist	Een datastructuur met een reeks data-elementen die naar de volgende of vorige element van hetzelfde type verwijst.
MoSCoW analyse	Wijze van prioriteiten stellen. Het is een afkorting van de volgende opsomming: **Must:** Deze eisen moeten gehaald worden **Should:** Deze eisen zijn gewenst, maar zijn niet essentieel **Could:** Deze eisen komen aan bod als er tijd over is **Won't:** Deze eisen worden niet uitgevoerd tijdens dit project
PCB	Printed Circuit Board, printplaat zonder bedrading en met in plaats daarvan koperen sporen.
PHP	Hypertext Preprocessor, een scripttaal, die bedoeld is om op web servers dynamische webpagina's te creëren. De taal wordt vooral gebruikt om informatie te verwerken tot HTTP.

Protocol	Regels en afspraken bij de communicatie tussen twee machines/apparaten.
Raspberry Pi	Een single board computer met de afmetingen van een bankpas, op de hoogte na.
Repository	Centrale digitale opslagplaats voor programma's die geïnstalleerd kunnen worden.
RDS	Radio Data System, hierbij wordt digitale informatie meegestuurd met een FM-radiosignaal (meestal zendernaam).
RGB	Red, green en blue, kleurcodering. Een kleur wordt uitgedrukt door rood groen en blauw te combineren. Alle drie de kleuren kunnen waardes van 0 tot 255 aannemen. Dit getal wordt gebruikt om de aanwezige hoeveelheid van een kleur te bepalen.
Single board computer	Computersysteem gebouwd op een enkele printplaat, beschikt over een processor, geheugen en i/o (input/output).
Smartphone	Mobiele telefoon met computermogelijkheden.
Tablet	Computer van klein formaat met een aanraakscherm en geen toetsenboord.
Web server	Ontvangt HTTP-verzoeken en voert de ontvangen commando's uit.
WLAN	Wireless Local Area Network, draadloos netwerk , ook wel WiFi genoemd.
XBee	Radio modules die met andere modules (van hetzelfde type) draadloos communiceert.
XML	Extensible Markup Language, wordt vaak gebruikt om gegevens op te slaan, is zowel leesbaar voor de mens als de machine.
XML Parser	Programma (of code) dat bestanden in XML format kan uitlezen en dit kan weergeven of opslaan in een nuttige vorm.
XML Tree	Structuur van een XML bestand

Samenvatting

Domotica wordt tegenwoordig steeds meer toegepast om apparatuur aan te sturen in een huis. Binnen ICT Automatisering B.V. is hier een begin gemaakt in de vorm van een miniatuur huis, dit moet nu uitgebreid worden met meerdere modules (Bij de aanvang van de opdracht alleen module LED aanwezig). Het uiteindelijke geheel zal dienen als een demonstratie opstelling op beurzen en bedrijfsdagen. De opdracht zal bestaan uit de volgende drie onderdelen:

- Toevoeging van een nieuw module en vervanging van de voeding
- Telefoon/tablet applicatie uitbreiden of indien nodig opnieuw opbouwen
- Beveiliging van het systeem en nieuwe modules toevoegen indien er tijd over is

Voordat er ontwerpen gemaakt konden worden moest eerst een onderzoek uitgevoerd worden naar wat er al aanwezig was in het bestaande systeem. Dus is de onderzoeksvraag als volgt: Wat is er al aanwezig en wat moet aangepast worden om nieuwe modules toe te voegen?

Het systeem bestaat uit de Frontend en de Backend. De Frontend bestaat uit een (Android) applicatie geschreven in Javascript en de Backend bestaat uit een web server (Raspberry Pi), die de data vanuit de applicatie verwerkt, en de modules zelf (Arduino). De Raspberry Pi communiceert met de Arduino door middel van XBee. Om meer modules toe te voegen moet een aantal C bestanden op de web server aangepast moeten worden en er moet code geschreven worden voor het Arduino gedeelte (code verschilt per module), daarna zal een GUI voor de module gemaakt moeten worden op de applicatie.

Nieuwe modules die in deze opdracht zijn gerealiseerd zijn de temperatuursensor en de radio. In het bestaande systeem was nog geen bi-directionele communicatie aanwezig, maar dit is noodzakelijk voor modules zoals de temperatuursensor. Dit is dus als eerste gerealiseerd en hiervoor is het protocol (opgesteld door vorige afstudeerder) uitgebreid met een antwoordframe, foutenlijst en een commandolijst. Hiernaast is ook een herontwerp van de web server gemaakt omdat de bestaande functies er inflexibel waren en alleen konden omgaan met de module LED. In het herontwerp is ook de manier voor opslag van modules veranderd naar XML formaat (hiervoor was het een tekstbestand). Nadat de temperatuursensor toegevoegd was aan de Backend, werd er een afweging gemaakt tussen de applicatie uitbreiden of het opnieuw opbouwen. De uiteindelijk keuze is gevallen op uitbreiden omdat een applicatie vanuit niets opbouwen veel tijd zou kosten en ook omdat de expertise van de afstudeerder niet in dit gebied lag. Ten slotte was er nog een afweging geweest tussen het uitvoeren van een onderzoek naar beveiliging of het toevoegen van een tweede module, de voorkeur van de opdrachtgever ging hierbij uit naar de radio module. Bij de realisatie van de radio enkele problemen voorgekomen dat nader besproken wordt in het verslag.

Van de gestelde eisen waren de belangrijkste het leveren van een werkende demonstratie opstelling die gebruikt kan worden op bedrijfsdagen en beurzen, toevoegen van een nieuwe module en het uitbreiden of opnieuw opbouwen van de applicatie. Een niet behaalde eis is het onderzoek naar beveiliging, maar dit is geen essentieel onderdeel om het systeem te laten werken. Dus mijn conclusie is: Het huidige systeem voldoet aan de meeste van de gestelde eisen en kan dienen als een demonstratie opstelling voor beurzen en open dagen.

Een aanbeveling bij verdere ontwikkeling aan de demonstratie opstelling is het toevoegen van een beveiliging voor de web server. Verder is er ook nog geen foutenafhandeling aanwezig, foutmeldingen worden gegenereerd door de modules maar worden nog niet afgehandeld door de web server. Ten slotte is het ook aangeraden de applicatie, indien het opnieuw opgebouwd word, te herbouwen in een ontwikkelomgeving met een grafische weergave voor het GUI ontwerp.

Document historie

Versie	Status	Datum	Omschrijving
0.1	Initieel	11-03-2014	Opzet document + lay-out
0.2	Concept	04-03-2014	Hoofdstukindeling + paragraafindeling
0.3	Concept	17-04-2014	Hoofdstuk Inleiding
0.4	Concept	18-04-2014	Paragrafen Bedrijf, Opdracht en Werkmethode
0.5	Concept	22-04-2014	Hoofdstuk Onderzoek en Ontwerpen
0.6	Concept	23-04-2014	Paragraaf Onderzoek
0.7	Concept	28-04-2014	Paragraaf Ontwerpen
0.8	Concept	29-04-2014	Hoofdstuk Realisatie
0.8.1	Concept	01-05-2014	Feedback van Khalid el Mrabet verwerkt
0.9	Concept	08-05-2014	Feedback van Khalid el Mrabet en Billy van der Burgt verwerkt
0.9.1	Concept	20-05-2014	Verklarende woordenlijst gemaakt
0.9.2	Concept	26-05-2014	Conclusie toegevoegd
0.9.3	Concept	27-05-2014	Rework feedback Paul Witte
0.9.4	Concept	30-05-2014	Voorwoord + samenvatting
0.9.5	Concept	31-05-2014	Samenvatting + Verklarende woordenlijst gemaakt
1.0	Final	05-06-2014	Feedback verwerkt

Distributie

Versie	Datum	Aan	Bedrijf

Inhoudsopgave

Figurenlijst

1. Inleiding

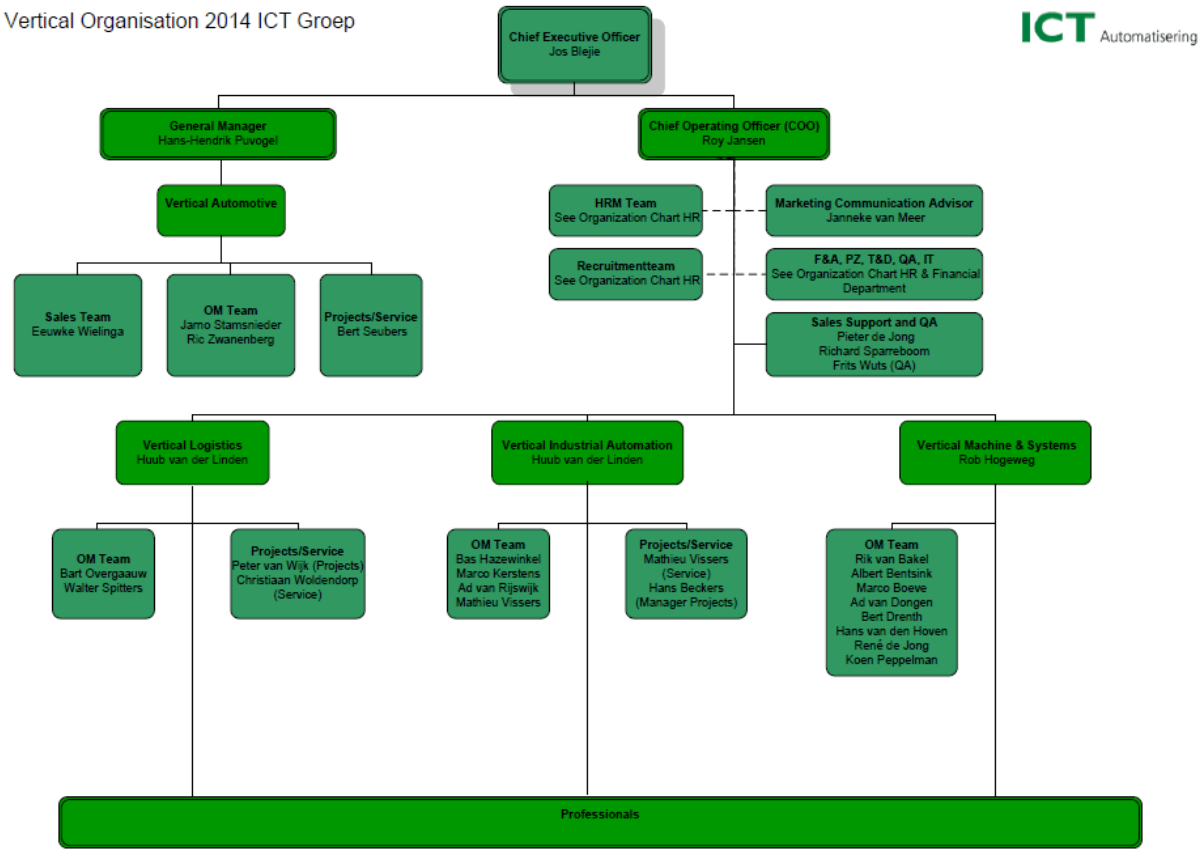
In dit hoofdstuk wordt het bedrijf, de opdracht en de werkmethode beschreven. Onder de paragraaf “Opdracht” staan naast de opdrachtomschrijving ook de MoSCoW analyse en de lijst met betrokken personen.

1.1. Het bedrijf

ICT Automatisering B.V. (Hierna aangeduid als ICT) is opgericht in 1978 en staat voor Industriële Computer Toepassingen. Het bedrijf beschikt over verscheidene expertises en kan hiermee oplossingen op maat leveren voor haar klanten. Het bedrijf is onderverdeeld in zes verticals en elke vertical hanteert een verschillende deel van de markt[Bron: Bronnenlijst 1.]. De verticals zijn:

- Automotive
- Logistics
- Industrial Automation
- Machine & Systems
- Health Care
- Energy

Het project “Domotica” behoort tot de vertical “Machine & Systems”, dit vertical levert zowel specifieke kennis voor het realiseren van machine besturingen als ook gehele lijnbesturingen voor de discrete productielijnen.[Bron: Bronnenlijst 2.]

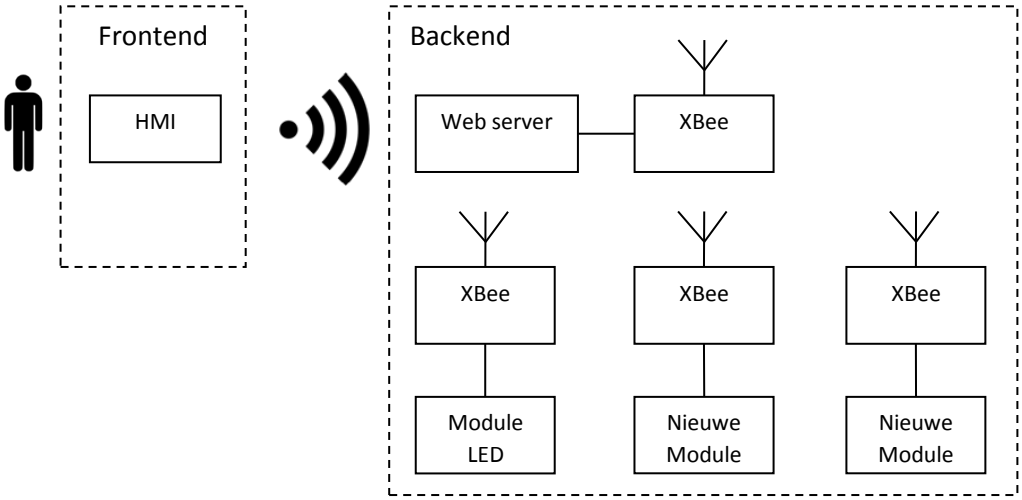


Figuur 1: Organogram ICT Automatisering

1.2. Opdracht Domotica

1.2.1. Aanleiding opdracht

Domotica wordt tegenwoordig steeds meer toegepast om apparatuur aan te sturen in een huis. Binnen ICT Automatisering B.V. is hier een begin gemaakt in de vorm van een miniatuur huis, dit wordt nu uitgebreid worden met meerdere modules. Het uiteindelijke geheel zal dienen als een demonstratie opstelling voor ICT op beurzen en bedrijfsdagen. Figuur 2 geeft de situatie weer bij de aanvang van de opdracht. Met een (Android)applicatie (HMI)kan met WLAN een verbinding maken met een Raspberry Pi (web server). De Raspberry Pi stuurt dan de Arduino (Module) aan met behulp van XBee. Bij de aanvang van de opdracht was alleen de module LED aanwezig.



Figuur 2: Huidig systeem

1.2.2. Opdrachtomschrijving

Verder ontwikkelen van een Domotica systeem waarmee met een telefoon/tablet apparaten zoals lichten, deuren en gordijnen bediend kunnen worden. Er is door ICT een specificatie van een protocol opgesteld dat gebruikt kan worden om apparatuur met behulp van Xbee aan te sturen. In deze opdracht is het de bedoeling dat ook de applicatie op de telefoon/tablet aangepast wordt en dat het protocol uitgebreid wordt voor meerdere modules. In de originele opdracht zijn de uit te voeren onderdelen als volgt:

- Smartphone/Tablet applicatie uitbreiden of indien nodig opnieuw opbouwen (Frontend)
- Beveiliging van het systeem
- Toevoeging van nieuwe modules indien er tijd over is (Backend)

Doordat de achtergrond van de afstudeerder meer neigt naar de hardware kant (Elektrotechniek), is de volgorde van de onderdelen in nader bespreking met de opdrachtgever iets aangepast:

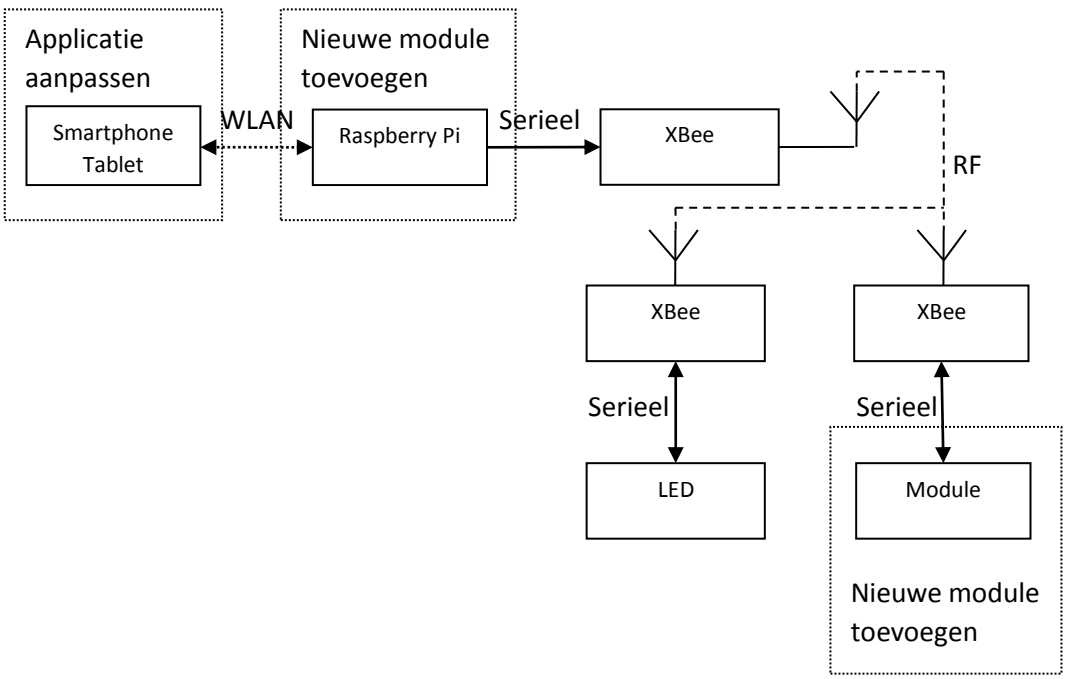
- Toevoeging van een nieuw module en vervanging van de voeding
- Telefoon/tablet applicatie uitbreiden of indien nodig opnieuw opbouwen
- Beveiliging van het systeem en nieuwe modules toevoegen indien er tijd over is

In het eerste deel moet kennis worden opgedaan over de Backend en deze uitgebreid worden met een nieuw module. Ook moet de huidige voeding vervangen worden door een kleinere, op het moment is dit een computervoeding met een moederboord. Daarna wordt er onderzocht hoe de telefoon/tablet applicatie het beste uit te breiden is of indien nodig zelfs helemaal opnieuw opbouwen. Hierbij moet wel gebruik gemaakt worden van het al opgestelde protocol voor de communicatie met de Raspberry Pi. Vervolgens moet dit systeem worden ontworpen en ontwikkeld waarbij de nadruk ligt op:

- Opzetten van een duidelijke interface voor het gebruik van het systeem
- Uitbreidbaarheid

Uiteindelijk, indien er nog tijd over is, wordt of een onderzoek naar beveiligingsmogelijkheden uitgevoerd en toegepast of een tweede module wordt toegevoegd aan het miniatuurhuis.

Figuur 3 geeft de stippellijn de reikwijdte van de opdracht weer, de Frontend bestaat uit een smartphone/tablet en de Backend bestaat uit de Raspberry Pi, Xbee en Arduino Pro Mini. Bij het toevoegen van een nieuwe module worden gedeeltes in de Backend aangepast. Hierna wordt de applicatie aangepast met een GUI voor de nieuwe module en ten slotte worden de Backend en Frontend met elkaar geïntegreerd.



Figuur 3: Reikwijdte

1.2.3. MoSCoW Analyse

Must

- Uitwerken user interface op de smartphone/tablet
- Gebruikershandleiding voor de user interface
- Hardware aansturen met Xbee en Raspberry Pi
- Toevoegen van een nieuwe module
- Werkende demonstratie opstelling (aansturen van modules vanuit de applicatie)
- Documentatie opleveren voor verdere ontwikkeling
- Voeding vervangen

Should

- Onderzoek beveiligingssysteem (indien tijd over)

Could

- Meerdere modules toevoegen (indien tijd over)

Won't

- Bestaande hardware (behalve de voeding) aanpassen

1.2.4. Betrokken partijen

Functie: Afstudeerder
Naam: David Ren
Mail: d.ren2207@gmail.com

Functie: Opdrachtgever
Naam: Billy van der Burgt
Mail: Billy.van.der.Burgt@ict.nl

Functie: Bedrijfsmentor
Naam: Khalid el Mrabet
Mail: Khalid.el.Mrabet@ict.nl

Functie: Afstudeercoach
Naam: Paul Witte
Mail: p.m.witte@hhs.nl

1.2.5. Activiteiten

Doordat er met Scrum (Hoofdstuk 1.3) gewerkt wordt zijn de activiteiten van dit project in sprints verdeeld. Hierdoor komen de activiteiten van Scrum terug in elke sprint, die zijn als volgt:

- Daily meetings
- Sprint demo
- Retrospective
- Sprintplanning

De activiteiten die voor school uitgevoerd moeten worden zijn:

- Plan van aanpak
- Afstudeerverslag
- Posterpresentatie
- Eindpresentatie

Voor de toevoeging van nieuwe modules zijn de activiteiten als volgt:

- Bedenken van mogelijke modules
- Ontwerp van de modules
- Modules toevoegen
- Bestuderen documentatie (voorganger)
- Bestuderen code Backend (voorganger)
- Code aanpassen/schrijven
- Module testen
- Documentatie

Voor het uitbreiden en aanpassen van de smartphone/tablet applicatie zijn de activiteiten als volgt:

- Javascript bestuderen
- Doorlezen documentatie (voorganger)
- Bestuderen code applicatie (voorganger)
- Applicatie aanpassen/opnieuw opbouwen
- Testen
- Documentatie

Indien er nog tijd over is:

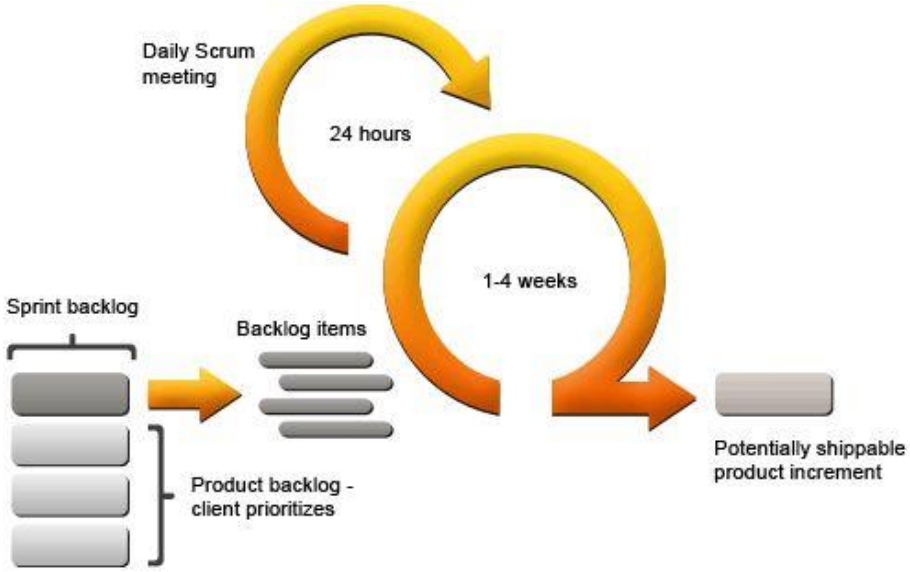
- Onderzoek beveiligingsmogelijkheden
- Afweging beveiligingsmogelijkheden
- Beveiliging toepassen
- Toevoegen van meerdere modules
- Documentatie

1.3. Werkmethode

Bij ICT Automatisering wordt gebruik gemaakt van “Scrum”, dit is een agile methode. Dit is een lenige methode dat met korte overzichtelijke periodes werkt (iteraties).

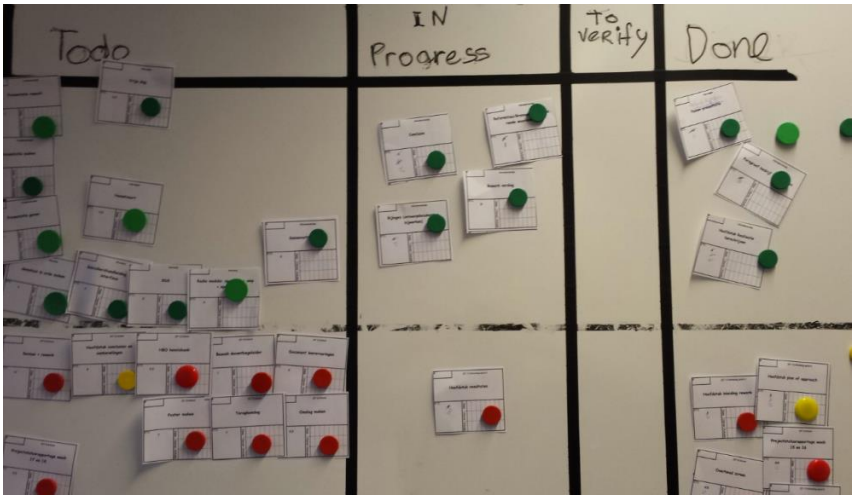
1.3.1. Scrum

Scrum is een methode die werkt met korte overzichtelijke periodes, genaamd “sprints”. Een sprint is een periode van twee tot vier weken waarbij de ontwikkeldoelen vastgelegd zijn. De hoofddoelen van een sprint bestaan in de vorm van “stories”, die worden verder verdeeld in taken waarbij een geschatte tijdsduur hoort. De “definition of done” van de story geeft aan wanneer de betreffende story af is. De stories die bedacht zijn worden gedocumenteerd in het “product backlog”, tijdens het plannen van een sprint worden hieruit de stories van de aankomende sprint gekozen. Dit document wordt constant aangepast gedurende het gehele project.



Figuur 4: Scrum methode

Tijdens de sprint wordt elke ochtend een “daily meeting” gehouden, hierbij worden de taken op het “Scrum board” (Figuur 5) verschoven en voor deze taken worden een nieuwe tijdschatting gemaakt. Naast het verschuiven van de taken worden ook “blocking issues” aangegeven (Indien aanwezig). Dit zijn problemen waarbij hulp ingeschakeld moet worden van derde partijen.



Figuur 5: Voorbeeld Scrum board

Figuur 6 geeft een “burndown chart” weer, dit wordt dagelijks aangepast en geeft de huidige situatie van de sprint weer. De verticale as geeft resterende uren weer en de horizontale as de dagen.

Figuur 6: voorbeeld burndown chart

Aan het eind van elke sprint wordt een demonstratie gegeven van de behaalde resultaten van een sprint. Dit kan bestaan uit een presentatie, een document en/of een werkelijke demonstratie. Naast de demonstratie worden ook een “retrospective” en de “sprint planning” gehouden. Bij de retrospective wordt een lijst opgesteld van plus- en verbeterpunten, hierbij wordt ook gekeken of de verbeterpunten van de voorgaande retrospective verholpen zijn. Bij de sprint planning worden nieuwe stories vanuit het product backlog gekozen, wanneer de sprintplanning goedgekeurd is (door de opdrachtgever) geeft het Scrum team een commitment, hierbij wordt beloofd dat wat afgesproken is ook daadwerkelijk opgeleverd wordt.

Als tijdens de sprint al blijkt dat een taak niet af zal komen, wordt er eerst samen met het Scrum team en de “Scrum master” of de demonstratie toch nog gepresenteerd kan worden op een creatieve manier of hulp hiervoor ingeschakeld kan worden. Wanneer dit niet mogelijk is wordt met het “product owner” besproken of de taak geschrapt kan worden en ten slotte als het laatste oplossing het afbreken van een sprint. [Bronnenlijst 3. en 4.]

1.3.2. Rollen bij Scrum

Product owner

Dit is de opdrachtgever/klant en compileert alle stories in een product backlog, tijdens de sprint planning geeft de product owner de prioriteiten van de stories aan.

Scrum master

Dit is de coach van het Scrum team, elke dag wordt door de Scrum master een daily meeting gehouden om de voortgang van de sprint te bespreken. De Scrum master zorgt er ook voor dat het Scrum team de beste omgeving bezit om de doelen van een sprint te behalen. Ook zorgt de Scrum master ervoor dat het team zo min mogelijk gestoord wordt door invloeden van derde partijen.

Scrum team

Dit is het team van ontwikkelaars dat de taken van een story uitvoert, de volgorde van de taken worden door het team zelf bepaald. In het Scrum team zijn er geen rollen en dus zijn de taken onderling uitwisselbaar.

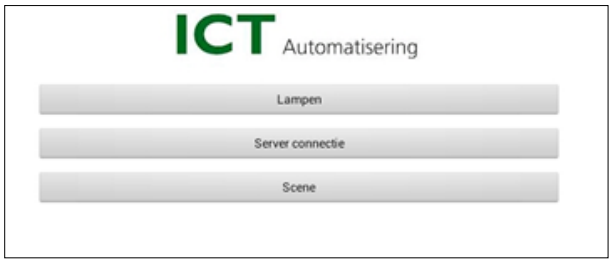
2. Onderzoek van het bestaande systeem

In dit hoofdstuk wordt het onderzoek van het bestaande systeem besproken. Het onderzoek is gedaan om erachter te komen van wat al aanwezig was bij de aanvang van het project. De gemaakte ontwerpen zijn gebaseerd op dit onderzoek. Bij dit onderzoek hoort de volgende onderzoeksvraag:

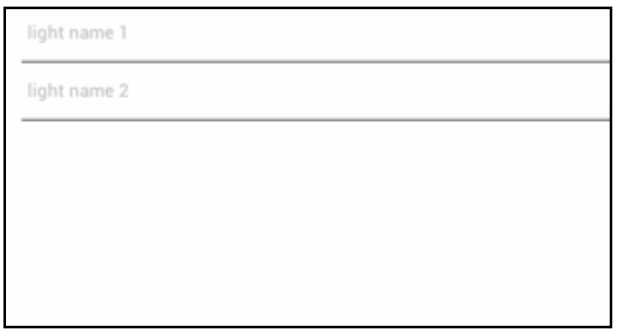
Wat is er al aanwezig en wat moet aangepast worden om nieuwe modules toe te voegen?

2.1. Frontend

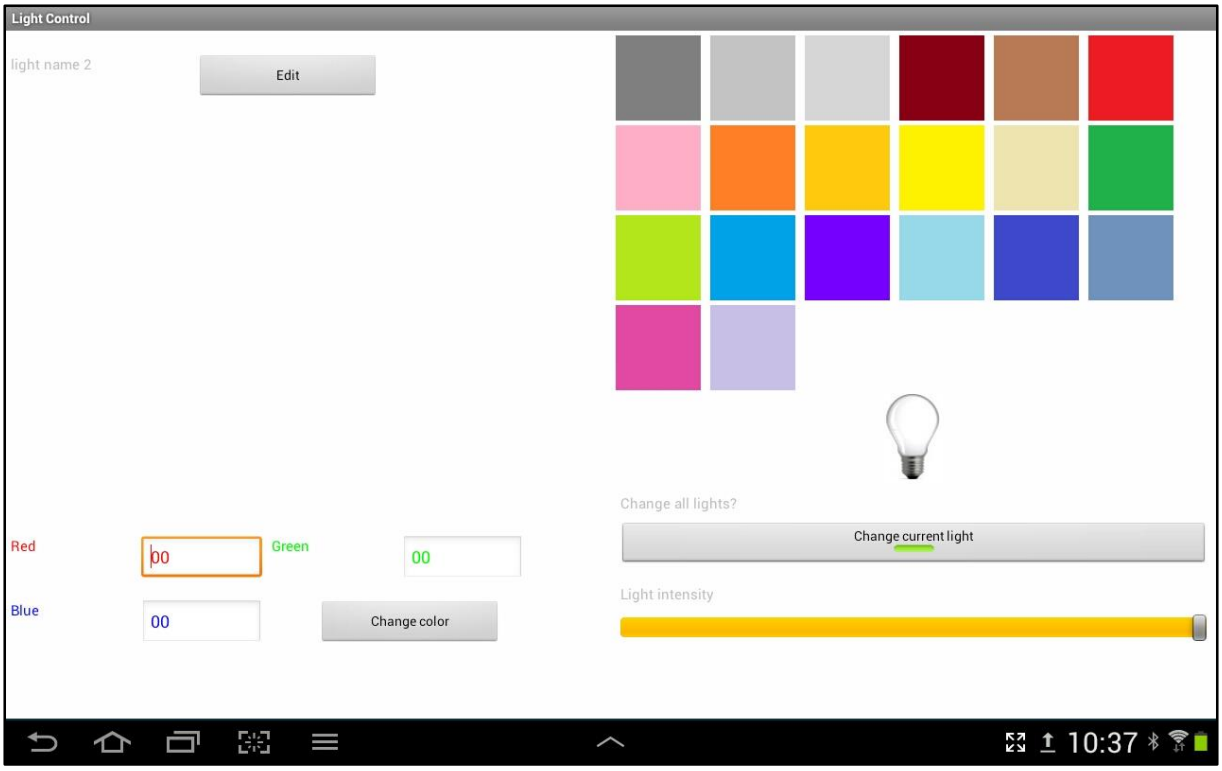
De Frontend is het gedeelte van het systeem waarmee de gebruiker de Backend aanstuurt. Het is in dit geval een applicatie op een tablet (Android). De applicatie is geschreven met het programma Titanium Studio, dit is een ontwikkelomgeving waarbij Javascript de programmeertaal is[6]. Figuur 7 geeft het startscherm van de applicatie weer, dit wordt vertoond bij het starten van de applicatie. Bij het drukken van de knop “Lampen” worden alle LED modules vertoond in een lijst (Figuur 8) en met de “Server connectie” knop kunnen gegevens van de Backend worden ingesteld zodat een connectie gemaakt kan worden met de web server (Figuur 10). Na het klikken op een naam in de lijst van figuur 8 wordt een nieuw scherm geopend voor de geselecteerde module.



Figuur 7: Startscherm



Figuur 8: Lijst van LED modules



Figuur 9: LED besturing scherm

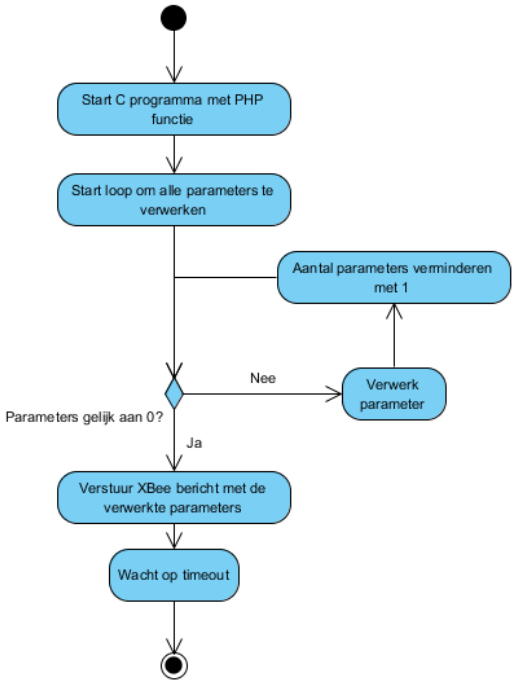
Figuur 9 geeft het besturing scherm van de LED module weer, met de gloeilamp kan een LED aan of uit worden gezet. De kleur kan veranderd worden door waardes van 0 tot FF (Hexadecimaal) in de velden linksonder het scherm in te vullen of door op een van de kleurvelden rechtsboven te klikken. De balk rechtsonder regelt de intensiteit van de LED en ten slotte wordt met de knop “Change current light” gekozen om een enkele module of alle modules tegelijk aan te sturen.

Op de Backend draait een web server (Raspberry Pi) waarmee de applicatie verbinding kan maken met een HTTP Client. De gegevens van de web server worden ingesteld in het configuratiescherm in figuur 10. De ingestelde configuraties worden dan gebruikt om met de web server een verbinding te maken.

Figuur 10: Configuratiescherm voor gegevens Backend

2.2. Backend

De Backend bestaat uit de Raspberry Pi, XBee en Arduino. De Raspberry Pi verwerkt de ontvangen data (van de Frontend) en verstuurt dit door naar de modules (Arduino). De Frontend gebruikt functies in een PHP bestand waarmee een C programma gestart kan worden met de betreffende parameters. Het onderstaande activiteitsdiagram (Figuur 11) geeft het proces weer voor het verwerken van data. Hiernaast worden bij het opstarten van de Raspberry Pi adressen van nieuwe XBee modules opgevraagd en opgeslagen in een tekstbestand, deze adressen krijgen een nummer toegekend die door de Frontend gebruikt wordt om de juiste module aan te spreken.



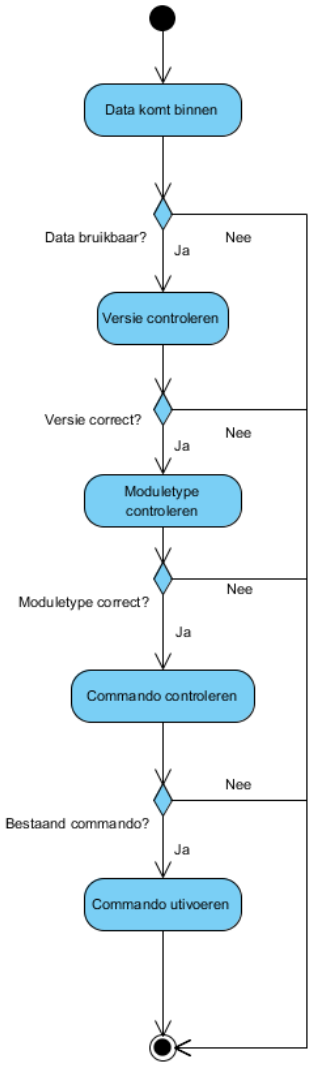
Figuur 11: Activiteitsdiagram Raspberry Pi

Op de Raspberry Pi zijn er verscheidene C bestanden aanwezig waar het C programma gebruik van maakt, dit zijn:

- **processData.c**
  Verwerkt de ontvangen parameters van de Frontend
- **XBeeModule.c**
  Hierin staan functies die te maken hebben met XBee, zoals het opvragen van adressen en het versturen van data naar een specifieke XBee. Dit bestand is afhankelijk van libxbee.
- **linkedlist.c**
  Hierin staan de functies die gebruikt worden om de gevonden XBee modules weg te schrijven of in te lezen.
- **log.c**
  Hierin staan functies om een log te maken voor debug doeleinden.

Als de data verwerkt en verstuurd is naar de module (Arduino) dan sluit het C programma af en is de rol van de Raspberry Pi afgelopen.

Figuur 12 geeft het activiteitsdiagram van de modules weer, de ontvangen data (vanuit de Raspberry Pi) wordt door de Arduino verwerkt en het ontvangen commando wordt dan uitgevoerd.



Figuur 12: Activiteitsdiagram Arduino

De volgende bestanden zijn op de Arduino aanwezig:

- **De "main" (voorbeeld LED.ino)**
  Initialisatie van de module en wacht daarna op commando's van de Frontend.
- **processData.ino**
  Hierin staan de functies voor het aansturen van de module en het verwerken van ontvangen data, voor de module LED zijn ze als volgt:
  - Verwerken van ontvangen data
  - LED Aan/uit zetten
  - LED Kleur veranderen
  - LED Intensiteit veranderen
- **XBeeModule.ino**
  Hierin staan te functies die te maken hebben met de XBee modules, hierbij horen initialisatie van de XBee en het valideren van ontvangen berichten (van de Raspberry Pi).

2.3. Conclusie onderzoek

In de paragrafen 2.1. Frontend en 2.2. Backend is beschreven hoe het huidige systeem in elkaar zit. Om nieuwe modules toe te voegen moeten de onderstaande gedeeltes aangepast/gemaakt worden:

- **Applicatie**
  - User interface maken voor de nieuwe module
- **Raspberry Pi**
  - PHP bestand uitbreiden
  - processData.c uitbreiden
  - C bestand voor de nieuwe module
- **Arduino**
  - processData.ino aanmaken (functies verschillen per module, naam blijft hetzelfde)

XBeeModule.c en linkedlist.c moeten ook eenmalig aangepast worden om het moduletype te kunnen opvragen (bi-directionele communicatie), in de voorgaande opdracht is er vanuit gegaan dat alleen het moduletype LED aanwezig is. Op de Arduino wordt ook XBeeModule.ino uitgebreid met een functie voor het versturen van XBee berichten.

3. Ontwerp voor het nieuwe systeem

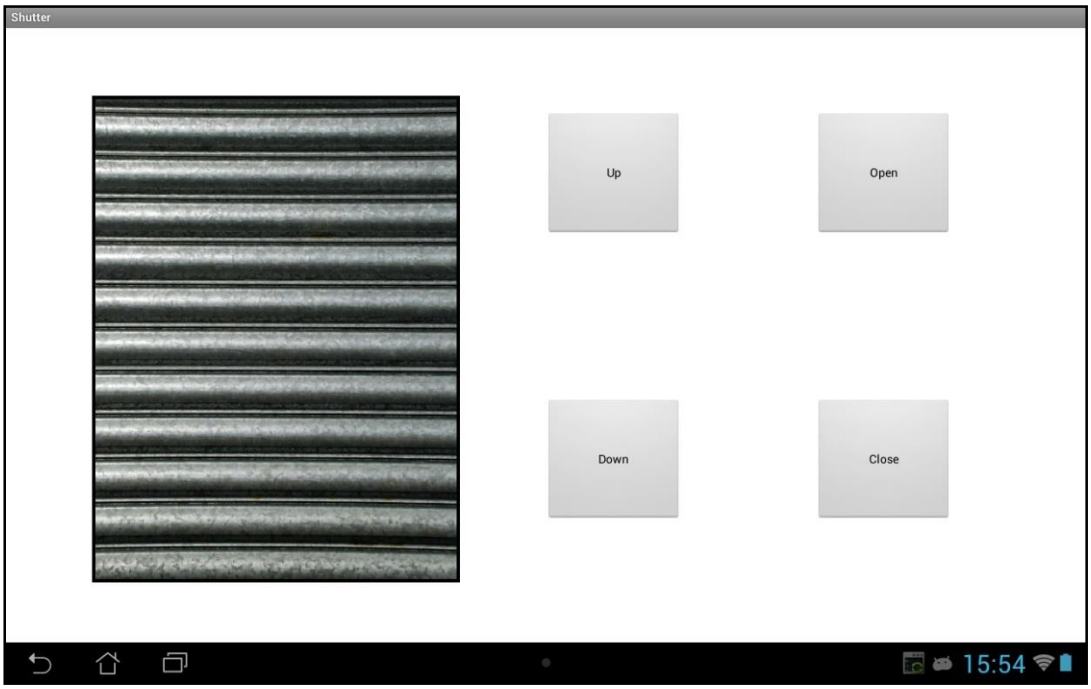
Voordat de modules gerealiseerd worden, moeten er eerst ontwerpen gemaakt worden. Van de bedachte ideeën (Bijlage ”Nieuwe modules”) zijn er een aantal uitgekozen om verder uit te werken, de gekozen modules zijn als volgt:

- Temperatuursensor
- Radio
- Rolluik

Om deze modules toe te voegen zal ten eerste de Backend uitgebreid moeten worden met bi-directionele communicatie en code voor de nieuwe module, hiervoor wordt het protocol uitgebreid met een antwoordframe (Bijlage “Protocol”). Behalve dit antwoordframe bezit het protocol nu ook over een commando- en foutenlijst. Hierna wordt voor de Frontend een GUI ontworpen en gerealiseerd en dit wordt geïntegreerd met de Backend. Naast het toevoegen van nieuwe modules wordt ook een herontwerp van de Raspberry Pi (gedeelte van Backend) gemaakt.

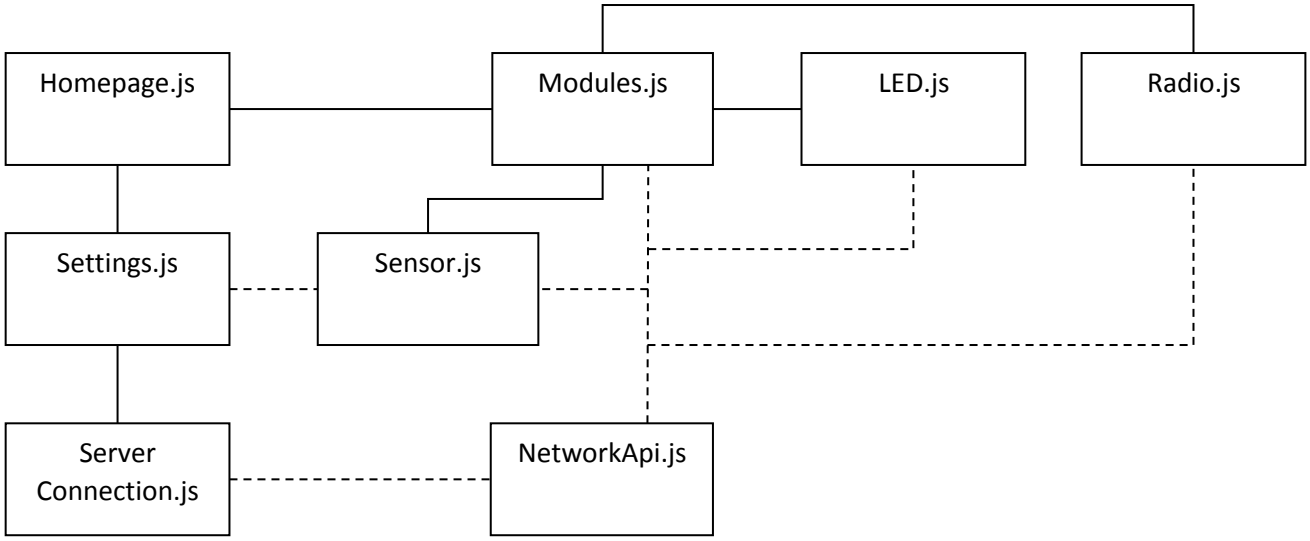
3.1. Frontend

Bij het toevoegen van een nieuwe module moest de applicatie aangepast worden, hier moest eerst een beslissing gemaakt worden tussen het uitbreiden van de applicatie of de applicatie opnieuw opbouwen. De keuze was uiteindelijk gevallen op het uitbreiden en de applicatie opnieuw opbouwen te bewaren voor een andere opdracht. De belangrijkste reden voor deze beslissing is de tijdsfactor, maar ook de expertise van de afstudeerder speelt hierbij een rol. Voor elke module is een nieuwe GUI ontworpen, de ontwerpen voor modules worden vertoond en uitgelegd in 4. Realisatie. Hiervan is het rolluik niet gerealiseerd en zal hieronder worden vertoond.



Figuur 13: Rolluik GUI

Figuur 13 geeft de GUI van het rolluik weer. Het rolluik is op verschillende manieren te bedienen, bij het inhouden van de “Up” of “Down” knop zal het bewegen in de gekozen richting. “Open” of “Close” zorgt ervoor dat het rolluik geheel open of dicht gaat en uiteindelijke kan ook het rolluikplaatje gesleept worden naar de gewenste positie.



Figuur 14: Structuur applicatie

Figuur 14 geeft het structuur van de applicatie weer, dit zijn alle gebruike bestanden in de applicatie. De solide lijnen geven aan hoe de applicatie genavigeerd kan worden, de gebruiker zal bij het starten van de applicatie starten bij het scherm van Homepage.js. De stippellijnen geven aan welke bestanden in de achtergrond gebruikt worden. Hieronder volgt een korte uitleg van alle bestanden:

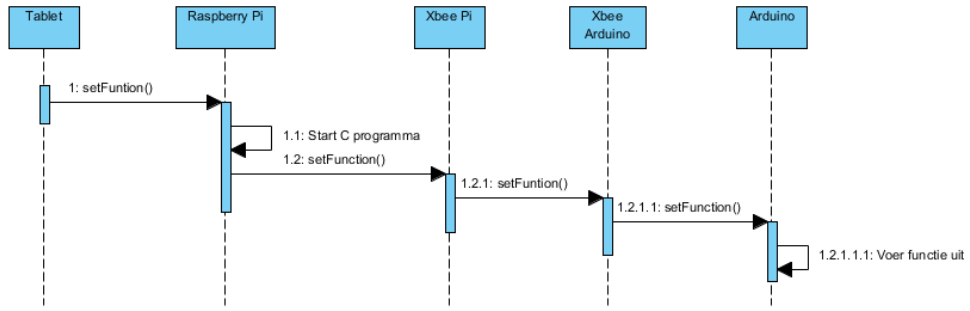
- **Homepage.js**
  Startscherm waar de gebruiker terechtkomt bij het opstarten van de applicatie
- **Settings.js**
  Hierin worden alle configuratie gezet die gebruikt zullen worden in de applicatie. Er zijn twee knoppen aanwezig:
  - Connection Settings: Opent ServerConnection.js
  - Celsius/Fahrenheit: Keuze uit Celsius/Fahrenheit voor temperatuurweergave.
- **ServerConnection.js**
  In dit scherm worden de instellingen die nodig zijn om met de Raspberry Pi te verbinden gezet. De functionaliteiten zijn precies hetzelfde als bij de aanvang van de opdracht
- **Modules.js**
  Lijst van beschikbare modules die door de gebruiker geselecteerd kunnen worden, na de selectie wordt het betreffende scherm geopend. Ook is een “Refresh list” knop aanwezig om nieuwe modules toe te voegen of niet meer bestaande modules weg te halen.
- **LED.js**
  Besturing voor de LED, zie “4.3. LED”.
- **Sensor.js**
  Weergave van de temperatuursensor, zie “4.2. Temperatuursensor”.
- **Radio.js**
  Besturing voor de radio, zie “4.4. Radio”.
- **NetworkApi.js**
  Hierin staan de functies die aangeroepen worden om te verbinden met de Raspberry Pi, dit wordt uitgevoerd door middel van een HTTP Client.

3.2. Backend

De Backend bestaat uit een web server (Raspberry Pi en XBee)en de modules (Arduino en XBee). Zoals al vermeld staat in “2.3. Conclusie onderzoek” moet bi-directionele communicatie toegevoegd worden aan het systeem om een nieuwe module te realiseren.

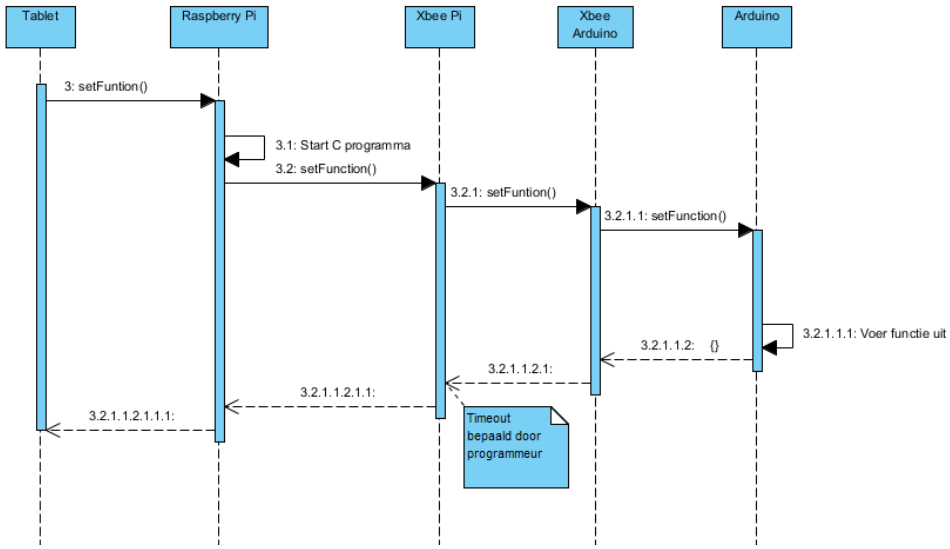
3.2.1. Bi-directionele communicatie

Bij de aanvang van de opdracht is het alleen mogelijk om berichten naar modules toe te sturen en van de modules wordt dan ook geen antwoord verwacht, maar om modules zoals temperatuursensoren te kunnen realiseren is bi-directionele communicatie noodzakelijk.



Figuur 15: Sequentiediagram met alleen sturen

Figuur 15 geeft aan hoe de communicatie was bij de aanvang van de opdracht, het programma sluit zich af na het versturen van een XBee bericht. De communicatie zoals het nu is wordt weergeven in figuur 16, het programma wacht op een antwoord (tot aan een ingestelde tijdslimiet, dit kan zolang duren als de programmeur het wilt) van de module voordat het zal afsluiten.

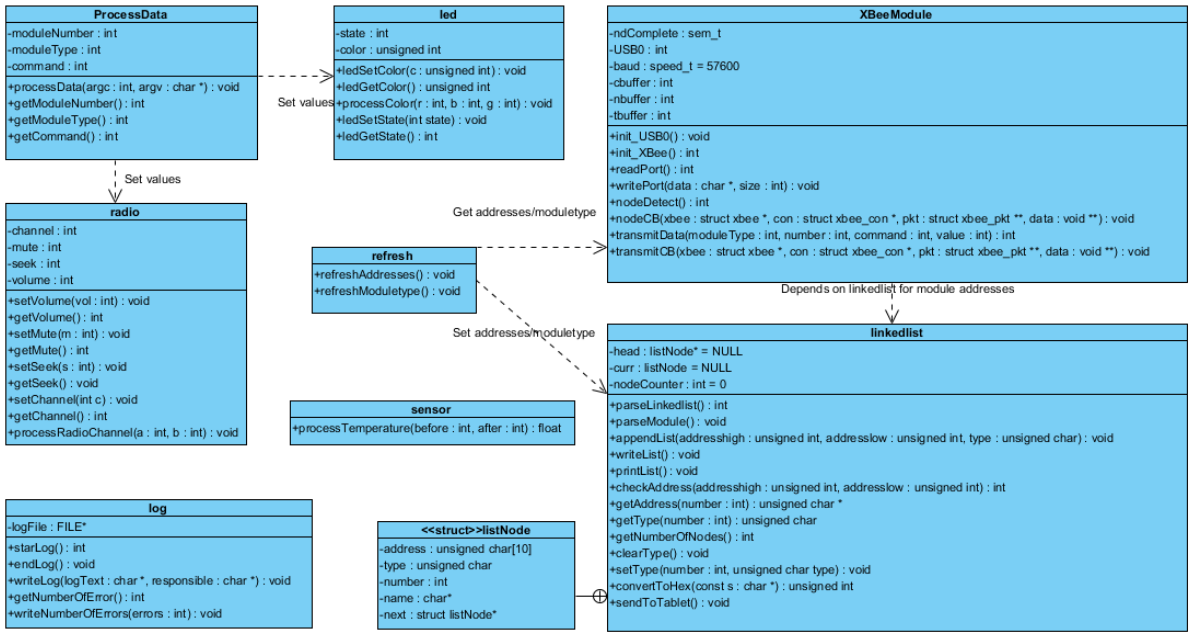


Figuur 16: Sequentiediagram bi-directionele communicatie

Er is geen onderzoek uitgevoerd naar de meest geschikte library om dit te realiseren omdat de gebruikte library (libxbee) al gekozen was bij de aanvang van de opdracht.

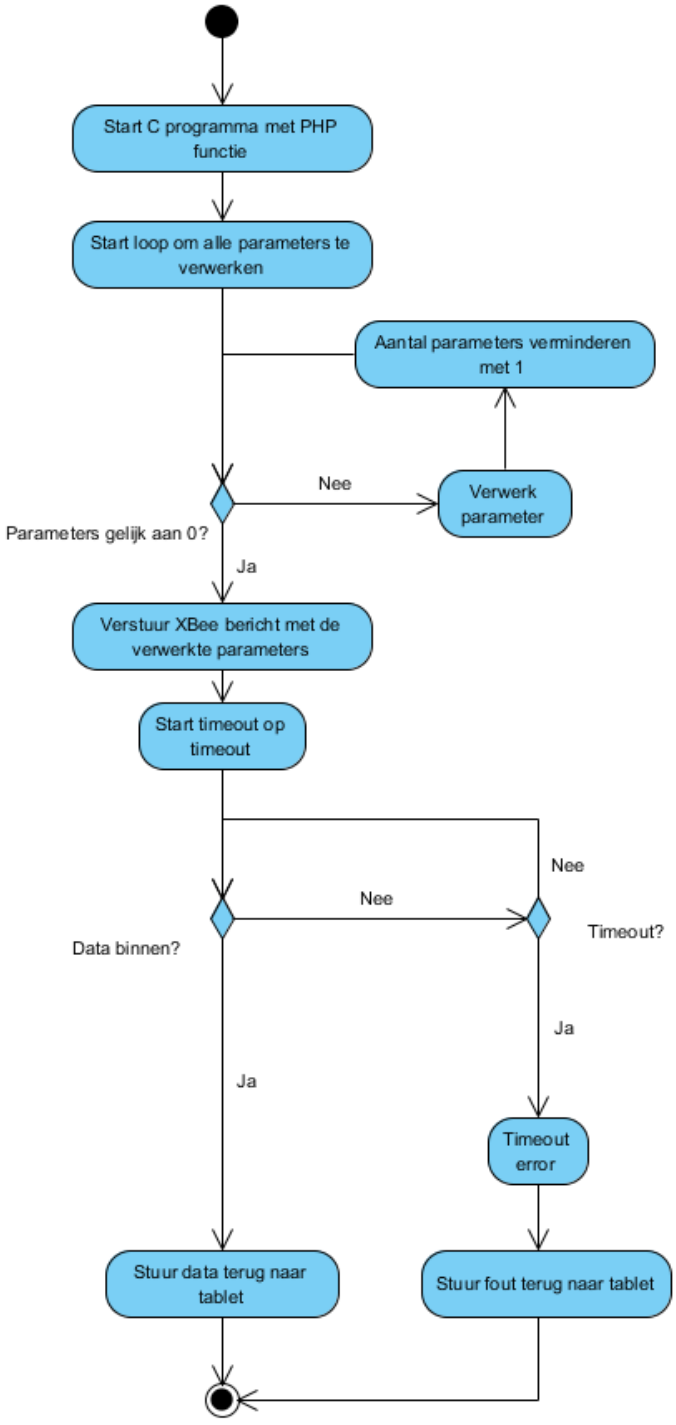
3.2.2. Web server

Naast het toevoegen van bi-directionele communicatie is ook een herontwerp van de Raspberry Pi gemaakt omdat een groot deel van de functies herschreven zijn. De reden hiervoor is omdat de bestaande functies erg inflexibel zijn en alleen konden omgaan met het moduletype LED.



Figuur 17: Klassendiagram Raspberry Pi

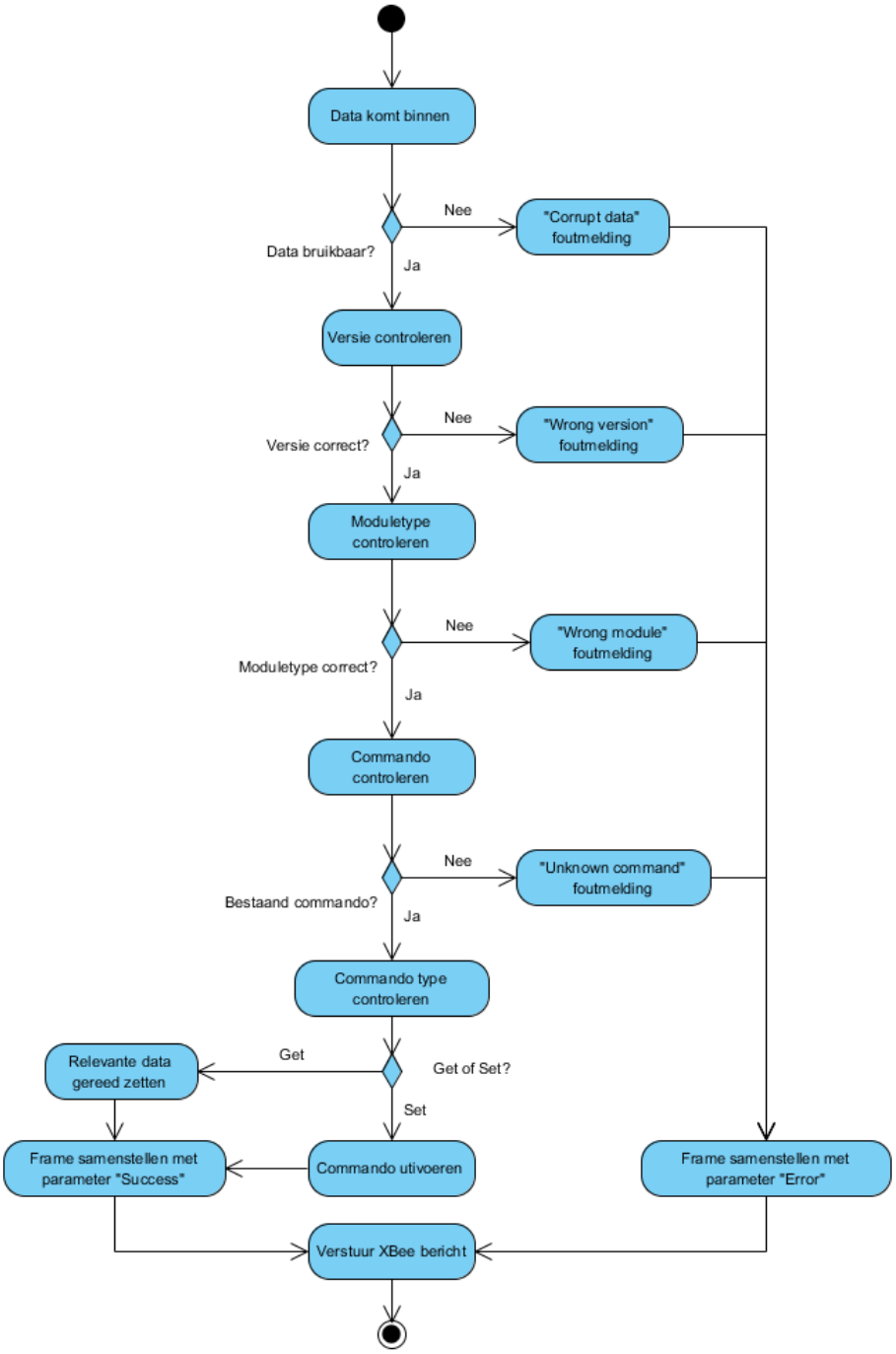
Figuur 17 geeft het klassendiagram van het herontwerp weer, hierbij zijn alle functies die te maken hebben met een module (in processData.c) opgesplitst in aparte C bestanden. Linkedlist.c is volledig herschreven, de modules worden nu opgeslagen in XML formaat. De reden hiervoor is dat de voorgaande methode de data niet valideert, het gebruikte vaste posities in een tekstbestand om de data weg te schrijven of uit te lezen. Libraries die gebruikt konden worden om een XML bestand uit te lezen in C waren “expat” en “libxml2”. De keuze was uiteindelijk op “libxml2” gevallen omdat deze library een XML Tree opbouwt in het geheugen en het hierdoor makkelijker maakt voor de programmeur om te navigeren. Bij “expat” zal de positie en diepte van de XML Tree door de gebruiker zelf bijgehouden moeten worden. In XBeeModule.c is de functie voor het versturen van berichten herschreven (transmitData), omdat de voorgaande functie alleen om kon gaan met de moduletype LED. Figuur 18 (pagina 25) geeft de nieuwe activiteitsdiagram van de Raspberry Pi weer, het C programma zal pas afsluiten wanneer er een “Timeout” optreed of wanneer er een antwoord terugkomt van de module.



Figuur 18: Activiteitsdiagram Raspberry Pi met bi-directionele communicatie

3.2.3. Module

De modules bestaan uit ten minste een Arduino en een XBee, hiernaast hangen nog componenten aan vast die nodig zijn voor de module. De functies die uitgevoerd moeten worden door de Arduino zijn per module verschillend, maar de manier van data verwerken is voor elke module hetzelfde. In figuur 19 staat de activiteitsdiagram voor het verwerken van data op de Arduino. Om dit te realiseren moet een functie geschreven worden (op de Arduino) dat XBee berichten kan sturen (naar de Raspberry Pi).



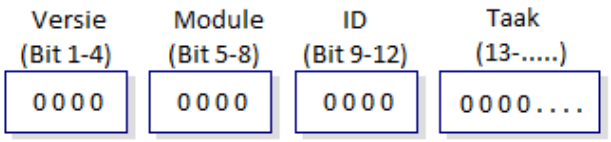
Figuur 19: Activiteitsdiagram Arduino met bi-directionele communicatie

4. Realisatie van het systeem

In dit hoofdstuk wordt de realisatie van de ontwerpen gemaakt in “3. Ontwerp voor het nieuwe systeem” en de problemen die hierbij opgetreden zijn besproken.

4.1. Protocol en bi-directionele communicatie

Voordat bi-directionele communicatie gerealiseerd werd moest eerst het protocol uitgebreid worden. Het verstuurframe wordt opgebouwd binnen een XBee API frame (Bijlage “XBee API frame”) in het gedeelte van RF Data.



Figuur 20: Verstuurframe

Figuur 20 geeft het verstuurframe weer. Het frame wordt samengesteld met behulp van de gebruikte library (libxbee). Dit frame was opgesteld door de vorige afstudeerder.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
State	CMD	Data	Data	Data	Data	Data	Data

Figuur 21: Antwoordframe

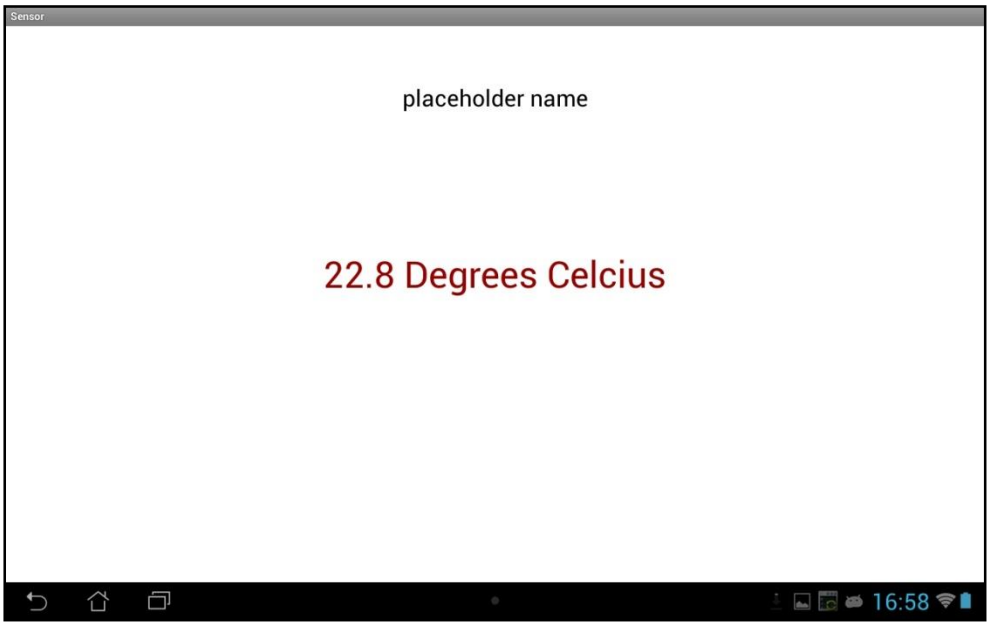
Figuur 21 geeft het antwoordframe weer, dit verschilt met het verstuurframe in de mate dat het gebruik van hele bytes (8 bits) in plaats van halve bytes (4 bits). Dit is gedaan omdat de formaat van het verstuurframe alleen 1 byte aan extra ruimte zal opleveren en veel meer code voor het verwerken van de data. Hieronder volgt nog een korte uitleg van het frame:

- **Byte 1**
  Geeft aan of het commando succesvol uitgevoerd is.
- **Byte 2**
  Geeft het ontvangen commando weer.
- **Byte 3-8**
  De data die teruggestuurd wordt. Als het commando niet succesvol uitgevoerd is dan geeft Byte 3 aan welke fout opgetreden is in plaats van de Data.

Ten slotte is het protocol uitgebreid met een commandolijst en een foutenlijst (Bijlage “Protocol”). Nadat het protocol uitgebreid is kon bi-directionele communicatie gerealiseerd worden. Dit is gedaan door een “Callback” te implementeren in het programma, dit betekent dat het programma niet meer meteen beëindigd wordt na het versturen van een Xbee bericht en zal wachten op een antwoord tot een tijdslimiet bepaald door de programmeur. Tijdens het implementeren was het probleem voortgekomen dat de Raspberry Pi geen berichten ontving van de modules, het probleem hierbij was dat de XBee module van de Arduino de berichten terugstuurden in 16-bit formaat terwijl “libxbee” alleen werkt met 64–bit adressen. Dit is opgelost door de XBee (op de modules) te initialiseren in 64-bit mode.

4.2. Temperatuursensor

Een van de hoofdeisen van de opdracht is het toevoegen van een nieuwe module, hierbij is de temperatuursensor als eerste toegevoegd. Bij het ontwerp is er gekozen om een digitale temperatuursensor te gebruiken met de belangrijkste redenen als volgt: minder componenten en meer uitbreidingsmogelijkheden. De gekozen sensor (DS18B20) gebruikt namelijk de “1-Wire bus”, hierbij is maar 1 datalijn nodig om de sensor te besturen en uit te lezen. Aan dezelfde datalijn kan ook meerdere componenten bestuurd en uitgelezen worden, indien ze ook gebruikmaken van de “1-Wire bus” [Bron: Bronnenlijst 7.]. De uitgelezen temperatuur is een float waarde en zal omgezet moeten worden zodat het in het antwoordframe past.

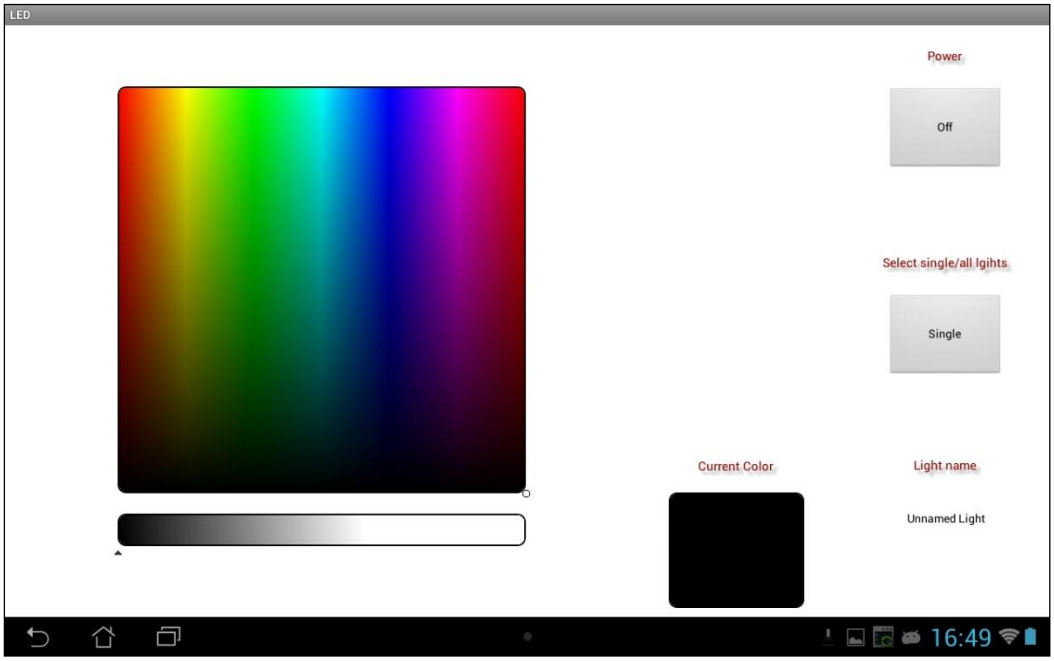


Figuur 22: Scherm voor temperatuursensor

Figuur 22 geeft het scherm voor de temperatuursensor op de applicatie weer, voordat het scherm geopend wordt is er de mogelijkheid om ervoor te kiezen de temperatuur in Fahrenheit te vertonen in plaats van Celsius. De applicatie vernieuwt de temperatuurwaarde om de 5 seconden zolang het scherm openblijft. De naam van de temperatuursensor kan veranderd worden door op “placeholder name” te klikken, dit is nog niet geïmplementeerd in de Backend.

4.3. LED

Nadat de temperatuursensor geïmplementeerd was is het nieuwe GUI ontwerp van de LED module ook gerealiseerd. Hierbij was het de bedoeling om bi-directionele communicatie toe te passen bij de LED module. De applicatie zal bij het openen van het scherm de huidige kleur en staat van de LED module opvragen en geeft hierdoor de actuele situatie weer.



Figuur 23: Scherm voor LED

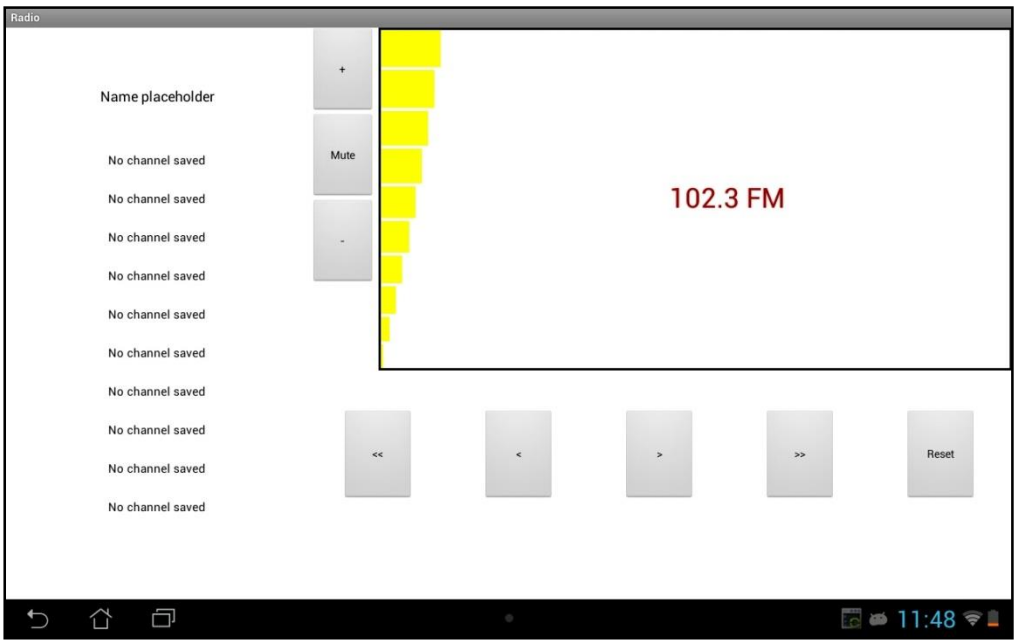
Figuur 23 geeft het nieuwe GUI voor de LED weer. In het nieuwe ontwerp is de mogelijkheid om een RGB handmatig in te voeren en de sleepbalk om intensiteit te veranderen verwijderd, deze elementen zijn nu overbodig geworden. Het kleurenveld links genereert een HSL waarde die dan geconverteerd wordt naar een RGB waarde. Op het veld zelf wordt de H (Hue) en de S (Saturation) waardes bepaald en de onderliggende balk bepaalt de L (Lightness) waarde, dit is ook de reden waarom de sleepbalk voor de intensiteit overbodig werd. Met de knop “Power” kan de LED module aan of uit worden gezet en met “Light name” kan de naam de module worden veranderd (nog niet geïmplementeerd in Backend). Het knop voor “Select single/all lights” kan een enkel of alle modules LED tegelijk aangestuurd worden.

4.4. Radio

Na het toevoegen van de temperatuursensor en het uitbreiden van de applicatie waren de hoofdeisen van de opdracht vervuld, hierna waren de volgende opties beschikbaar:

- Onderzoek voor (netwerk)beveiligingsmogelijkheden voor de Raspberry Pi
- Toevoegen van een nieuw module

De keuze was hierbij gevallen op het toevoegen van een nieuwe module door voorkeur van de opdrachtgever. Voor de Radio module is het Si4703 FM Tuner Breakout Board gebruikt die verbonden is met een Klasse-D audio versterker, een 1.45 meter draad gebruikt als een antenne voor de Si4703 FM Tuner[8]. Tijdens de realisatie is het probleem voortgekomen dat de radio geen zender kon afspelen (alleen ruis). De probleemoorzaak lag bij de gekozen audio versterker, die zorgde voor interferentie bij de ontvangst van de antenne. Het is tijdelijk opgelost door een computer speaker erop aan te sluiten, een permanente oplossing is om de audio versterker te isoleren. Een ander probleem is het ontvangen van RDS data, de verkregen data is altijd maar 2 karakters groot met een frequentie van ongeveer 5 Hertz. Dit is te snel voor de XBee module om het constant door te sturen naar de Raspberry Pi. Hierover moet over nagedacht worden voor een oplossing. De bovenstaande problemen zijn helaas niet opgelost binnen de opdrachtperiode.



Figuur 24: Scherm voor radio

In figuur 24 is het scherm van de radio weergeven. Het huidige kanaal wordt hierop vertoond en kan 0.1 MHz verhoogd of verlaagd worden met de “<” en “>” knoppen. Het is ook mogelijk om de module automatisch naar een zender te laten zoeken met de “<<” en “>>” knoppen, hierbij is een filter ingesteld om het aantal “nep” zenders te verminderen[Bron: Bronnenlijst 9.]. Met de “+”, “-” en “Mute” knoppen kan het volume geregeld worden. Met de “Reset” knop kan de module opnieuw gedigitaliseerd worden en ten slotte kan een kanaal opgeslagen worden in de lijst links van het scherm.

4.5. Voeding

Een van de gestelde eisen van de opdracht is het vervangen van de voeding. Bij de aanvang van de opdracht is een moederbord met een computervoeding aanwezig. De enige taak van de moederbord hierbij is het aanzetten van de voeding, dit is inefficiënt en neemt veel ruimte in beslag. Om de voeding te vervangen is een lijst opgesteld van de totale stroomgebruik en de aanwezige spanningsniveaus, dit is te vinden in de bijlage ”Nieuwe modules” onder de hoofdstuk “Voeding”. Het voedingsblok dat hierbij gekozen is kan 12 Volt en 5 Ampère leveren. De reden dat een 12 Volt voeding gekozen is omdat de stappenmotor voor de module Rolluik (nog niet geïmplementeerd) dit nodig heeft.

Om verschillende spanningsniveaus te creëren werd een keuze gemaakt tussen een lineaire of een schakel regulator. De keuze is gevallen op een lineaire regulator met de volgende redenen[Bron: bronnenlijst 10.]:

- Geen hogere voltage dan 12 Volt nodig
- De voedingsblok is aangesloten op het lichtnet, dus efficiëntie is niet kritiek
- Lineaire regulator is goedkoper dan een schakel regulator
- Lineaire regulator vangt een storing op de uitgang en ingang sneller op

5. Conclusie en aanbevelingen

In dit hoofdstuk worden de behaalde resultaten, conclusie en aanbevelingen besproken.

5.1. Resultaten

De meeste van de gestelde eisen van de opdracht zijn behaald, hierbij zijn alle “Must” eisen behaald. Hiernaast zijn ook de aanbevelingen van de vorige afstudeerder gedeeltelijk geïmplementeerd. De aanbevelingen zijn als volgt:

- Modulelijst
  - Inactieve modules worden gezien door de applicatie
  - Modulelijst kan niet worden vernieuwd
- Foutmeldingen
- Beveiliging voor het systeem

5.1.1. Aanbevelingen en behaalde eisen

Hieronder volgt een lijst van alle behaalde eisen en aanbevelingen:

Aanbevelingen

- Modulelijst
  - Inactieve modules blijven in lijst en wordt niet door de applicatie gezien
- Foutmelding
  - Foutenlijst opgesteld
  - Foutmeldingen geïntegreerd in het protocol
  - Nog geen foutenafhandeling aanwezig

Must

- Uitwerken user interface op de smartphone/tablet
  - GUI ontworpen en geïmplementeerd (Radio, LED, Rolluik en Temperatuursensor)
- Documentatie opleveren voor verdere ontwikkeling
  - Ontwerpdocumenten (nieuwe modules, protocol en bi-directionele communicatie)
- Toevoegen van een nieuwe module
  - Temperatuursensor toegevoegd
- Gebruikershandleiding voor de user interface
- Hardware aansturen met Xbee en Raspberry Pi
- Voeding vervangen
- Werkende demonstratie opstelling (aansturen van modules vanuit de applicatie)

Could

- Meerdere modules toevoegen indien er tijd over is
  - Radio toegevoegd

5.1.2. Niet behaalde eisen

Een niet behaalde eis is het onderzoek naar een beveiligingssysteem, de reden hiervoor is omdat de voorkeur van de opdrachtgever lag bij het toevoegen van een nieuwe module (Radio).

5.2. Conclusie

De “Must” eisen van de opdracht zijn behaald, de belangrijkste onderdelen hiervan waren het toevoegen van een nieuwe module, het aanpassen van de applicatie en het leveren van een werkende demonstratie opstelling. Onderzoek naar een beveiligingssysteem is niet uitgevoerd, maar dit is niet een essentieel onderdeel en het systeem kan ook functioneren zonder dit gedeelte. Dus mijn conclusie is:

Het huidige systeem voldoet aan de meeste van de gestelde eisen en kan dienen als een demonstratie opstelling voor beurzen en open dagen.

5.3. Aanbevelingen

Al eerder in de conclusie vernoemd is dat er geen beveiliging aanwezig is voor het systeem en hierdoor kwetsbaar is voor externe invloeden. Het systeem kan gezien worden als een web server en iedereen kan dan de functies aanroepen wanneer er geen beveiliging geïmplementeerd is.

De huidige tablet applicatie is ontwikkeld in Titanium Studio met Javascript als programmeertaal. Hier was destijds voor gekozen omdat dezelfde code gebruikt kan worden voor zowel iOS als Android. Doordat iOS niet gebruikt wordt in de opdracht zou het een optie zijn om de code te herschrijven in een andere ontwikkelomgeving (bijvoorbeeld Android Studio). Een groot nadeel bij Titanium Studio is dat er geen grafische weergave is voor GUI ontwerpen, dit gebeurt geheel in code. Hierdoor wordt de code al snel lang en onoverzichtelijk.

Foutenafhandeling is ook nog niet aanwezig in het systeem. Foutmeldingen zijn wel geïntegreerd in het protocol, maar er worden door het systeem nog geen acties uitgevoerd na het optreden van een fout.

Ten slotte is er nog een aantal bugs in het systeem, een van de bugs is een enkele module/alle modules LED tegelijk besturen leverde een probleem op met intensiteit. Dit is nu opgelost door HSL kleuren te gebruiken, maar omdat de modules niet meer op de applicatie opgeslagen wordt, moet dit bijgehouden worden op de Raspberry Pi.

6. Literatuurlijst

6.1. Bronnen figuren

Figuur 1:https://intranet.ict.nl/C125656A004662CF/documents/NT0000A622?Open&portal=1
Figuur 4:http://www.certiconglobal.com/sites/default/files/images/imce/scrum.jpg

6.2. Bronnenlijst

1. http://ict.eu/nl/over-ons/
2. http://ict.eu/nl/verticals/machine-systems/omschrijving/
3. http://www.scrumalliance.org/why-scrum/
4. https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf#zoom=100
5. http://www.scrumalliance.org/why-scrum/core-scrum-values-roles
6. http://www.appcelerator.com/titanium/
7. http://www.maximintegrated.com/products/1-wire/
8. http://read.pudn.com/downloads141/sourcecode/embed/612419/AN231.pdf
   SI4700/01 HEADPHONE AND ANTENNA INTERFACE, patina 3
9. http://dlnmh9ip6v2uc.cloudfront.net/datasheets/BreakoutBoards/AN230.pdf
   Si4700/01/02/03 PROGRAMMING GUIDE, figuur 19, pagina 39

Bijlage – XBee API frame

Bijlage – Nieuwe modules

Bijlage – Protocol