
Data wegschrijven naar klimaatcomputers

Afstudeerverslag

Haagse Hogeschool
Faculteit: ICT & Design
Opleiding: Technische Informatica
Rotterdamseweg 137
2628 AL Delft

Student:
Eray Soy - 12062154

Begeleidend Examiner:
Sjaak van Peski

Tweede Examiner:
Hans de Vreught

Blok: Afstudeerperiode 2016-1.1
Datum: 3 juni 2016
Plaats: Vlaardingen
Versie: 1.0

Referaat

Eray Soy, “Data wegschrijven naar de klimaatcomputers”. Afstudeerverslag afstuderen opleiding Technische Informatica, Haagse Hogeschool, 2016.

Samenvatting

LetsGrow.com levert op dit moment online diensten voor tuinbouwondernemers, adviseurs, onderzoekers en andere belanghebbenden in de tuinbouwbranche. Al langer willen ze een systeem hebben, waarbij ze door middel van hun eigen Web API data kunnen uitwisselen met verschillende klimaatcomputers. Het project dat LetsGrow.com in gedachten had, was het ontwerpen van een prototype waarmee vanuit LetsGrow.com bestaande regelingen in klimaatcomputers op universele manier beïnvloed kunnen worden.

Dankzij het ontworpen prototype is LetsGrow.com in staat om gegevens te kunnen beïnvloeden van de uiteenlopende klimaatcomputers ongeacht het merk. Na installatie door een servicemonteur kan het prototype zonder tussenkomst van personen functioneren.

Het prototype is bruikbaar om bij klanten ingezet te kunnen worden voor verschillende toepassingen.

Dankzij de afstudeerder Eray Soy is de gehele opdracht met succes afgerond. Het ontworpen prototype laat zien hoe data van de LetsGrow servers opgehaald kan worden en kan worden doorgestuurd naar de klimaatcomputers.

Descriptoren

Klimaatcomputers, Regelingen Klimaatcomputers,

Voorwoord

Het afstudeerverslag is geschreven in het kader van mijn afstuderen aan de opleiding Technische Informatica aan de Haagse Hogeschool in Delft.

Het uiteindelijke resultaat zou ik niet bereikt hebben zonder de steun en actieve bijdrage van een aantal belangrijke personen. Hierbij wil ik de volgende belangrijke personen bedanken.

- Dhr. Peter Hendriks voor het ondersteunen tijdens de gehele stageperiode.
- Dhr. Christiaan Posthumus Meyes voor het ondersteunen tijdens de gehele stageperiode.
- Dhr. Edwin Vis voor het begeleiden van het gehele project
- Dhr. Leon Batta voor de ondersteuning met het project.
- Dhr. Hans de Vreught voor het geven van feedback tijdens het tussentijds assessment.
- Dhr. Sjaak van Peski voor het begeleiden en actieve ondersteuning van de gehele stageperiode.
- Dhr. Dave Stikkolorum voor het geven van feedback op de architectuur.
- Het LetsGrow.com team voor het zorgen voor een leerzame werkomgeving.

Eray Soy

3 juni 2016

Inhoudsopgave

1	Inleiding	1
2	Organisatie	2
2.1	Plaats van afstudeerder in organisatie.	3
3	Opdrachtomschrijving	4
3.1	Aanleiding	4
3.2	Probleemstelling	4
3.3	Doelstelling	4
3.4	Resultaat	4
4	Aanpak	5
4.1	Verantwoording aanpak	5
4.2	Planning	6
5	Oriëntatiefase	8
5.1	Opdracht verhelderen	8
5.2	Afbakening	8
5.3	Risicoanalyse	9
5.4	Afspraken met de opdrachtgever en bedrijfsmentoren	11
5.5	Kostenaspect	11
6	Definitiefase	12
6.1	Inlezen benodigde literatuur	12
6.2	Analyse huidige situatie	12
6.3	Analyse hardware, besturingssystemen en ontwikkeltools	14
6.4	Conclusie analyse hardware, besturingssystemen en ontwikkeltools	19
6.5	De wens van de opdrachtgever	20
6.6	De requirements	21
6.6.1	Functionele eisen:	21
6.6.2	Niet-functionele eisen:	21
6.7	De keuze voor het besturingssysteem, ontwikkeltool en applicatie type	22
6.8	Vervolg analyse shields	25
6.9	De keuze voor het shield	29
6.10	Architectuur nieuwe situatie	30
6.11	UML diagrammen en beschrijvingen	31
7	Ontwikkelfase	38
7.1	Ontwerp architectuur	38
7.2	Prototype realisatie	43
7.3	Testen van het prototype	45
7.4	Handleiding prototype	45
8	Evaluatie	47
8.1	Producten	47
8.1.1	Plan van Aanpak	47
8.1.2	Definitierapport	47
8.1.3	Ontwerprapport	47
8.1.4	Testplan	47
8.1.5	Handleiding	47
8.1.6	Prototype	48

8.2	Proces	48
8.2.1	Oriëntatiefase:	48
8.2.2	Definitiefase:	48
8.2.3	Ontwikkelfase:	49
8.3	Competenties	49
8.4	Kwaliteitsniveau project	50
8.5	Doel en Probleem	50
Referenties		51
Bijlagen		53
A	Klassendiagram Cliënt applicatie deel 1	54
B	Klassendiagram Cliënt applicatie deel 2	55
C	Klassendiagram Cliënt applicatie deel 3	56
D	Ontwerp Klassendiagram Cliënt applicatie deel 1	57
E	Ontwerp Klassendiagram Cliënt applicatie deel 2	58
F	Ontwerp Klassendiagram Cliënt applicatie deel 3	59
G	Klassendiagram LetsGrow.com Web API server applicatie	60
H	Ontwerp Klassendiagram LetsGrow.com Web API server applicatie	61
I	Plan Van Aanpak	62
J	Definitierapport	77
K	Ontwerprapport	154
L	Testrapport	173
M	Gesprek Opdrachtgever	185
N	Gesprek Christiaan Posthumus Meyes	188
O	Gesprek Edwin Vis	191
P	Interview Jack Schipper	193
Q	Gesprek Bedrijfsmentoren	195
R	Handleiding Prototype	198

1 Inleiding

Voor het afronden van mijn studie, moest ik bij een bedrijf gaan afstuderen. Ik liep bij LetsGrow.com stage en als oproepkracht was ik ook bij LetsGrow.com werkzaam. Ze hadden een hele interessante opdracht liggen, waarbij er een prototype ontworpen moest worden om de regelingen in verschillende klimaatcomputers te beïnvloeden. Deze opdracht zou met een Raspberry Pi 2 en een shield gerealiseerd worden. Tegenwoordig wordt de Raspberry Pi voor verschillende doeleinden gebruikt. Tevens was ik ook zelf benieuwd of het door middel van een Raspberry Pi en een shield een klimaatcomputer beïnvloed kon worden, want grotendeels wordt Raspberry Pi 2 gebruikt voor hobby-projecten. Het was interessant om het in een bedrijf te kunnen gebruiken als een oplossing. Dit sprak mij heel erg aan en dus heb ik deze opdracht met veel plezier aangenomen.

2 Organisatie

In dit hoofdstuk wordt het stagebedrijf beschreven en de plaats van de afstudeerder binnen het bedrijf.

LetsGrow.com

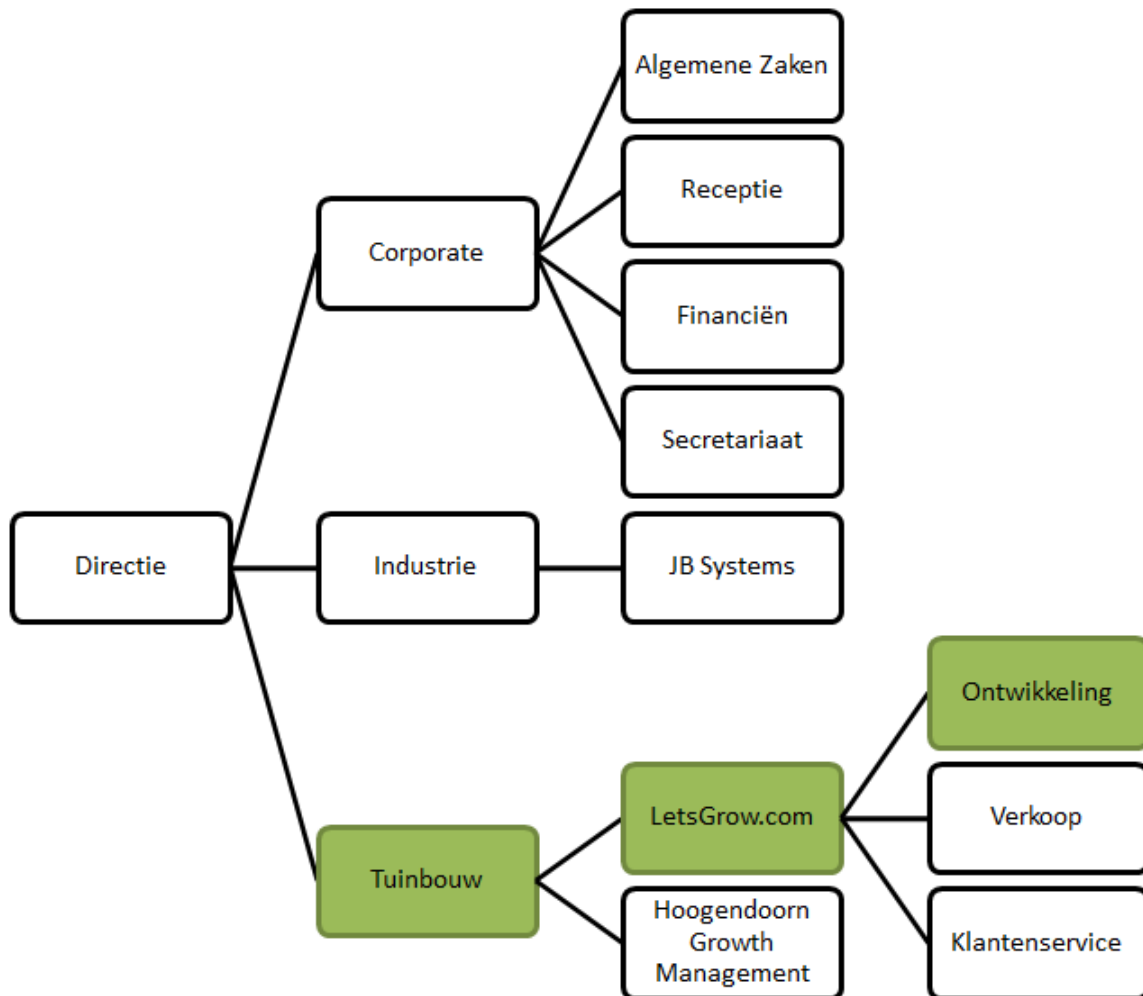
LetsGrow.com (figuur 1) is een van de dochterondernemingen van Hoogendoorn Automatisering B.V. De Hoogendoorn B.V is een zelfstandige werkmaatschappij van Batenburg Techniek. Batenburg Techniek is een technische dienstverlener op het gebied van installatietechniek en technische handel. LetsGrow.com maakt deel uit van de Business Unit van Hoogendoorn B.V.[Hoogendoorn, 2016]. De missie van LetsGrow.com is om tuinbouwondernemers te voorzien van waardevolle informatie ten behoeve van het optimaliseren van de bedrijfsvoering [LetsGrow.com, 2015]. Hierbij streeft LetsGrow.com om maximale kwaliteit en een sterke service aan te bieden met een goede prijs/prestatie-verhouding. LetsGrow.com biedt de mogelijkheid teeltgerelateerde gegevens te registreren en te analyseren. Via het online platform zijn de gegevens van onder andere het kasklimaat, het gewas, arbeid en het energieverbruik in te zien. Tevens wil LetsGrow.com zich in de toekomst op andere branches buiten de tuinbouw richten.



Figuur 1: Logo LetsGrow.com

LetsGrow.com streeft ernaar om wereldwijd de online service te zijn voor tuinbouwondernemers, adviseurs, onderzoekers en andere belanghebbenden in de tuinbouwbranche[LetsGrow.com, 2015]. Dit is tevens hun visie.

Hieronder is het organogram (figuur 2) opgesteld van LetsGrow.com. Zoals in het organogram te zien is, valt LetsGrow.com onder Tuinbouw.



Figuur 2: Organogram LetsGrow.com

2.1 Plaats van afstudeerder in organisatie.

Binnen LetsGrow.com heb je een programmeerteam dat uit twee ervaren programmeurs bestaat. Voor dit project wilden ze extra mankracht (afstudeerder) hebben om het prototype te kunnen realiseren. Door het aantrekken van een extra kracht, konden de LetsGrow programmeurs zich blijven richten op hun reguliere taken. Het gehele LetsGrow.com team bevond zich op dezelfde afdeling als de afstudeerder waardoor de communicatie met andere medewerkers werd vergemakkelijkt.

3 Opdrachtschrijving

In dit hoofdstuk wordt de opdrachtschrijving behandeld. Deze opdrachtschrijving is tevens te vinden in het afstudeerplan.

3.1 Aanleiding

In de toekomst wil LetsGrow.com data door middel van een minicomputer direct door kunnen sturen naar de klimaatcomputers van klanten onafhankelijk van het type of merk. Hiermee willen ze zelf de regelingen in de klimaatcomputers kunnen beïnvloeden.

3.2 Probleemstelling

LetsGrow heeft momenteel niet de mogelijkheid om de regelingen in de klimaatcomputers van derden (Hortimax, SerCom en Priva) te kunnen beïnvloeden zonder de tussenkomst van de klimaatcomputer leveranciers.

3.3 Doelstelling

Een werkende oplossing (prototype) bouwen voor LetsGrow.com, zodat met behulp van dit prototype regelingen in de klimaatcomputers kunnen worden beïnvloed, zonder de tussenkomst van de klimaatcomputer leveranciers.

3.4 Resultaat

Een prototype waarmee door LetsGrow.com op afstand regelingen kunnen worden beïnvloed zonder de tussenkomst van de klimaatcomputer leveranciers.

4 Aanpak

Voor het afstuderen had de afstudeerder al bepaald om het gehele project in verschillende fasen te uitvoeren. Deze fasering is de gehele afstudeerperiode ook toegepast.

De fasering bestond uit:

- Oriëntatiefase
- Definitiefase
- Ontwikkelfase

Oriëntatiefase stond in de teken van het opstellen van Plan van Aanpak, verhelderen van de opdracht, opstellen van de risicoanalyse en planning maken voor de gehele periode.

Definitiefase stond in de teken van de literatuurstudie, analyseren van hardware, besturings-systemen en ontwikkeltools, achterhalen van de eisen en de wensen, keuze maken over het besturingssysteem, ontwikkeltool en hardware componenten en uiteindelijk opstellen van de systeemarchitectuur.

Ontwikkelfase stond in de teken van ontwerp van het prototype, realiseren van het prototype, testen van het prototype en afronden van de gehele afstudeerperiode.

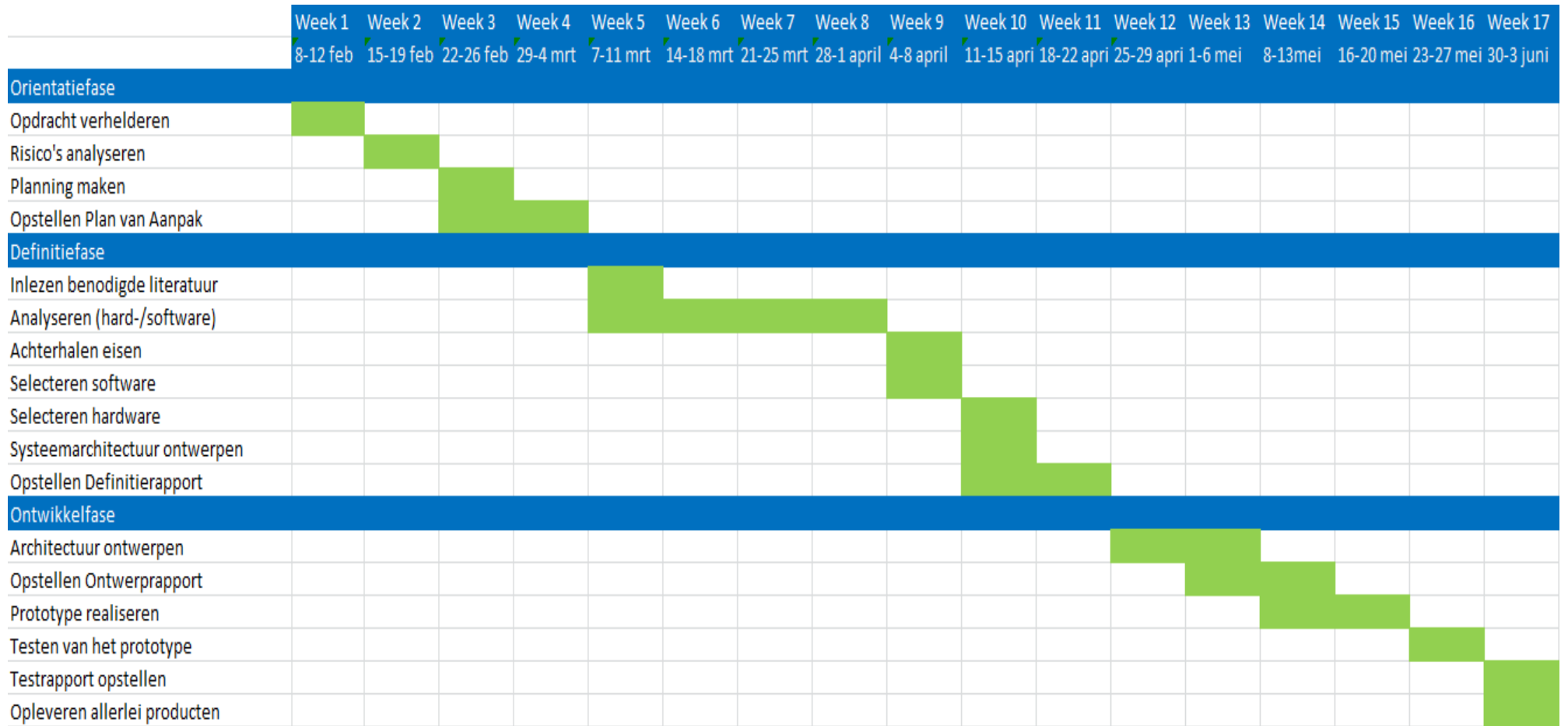
4.1 Verantwoording aanpak

Het gehele afstudeerproject is gerealiseerd aan de hand van de waterval methode waarbij er gebruikt werd gemaakt van 3 fasen. Er is voor deze fasering gekozen, omdat dankzij deze fasering het gehele afstudeerproces overzichtelijker werd. Tevens was het dankzij deze aanpak het gehele project beter beheersbaar. Alles werd stap voor stap gecontroleerd gerealiseerd en elk onderdeel werd goed gedocumenteerd. Daarnaast kon er van te voren alles goed ingepland worden. Deze aanpak maakte het mogelijk om van te voren alle beschikbare tijd goed in te kunnen plannen, zodat er voor elk onderdeel van het project voldoende tijd besteed kon worden.

Daarnaast is er ook de 'Agile'[van Houtum, 2013] methodiek, waar LetsGrow.com zelf gebruik van maakt. Deze methodiek wordt meestal in een 'kleine team-verband' gebruikt. Deze methodiek kon ook door de afstudeerder gebruikt worden, alleen is de afstudeerder de 'enige' persoon om het gehele project uit te voeren. Daarnaast had de opdrachtgever een behoefte en deze moest eerst geanalyseerd worden. Er moest dus gedocumenteerd worden. Tevens was de doel van te voren bekend en de kans op wijzigingen heel klein. Hierdoor is er gekozen om de faseringen aan te houden en niet een iteratieve methodiek te gebruiken. De afstudeerder heeft wel met de 'stand-ups' meegedaan, alleen voor het project zelf is er niet gebruik gemaakt van de 'Agile' methodiek.

4.2 Planning

De planning was al van te voren in het afstudeerplan gerealiseerd. Deze planning is de gehele afstudeerperiode ook toegepast. In het afstudeerplan was per fase een globale week periode gehouden. Dit hield in dat er bijvoorbeeld voor de eerste fase vier weken besteed kon worden. Dit was natuurlijk globaal en moest in detail ingepland worden. Dit is dan ook per week gedaan. De planning is op de volgende bladzijde in figuur 3 te vinden.



Figuur 3: Pva: Planning

5 Oriëntatiefase

In dit hoofdstuk en de komende hoofdstukken zullen de werkzaamheden die de afstudeerder de gehele afstudeerperiode verricht heeft, beschreven worden.

De oriëntatiefase was vooral bedoeld om het plan van aanpak voor het gehele project samen te stellen, opdracht te verhelderen, risico analyse opstellen en een planning maken voor het gehele project. Naast deze werkzaamheden zijn ook andere zaken zoals de afspraken en de rolverdelingen met de opdrachtgever, bedrijfsmentoren vastgesteld. Daarnaast is er ook een project afbakening opgesteld.

5.1 Opdracht verhelderen

Om opdracht te verhelderen, moest eerst het probleem en de context duidelijker uitgewerkt worden. Dit om de behoefte van de opdrachtgever goed helder te krijgen. Dit is gedaan aan de hand van de 6-W formule[Verhoeven, 2010].

Om al deze zaken te weten komen, is er gesprek gehouden met de opdrachtgever [Soy, 2016a]. Aan de hand van dit gesprek is de volgende probleemanalyse naar voren gekomen.

LetsGrow heeft momenteel niet de mogelijkheid om de regelingen in de klimaatcomputers van derden (Hortimax, SerCom en Priva) te kunnen beïnvloeden zonder de tussenkomst van de klimaatcomputer leveranciers. Ze willen in de toekomst veel minder afhankelijk zijn van de klimaatcomputer leveranciers en het kunnen beïnvloeden van regelingen als een aanvullende dienst gaan aanbieden naast de mogelijkheden die LetsGrow.com al heeft. Klanten krijgen de mogelijkheid om via het prototype te kunnen regelen op extra parameters die extern kunnen worden berekend. Hierdoor zullen de klanten het LetsGrow.com platform intensiever gebruiken en zullen klanten hun abonnement eerder verlengen/langer aanhouden. LetsGrow.com wil in de toekomst de klanten de mogelijkheid bieden om de klimaatcomputers regelingen te kunnen beïnvloeden. Op dit moment kunnen ze dit niet realiseren, omdat hiervoor geen functionaliteit beschikbaar is. De opdrachtgever heeft gevraagd om te kijken of het mogelijk is om door middel van een oplossing (Raspberry Pi 2 met een bijbehorende shield) de regelingen in de klimaatcomputers te kunnen beïnvloeden. Het prototype moet ervoor zorgen dat er waarden bij LetsGrow worden opgevraagd en kunnen worden doorgestuurd naar de klimaatcomputers.

5.2 Afbakening

De projectafbakening was belangrijk om te weten komen wat buiten de scope van het project viel. Dit was benodigd, omdat de afstudeerperiode 17 weken in beslag nam. Hierdoor kon niet alles onderzocht of geanalyseerd worden. In week 1 is er met de opdrachtgever de afbakening vastgesteld. Tijdens het opstellen van de afbakening had de afstudeerder een vraagteken bij het punt:

- Het prototype wordt getest op een Hoogendoorn Klimaatcomputer.

De afstudeerder vroeg zich af of het niet een groot risico was om een prototype te ontwikkelen voor merk onafhankelijke klimaatcomputers, terwijl het prototype niet op andere klimaatcomputers getest kon worden. De opdrachtgever en de afstudeerder zijn met een Priva klimaatcomputer dealer in contact gekomen en deze Priva dealer heeft mondeling meegedeeld dat de werking van Priva klimaatcomputers overeenkwam met de Hoogendoorn klimaatcomputer.

Daarnaast was de 'minicomputer' voor het prototype van te voren vastgesteld. Deze zou gerealiseerd worden aan de hand van de Raspberry Pi 2. Aan de Pi 2 zou een shield gekoppeld worden. Deze twee componenten vormden de hardware basis voor het prototype. Tevens is er met de opdrachtgever besloten om voor het analyseren van besturingssystemen gebruik te maken van het Windows IoT besturingssysteem en het Raspbian besturingssysteem met het programmeertaal C#. Raspbian[Foundation, 2016] was het officiële besturingssysteem voor de Raspberry Pi 2 en Windows IoT[Microsoft, 2016b] was een van de nieuwkomers in de markt.

5.3 Risicoanalyse

Er is in deze fase ook een risicoanalyse opgesteld om zoveel mogelijke bedreigingen in kaart te brengen voor het prototype. De risico's waren voor zowel de opdrachtgever als de afstudeerder van belang om bij een risico passende maatregelen bij de hand te hebben die van te voren bedacht zijn. Op de volgende pagina zijn sommige risico's van de Plan van Aanpak afgebeeld. De gehele risicoanalyse is gebaseerd op hoofdstuk 8 op bladzijde 10 (in de bijlage op de bladzijde 75) van het Plan van Aanpak.

Omschrijving	De opdrachtgever is tijdens de gehele periode meer dan 14 werkdagen afwezig wegens ziekte/ongeval.
Risico	De opdrachtgever is tijdens de gehele periode meer dan 5 werkdagen afwezig wegens ziekte/ongeval.
Kans	Klein
Kansverlagende maatregel	Geen
Impact	Middel
Impactverlagende maatregel	Met de opdrachtgever afspreken wie als alternatieve opdrachtgever beschikbaar kan zijn, indien de opdrachtgever meer dan 5 werkdagen afwezig is.
Plan B	Met de alternatieve opdrachtgever contact opnemen.

Omschrijving	De bestelde hardware komt niet op tijd aan/duurt meer dan 5 werkdagen.
Risico	De bestelde hardware komt niet op tijd aan en/of het duurt meer dan 5 werkdagen om geleverd te worden.
Kans	Middel
Kansverlagende maatregel	De hardware bestellen bij een leverancier die het snelst kan bezorgen. Indien het kan, ophalen bij een fysieke winkel.
Impact	Groot
Impactverlagende maatregel	Kijken wanneer de bestelling ontvangen zal worden. Als het langer duurt dan normaal, gelijk een alternatieve winkel zoeken waar dezelfde hardware besteld kan worden en de lopende bestelling/pakket annuleren.
Plan B	Met de bedrijfsmentoren en opdrachtgever bespreken en zo spoedig mogelijk een alternatieve winkel benaderen voor de nieuwe bestelling.

Omschrijving	De hardware loopt schade op.
Risico	De hardware loopt schade op.
Kans	Middel
Kansverlagende maatregel	De handleiding van de hardware goed lezen voor bijvoorbeeld hoeveel voltage hij aankan. Tevens na gebruik de hardware goed opbergen in een schermhoesje en in de kast leggen, zodat anderen er niet eraan kunnen zitten. Ook is het aangeraden om niet te eten of te drinken in de buurt van de hardware.
Impact	Groot
Impactverlagende maatregel	Vervangende hardware bestellen bij de leverancier.
Plan B	De hardware zo snel mogelijk retour sturen/langs brengen naar de leverancier en zo snel mogelijk een nieuwe bestellen. De opdrachtgever en de bedrijfsmentoren hierover informeren.

Figuur 4: Pva: Risicoanalyse

5.4 Afspraken met de opdrachtgever en bedrijfsmentoren

De afstudeerder heeft met zowel de opdrachtgever als bedrijfsmentoren duidelijke afspraken gemaakt rondom de communicatie en aanwezigheid. Dit was belangrijk om van te voren met beide partijen afspraak te maken, zodat alles op papier vastlag. Er kon teruggegrepen worden bij misverstanden.

5.5 Kostenaspect

Het kostenaspect was belangrijk voor de afstudeerder, omdat hij rekening mee moest houden met het gehele prototype. De kosten waren ook een doorslaggevende factor voor de hardware keuze. De opdrachtgever had een budget van 300 euro vastgesteld voor onderzoek en ontwikkeling van het prototype. Dit budget hield in dat de afstudeerder totaal een bedrag van 300 euro uit kon geven voor het gehele prototype onderzoek. Het uiteindelijke prototype mocht ook niet meer dan 250-300 euro kosten [Soy, 2016a].

6 Definitiefase

In de definitiefase is er eerst relevante literatuur bestudeerd, de huidige situatie geanalyseerd, aan de hand van het probleemstelling een hoofdvraag met deelvragen en sub-deelvragen gedefinieerd om te analyseren. Deze analyse was benodigd om eerst te kijken hoe de regelingen in de klimaatcomputers beïnvloed konden worden, zonder de tussenkomst van de betreffende klimaatcomputer leverancier. Pas daarna zijn in deze fase de eisen en de wensen achterhaald. Aan de hand van deze eisen en de analyse, is er vervolgens een selectie gemaakt op het gebied van hardware en software. Uiteindelijk is er ook een systeemarchitectuur opgesteld.

6.1 Inlezen benodigde literatuur

Voor dit project heeft de afstudeerder verschillende soorten literatuur van te voren ingelezen om kennis op te doen over verschillende aspecten. De literatuur die de afstudeerder van te voren gebruikt had:

- The MagPi Magazine issue 31[Foundation, 2015b].
- The MagPi Magazine issue 34[Foundation, 2015c].
- The MagPi Magazine issue 35[Foundation, 2015d].
- Learning Raspbian[Harrington, 2015].
- Windows 10 for the Raspberry Pi 2[Sargent,].

Deze literatuur heeft ervoor gezorgd dat de afstudeerder zijn kennis kon verbreden. Daarnaast was het ook raadzaam om te kijken welke projecten gerealiseerd kunnen worden met de Raspberry Pi 2. Hierdoor kon de afstudeerder zijn kennis uitbreiden en indien nodig ook deze nieuwe kennis meenemen in het te realiseren project.

6.2 Analyse huidige situatie

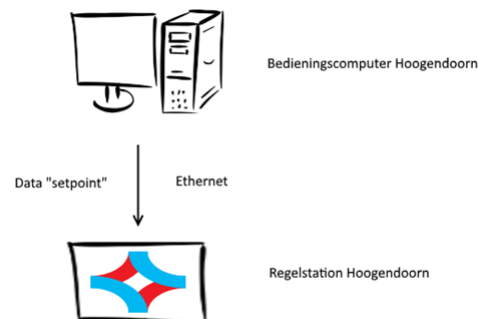
De huidige situatie is door de afstudeerder door middel van gesprekken met desbetreffende personen eerst in kaart gebracht. Dit was benodigd om eerst te kijken wat LetsGrow.com nu met de klimaatcomputers kan. Deze analyse is gebaseerd op het hoofdstuk 2 op bladzijde 2 (in de bijlage op de bladzijde 83) van het definitierapport.

LetsGrow.com zorgt door middel van internet dat er data bij de Hoogendoorn klimaatcomputers aangeleverd wordt (figuur 5).



Figuur 5: LetsGrow - Hoogendoorn klimaatcomputer communicatie

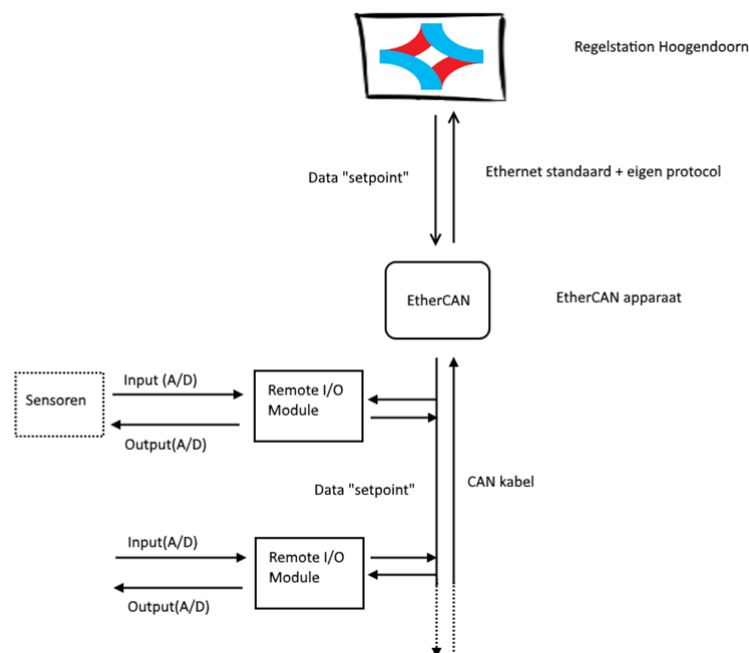
De Hoogendoorn klimaatcomputer zorgt ervoor dat de klant zijn klimaat in zijn kas (watergift, warmte-krachtkoppeling (WKK's) ed.) kan controleren en indien nodig kan sturen. De configuratie die de klant instelt, wordt doorgestuurd naar het regelstation. Dit gebeurt door de bedieningscomputer die alle data door middel van een ethernet doorstuurt naar het regelstation (figuur 6)[Soy, 2016b].



Figuur 6: Klimaatcomputer - Regelstation communicatie

De functie van het regelstation is het uitvoeren van de regelingen. Om te kunnen regelen zijn er metingen en sturingen nodig. Hiervoor zorgt het Input/Output-systeem. Het I/O-systeem bestaat uit een 'EtherCan' welke de gateway is tussen 'ethernet' en 'CAN-bus', vervolgens zijn er 'remote I/O modules' die decentraal opgesteld worden en aan de CAN-bus hangen. Aan de 'remote I/O modules' zijn de sensoren en actuatoren verbonden (figuur 7).[van Linden, 2014]

Metingen komen via het I/O-systeem bij het regelstation binnen, het regelstation bepaalt de actie aan de hand van de meting en het setpoint (sturing waarde). De actie wordt in een sturing omgezet die via het I/O-systeem naar het regelorgaan gaat. Bijvoorbeeld een motor om de ramen open te sturen[Soy, 2016b].



Figuur 7: Communicatie tussen de remote input/output modules, sensoren met het regelstation.

De 'remote input/output modules' worden op de Hoogendoorn computer zelf geconfigureerd. Dit houdt in dat wanneer een input of een output nodig is een servicemonteur de desbetreffende module aan de 'CAN-kabel' koppelt[Soy, 2016b]. Als er bijvoorbeeld een 10V spanning input nodig is, dan koppelt de servicemonteur een 'Beckhoff bus terminal' met de desbetreffende input-voltage. Dit verschilt dus per gekoppelde sensor.

Omdat de specificaties van een Hoogendoorn klimaatcomputer 100% bekend zijn bij Lets-Grow.com en andere klimaatcomputer leveranciers liever geen openheid van zaken geven kan LetsGrow.com alleen een 'setpoint' sturen naar de Hoogendoorn klimaatcomputer[Soy, 2016b].

6.3 Analyse hardware, besturingssystemen en ontwikkeltools

De afstudeerder heeft daarna een analyse uitgevoerd om te kijken hoe de regelingen in klimaatcomputers beïnvloed kunnen worden, zonder de tussenkomst van de betreffende klimaatcomputer leverancier, voordat er eisen opgesteld worden en hardware en software geselecteerd konden worden. Deze analyse is gebaseerd op het hoofdstuk 3 op bladzijde 4 (in de bijlage op de bladzijde 85) van het definitierapport.

Voor de analyse zijn de volgende onderzoeksmethodieken gebruikt:

- Analyseren van bestaande interne bronnen.
- Bronnenonderzoek
- Interviews/gesprekken met deskundig personeel over het onderwerp

Onder analyseren van bestaande interne bronnen wordt er verstaan dat de afstudeerder de interne elektronische bronnen omtrent technische specificaties over bepaalde hardware bestudeerd heeft. Deze bronnen zijn voor 'buitenstaanders' niet toegankelijk. Om deze bronnen te kunnen bereiken, moet een van de bedrijfsmentoren inloggen en de bron openen. Deze bronnen worden in een beveiligde omgeving bewaard.

Bronnenonderzoek is vooral uitgevoerd om verschillende informatiebronnen zoals het internet en boeken te raadplegen om een antwoord te vinden op de hoofdvraag. Deze bronnen zijn met nauwkeurigheid gekozen op betrouwbaarheid.

De interviews en gesprekken met deskundig personeel is vooral uitgevoerd om informatie te weten komen, indien er op papier niets gevonden kon worden over het desbetreffend ontwerp. In de vestiging van Hoogendoorn zijn medewerkers aanwezig die zich bezig houden met zaken variërend van verkoop tot hardware. Hierdoor is het makkelijk om iemand te vinden die wel informatie heeft over een bepaald onderwerp.

Om goed te kunnen analyseren, moest er een hoofdvraag opgesteld worden. De hoofdvraag is aan de hand van de probleemstelling geformuleerd.

De hoofdvraag luidt: *Hoe kunnen regelingen in klimaatcomputers beïnvloed worden zonder de tussenkomst van de betreffende klimaatcomputer leverancier?*

Uitgangspunt voor deze hoofdvraag was dat er gebruik gemaakt werd van een stukje hardware in de vorm van een Raspberry Pi 2 met een specifiek shield (prototype). Om deze hoofdvraag goed te kunnen beantwoorden, was deze hoofdvraag opgedeeld in deelvragen en deze deelvragen weer opgedeeld in verschillende sub-deelvragen. Sommige van deze deelvragen en sub-deelvragen zullen hieronder kort behandeld worden.

Wat is een klimaatcomputer

Een klimaatcomputer is een computer die binnen een tuinbouwbedrijf verantwoordelijk is voor de aansturing en controleren van allerlei processen zoals het klimaat (temperatuur, relatieve vochtigheid, CO₂, verwarming, koeling), watergift maar ook warmte-kracht-koppeling[Soy, 2016b].

De Hoogendoorn klimaatcomputer bestaat uit 2 onderdelen, de bedieningscomputer en het regelstation. De bedieningscomputer is een pc met Windows 7 besturingssysteem waarop een Hoogendoorn software draait. Het regelstation is een “Embedded mini-pc” die draait op Windows 7 Embedded besturingssysteem. De functie van het regelstation is het uitvoeren van de regelingen. De bedieningscomputer en het regelstation zijn met een ethernet kabel aan elkaar verbonden en wisselen voortdurend met elkaar gegevens uit. Op de bedieningscomputer waar de Hoogendoorn software opdraait, haalt van de servers de noodzakelijke gegevens op en stuurt die naar het regelstation. Vanuit het regelstation worden deze gegevens weer verstuurd naar de ingangen/uitgangen modules (remote input/output modules)[Soy, 2016d].

Wat is een regeling?

Een regeling is een model van berekeningen waarbij uiteenlopende zaken geregeld kunnen worden. Om te kunnen regelen zijn er metingen (meting waardes) en sturingen (sturing waardes) nodig. Hiervoor zorgt het Input/Output-systeem (Ingang/Uitgang-systeem). Het I/O-systeem bestaat uit een EtherCan welke de gateway is tussen ethernet en CAN-bus, vervolgens zijn er ‘remote Input/Output modules’ die decentraal opgesteld worden en aan de CAN-bus hangen. Aan de ‘remote I/O modules’ zijn de sensoren en actuatoren verbonden[Soy, 2016b].

Hoe kan een regeling worden beïnvloed?

Een regeling kan aan de hand van de verschillende meting waardes en sturing waardes (set-points) beïnvloed worden. Dit houdt in dat er digitale data afkomstig vanuit de klimaatcomputer en “data” afkomstig vanuit de ‘remote input/output module’ de regeling kunnen beïnvloeden[Soy, 2016d]. De sensoren die aan de ‘remote input/output module’ hangen moeten een minimale spanning tussen de 0-2V of 4-20mA produceren met minimaal 8-bit resolutie om de regelingen te kunnen beïnvloeden[Schipper, 2015]. Resolutie minder dan 8-bit beïnvloedt de regelingen niet.

Wat is een Raspberry Pi 2?

De Raspberry Pi 2 is een minicomputer dat ontwikkeld wordt als development platform voor scholen om studenten ermee te leren programmeren. Verder wordt het ook veel gebruikt door hobbyisten. Maar het kan ook heel goed gebruikt worden voor het besturen van allerlei processen uiteenlopend van heel simpel tot heel complex[Foundation, 2015b].

Welke communicatie interfaces zijn er mogelijk met de Raspberry Pi 2

Op de Raspberry Pi 2 zijn 40 “General Purpose Input Output” (GPIO) pinnen aanwezig. Hiermee kan er hardwarematig een extensie bordje (shield) aan de Raspberry Pi gekoppeld worden. Op deze manier kan de Raspberry Pi 2 communiceren met allerlei extra toepassingen, zoals sensoren, schermen enz. Voor het prototype zijn deze pinnen van groot belang, omdat de shields door middel van deze pinnen aan de Raspberry Pi aangesloten kunnen worden en data kunnen uitwisselen door middel van standaarden zoals ‘Serial Peripheral Interface’ (SPI), ‘Inter-integrated Circuit’ (I²C), ‘Universal Asynchronous Receiver Transmitter’ (UART)[Foundation, 2015a] en ‘Pulse Width Modulation’ (PWM). Tevens is er de mogelijkheid om door middel van de USB-poorten data uit te kunnen uitwisselen.

Wat is een shield?

Een shield is een modulaire printplaat waarop verschillende microchips, sensoren en allerlei modules geïntegreerd zijn. Een shield wordt een 'expansion board' oftewel een uitbreidingsbordje genoemd. Een shield wordt gebruikt om een Raspberry Pi of soortgelijke elektronische platformen te kunnen uitbreiden met extra functionaliteiten die normaliter niet aanwezig zijn op de Raspberry Pi [Jimb0, 2015].

Hoe wordt de communicatie opgebouwd tussen de shields en de Pi 2?

De communicatie tussen het shield en de Pi 2 hangt af van de communicatie interface die een shield ondersteunt en of het desbetreffende shield compatible is met de Pi 2. Zoals hierboven in de deelvraag 'Welke communicatie interfaces zijn er mogelijk met de Raspberry Pi 2' geanalyseerd te hebben, heeft de Pi 2 de mogelijkheid om door middel van de GPIO, SPI, I²C, UART, PWM en USB communicatie op te kunnen bouwen. Indien het shield een van deze communicatie interfaces ondersteunt, dan kan er door middel van deze interfaces communicatie opgebouwd worden met het desbetreffende shield. Het desbetreffende shield wordt dan op de GPIO pinnen van de Pi 2 gezet om er een verbinding mee te kunnen maken. Deze pinnen moeten natuurlijk juist geklikt worden op de desbetreffende GPIO pinnen, anders kan er geen communicatie optreden. Soms komt ook voor dat het shield niet rechtstreeks op de GPIO pinnen van de Pi 2 wordt aangesloten, maar door middel van een USB-kabel wordt gekoppeld [Phidgets.com, 2012]. Daarnaast moet er ook een stuk software gedraaid worden om het desbetreffende shield aan te kunnen spreken. Hiervoor hebben sommige shield leveranciers voorbeeld code of voorbeeld applicaties beschikbaar.

Welke shields zijn beschikbaar voor de Pi 2 om de regelingen te kunnen beïnvloeden in klimaatcomputers?

Uit de deelvraag: 'hoe kan een regeling worden beïnvloed' blijkt dat er een minimale spanning tussen de 0-2V geproduceerd moet worden met een minimale resolutie van 8-bit om de regelingen te kunnen beïnvloeden [Schipper, 2015]. Uitgaande van deze informatie moeten shields uitgezocht worden die een minimale spanning tussen de 0-2V kunnen genereren met minimaal 8-bit resolutie. De volgende shields zijn beschikbaar voor de Pi 2 om regelingen te kunnen beïnvloeden:

- 4-20 mA T click shield van Mikroe Elektronika [Mikroe.com, 2016].
- ADC-DAC Pi van AB electronics UK [ADC-DAC-PI-shield, 2016].
- Expander Pi van AB electronics UK [Expander-Pi-shield, 2016].
- Phidget Analog 4-Output van Phidgets.com [Phidgets.com, 2012].
- 4-20 mA sensor board voor Arduino, Raspberry Pi en Galileo van Cooking Hacks [Cooking-Hacks, 2016].
- UniPi board van UniPi .Technology [Technology, 2016].
- Custard Pi 2 van sf-innovations.co.uk [sf innovations, 2016].
- Open Source Analog Input/Output Expansion Board van Osexpansionboard.de [Laukien, 2015]

Hoe kan er communicatie opgebouwd worden tussen het prototype en LetsGrow.com?

Het prototype heeft een 'ethernet-poort' aan boord en kan met behulp van een 'ethernet-kabel' verbonden worden met het Internet. LetsGrow heeft een aantal methoden om met externe systemen te kunnen koppelen en om data beschikbaar te stellen. Varianten zijn SOAP en REST services[Soy, 2016c]. Dit gebeurt op basis van de LetsGrow.com Web API[Vis, 2016]. Daarnaast hebben deze hardware componenten die aan het internet verbonden zijn, een cliënt applicatie aan boord die door LetsGrow.com zelf is ontworpen. Aan de hand van de Web API die LetsGrow.com zelf ontworpen heeft, kunnen deze cliënt systemen een verbinding maken met de LetsGrow.com servers en gegevens uitwisselen. Deze cliënt applicaties doen hun werk op normale pc's. Om met LetsGrow communicatie op te kunnen bouwen, moet de LetsGrow.com Web API geïmplementeerd worden in deze cliënt applicatie van het prototype.

Hoe kan er communicatie opgebouwd worden tussen het prototype en de klimaat-computer?

Zoals eerder geanalyseerd te hebben, bestaat de klimaatcomputer uit een bedieningscomputer en een regelstation. Het regelstation regelt de regelingen en met behulp van de bedieningscomputer komen hier sturing waardes naartoe. Vanuit de "remote input/output module" komen er meting waardes het regelstation binnen[Soy, 2016b]. Het prototype moet dus een "analoge spanning" aan deze "remote input/output module" doorgeven. De communicatie tussen deze "input/output module" en de Pi 2 wordt gerealiseerd door de ingangen van het desbetreffende shield te koppelen met een draad aan de ingangen deze 'remote input/output module'. Deze draden kunnen op de analoge uitgangen van het desbetreffende shield gesoldeerd worden. Daarna kunnen de andere uiteinden van deze draden in de 'remote input/output module' ingeklikt worden. Hiermee kan het prototype communicatie opzetten met de klimaatcomputer.

In deze analyse zijn vervolgens ook de volgende deelvragen en sub-deelvragen behandeld.

- Hoe kan er middels de ontwikkelomgeving Visual Studio en de programmeertaal C# een applicatie gebouwd worden voor de Raspberry Pi 2 draaiend onder Windows IoT die een aangesloten shield aanstuurt?
 - Wat is Windows IoT
 - Wat is Visual Studio
 - Welke typen applicaties zijn er mogelijk op Windows IoT?
 - Welke communicatie interfaces ondersteunt de Raspberry Pi 2 met Windows IoT?
 - Hoe kunnen er door middel van Visual Studio applicaties gebouwd en gedebugd worden op het Windows IoT besturingssysteem?
 - Hoe kan er een applicatie gebouwd worden om de Input/Output van de Raspberry Pi 2 aan te kunnen sturen op het Windows IoT besturingssysteem?
- Hoe kan er middels de ontwikkelomgeving MonoDevelop en de programmeertaal C# een applicatie gebouwd worden voor de Raspberry Pi 2 draaiend onder Raspbian die een aangesloten shield aanstuurt?
 - Wat is Raspbian
 - Wat is MonoDevelop
 - Welke typen applicaties zijn er mogelijk op Raspbian?
 - Welke communicatie interfaces ondersteunt de Raspberry Pi 2 met Raspbian?
 - Hoe kunnen er door middel van MonoDevelop applicaties gebouwd en gedebugd worden op de Raspbian?
 - Hoe kan er een applicatie gebouwd worden om de Input/Output van de Raspberry Pi 2 aan te kunnen sturen op het Raspbian besturingssysteem.

Alleen zijn de bovengenoemde vragen niet opgenomen in dit verslag, omdat het op dezelfde manier is onderzocht als hierboven beschreven en deze vragen geven teveel gedetailleerd informatie weer.

6.4 Conclusie analyse hardware, besturingssystemen en ontwikkeltools

De hoofdvraag voor de analyse was: Hoe kunnen regelingen in klimaatcomputers beïnvloed worden zonder de tussenkomst van de betreffende klimaatcomputer leverancier?

Aan de hand van de analyse is gebleken dat dit gerealiseerd kan worden door middel van de onderstaande hardware/software componenten te gebruiken.

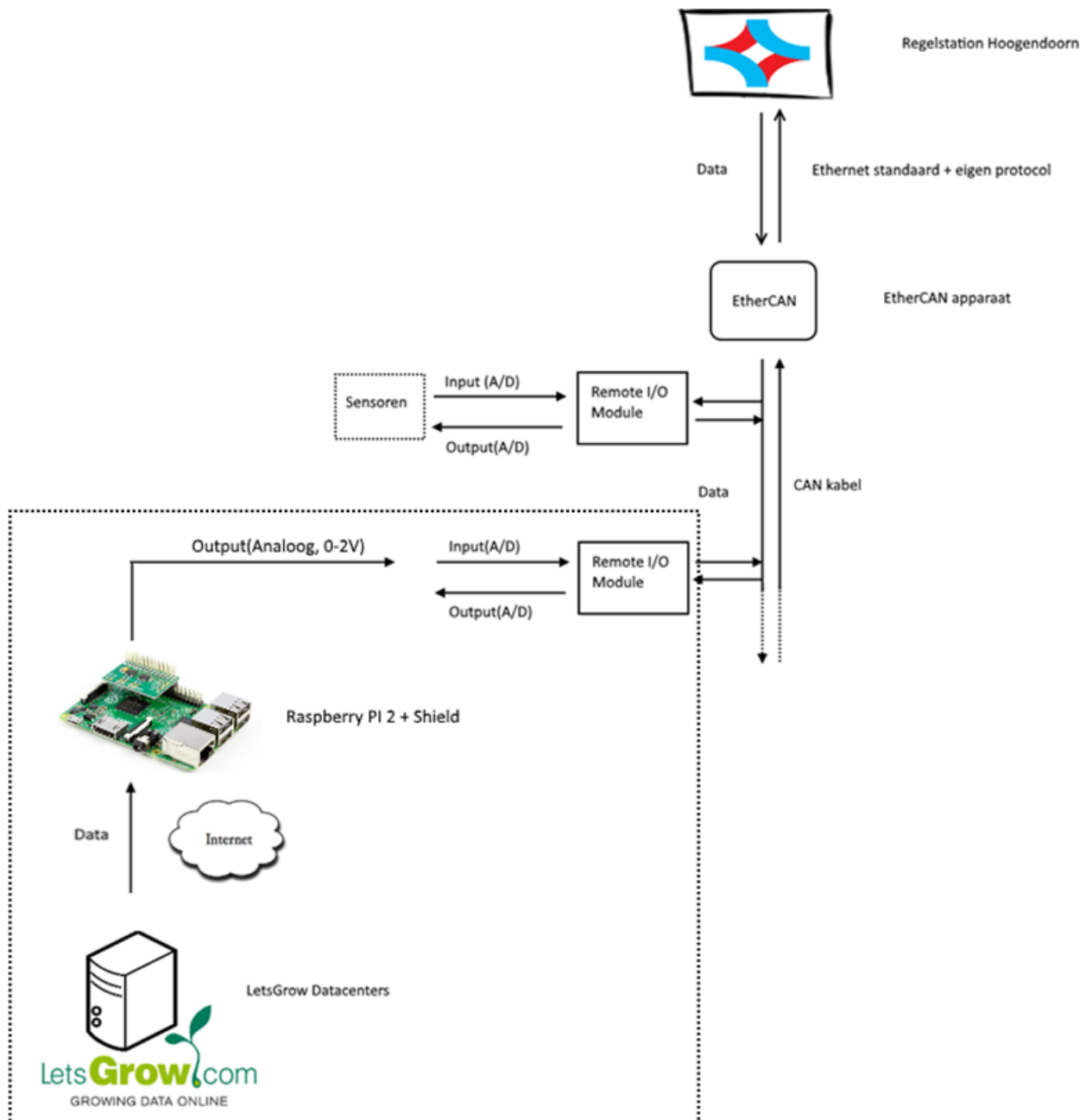
- Een “Power Supply” Micro-USB adapter voor de Raspberry Pi 2
- Een Micro SD-kaart van minimaal 8GB voor IoT en/of minimaal 4GB voor Raspbian
- Een Raspberry Pi 2 draaiend op het Windows IoT besturingssysteem en/of Raspbian besturingssysteem.
- Raspberry Pi 2 met een actieve internetverbinding door middel van een ‘UTP’ netwerkkabel.
- Een applicatie die geschreven is met het programmeertaal C# en de library LetsGrow Web API en een van de communicatie interface libraries bevat om met de ingangen en de uitgangen van de Pi 2 te kunnen aansturen.
- Een van de gevonden shields gekoppeld aan de Pi 2, die minimaal een spanning kunnen produceren tussen de 0-2V (voltages) en/of 4-20mA (milli ampère) met een 8-bit resolutie.
- Het prototype bestaand uit de bovengenoemde punten gekoppeld aan de “remote input/output module” van de klimaatcomputer.

Om een keuze te kunnen maken welke besturingssystemen, applicatie type en shield gebruikt gaan worden, moesten eerst de eisen achterhaald worden. Nadat de eisen en de wensen bekend waren, kon er een keuze gemaakt worden door de afstudeerder welk van de bovengenoemde componenten voor het prototype gebruikt gaan worden.

6.5 De wens van de opdrachtgever

De wens van de opdrachtgever is om naar “alle” merken klimaatcomputers data te kunnen sturen, zodat LetsGrow.com de regelingen van alle klimaatcomputers kan beïnvloeden.

Om deze te kunnen realiseren, is er uit de analyse gebleken dat het prototype een minimale spanning tussen de 0-2V met een minimale resolutie van 8-bit moet produceren. Het prototype moet aan de “input/output module” systeem van de klimaatcomputer gekoppeld worden door middel van bedradingen, zodat LetsGrow.com toch de mogelijkheid heeft om de regelingen te kunnen beïnvloeden zonder de tussenkomst van de klimaat leveranciers. Om de nieuwe situatie simpeler af te beelden, is hieronder een schematische tekening gemaakt van de 'gewenste situatie'.



Figuur 8: Gewenste situatie

6.6 De requirements

De eisen zijn met de bedrijfsmentoren vastgesteld [Soy, 2016e]. De functionele eisen waren belangrijk om te weten komen wat het systeem moest doen. Daarnaast zijn ook de niet-functionele eisen samengesteld, omdat deze ook heel erg belangrijk waren. Namelijk de afstudeerder moest ook de hardware componenten selecteren en hiervoor moest de afstudeerder weten of de bedrijfsmentoren ook eisen aan de hardware hadden vastgesteld. De onderstaande eisen zijn uiteindelijk gevalideerd door de bedrijfsmentoren.

6.6.1 Functionele eisen:

1. Het prototype moet automatisch opstarten, nadat het prototype voorzien is van stroom.
2. Het prototype moet zelf het verkeer initiëren met de servers om firewall problemen te kunnen voorkomen.
3. Het prototype moet indien er geen actieve netwerkverbinding aanwezig is een minimale tijd van 60 minuten niets doen.
4. Het prototype moet door middel van een 'uniek identificatienummer' bij de servers authenticeren.
5. Het prototype moet zelfstandig/automatisch zijn werking(en) uit gaan voeren, zonder de tussenkomst van een persoon.
6. Het prototype moet om de 5 minuten met de LetsGrow.com servers connectie maken om gegevens te kunnen ophalen.
7. Het prototype moet zijn configuratie informatie sturen naar de servers, indien deze niet aanwezig is.
8. Het prototype moet 'minstens één' analoge waarde kunnen sturen naar een klimaatcomputer.
9. De LetsGrow Web API server moet een onbekende 'uniek identificatienummer' loggen in zijn logging database.
10. Het prototype moet aan de hand van zijn 'leds' een indicatie kunnen tonen van zijn interne status.

6.6.2 Niet-functionele eisen:

1. De hardware van het prototype moet minimaal soldeerwerk bevatten. (minimaal in de zin van: het soldeerwerk moet beperkt blijven tot het solderen van headers/pinnetjes en kabeltjes op desbetreffende uitgangen; geen weerstanden/condensatoren).
2. De hardware van het prototype moet binnen 5 werkdagen geleverd kunnen worden.
3. Het prototype moet zijn besturingssysteem periodiek stil "unattended" updaten.
4. De hardware van het prototype moet van een bekende leverancier komen.
5. Een hardware onderdeel van het prototype mag niet meer dan 100 euro kosten.
6. De applicatie van het prototype moet gebouwd worden in Visual Studio met het programmeertaal C#.
7. Het prototype moet draaien op Windows IoT besturingssysteem.
8. Het prototype moet gebouwd worden met Visual Studio.
9. De applicatie code van het prototype moet voldoen aan de 'ifSQ level-1' richtlijnen en de 'naming-guidelines' richtlijnen van Microsoft.
10. Het prototype moet 24 uur per dag draaien.
11. Configureren van itemcodes moeten via de LetsGrow.com website gebeuren.

6.7 De keuze voor het besturingssysteem, ontwikkeltool en applicatie type

Na de analyse moest de afstudeerder een keuze maken voor besturingssysteem, ontwikkeltool en applicatie type die gebruikt gingen worden voor het prototype. Om een keuze te maken, heeft de afstudeerder eerst gebrainstormd wat het relevant zou zijn voor het te bereiken doel. Daarnaast is er aan de hand van analyse inzicht verkregen hoe sommige onderdelen gerealiseerd kunnen worden. Aan de hand van deze zaken zijn de volgende uitgangspunten opgesteld.

1. Ondersteunt het besturingssysteem de GPIO, SPI , I²C, USB interface communicatie van de Raspberry Pi 2?

Dit uitgangspunt is voortgekomen uit de analyse. Om de shields te kunnen aanspreken moeten de communicatie interfaces van de Raspberry Pi 2 aangesproken worden. Het besturingssysteem moet natuurlijk de communicatie interfaces aan kunnen sturen, anders kan er geen communicatie opgebouwd worden met het desbetreffende shield.

2. Is het mogelijk om automatische updates te kunnen uitvoeren op het desbetreffende besturingssysteem?

Dit uitgangspunt is voortgekomen uit de niet-functionele eisen. De opdrachtgever wilde een 'veilige omgeving' hebben. Hierdoor moet het besturingssysteem up-to-date zijn met de laatste beveiligingsupdates.

3. Welke applicatie types ondersteunt desbetreffend besturingssysteem?

Dit uitgangspunt is voortgekomen deels door niet-functionele eisen en deels door de afstudeerder. Uit de analyse en niet-functionele eisen blijkt dat het prototype zonder de tussenkomst van personen moet werken. Hierdoor moet er gekeken worden welke typen applicaties beschikbaar zijn. Een applicatie zonder gebruikersinterface kan dan gerealiseerd worden.

4. Is er ontwikkelsoftware beschikbaar voor het desbetreffende besturingssysteem?

Dit uitgangspunt was voor de afstudeerder belangrijk, omdat er ook buiten Visual Studio en MonoDevelop gekeken moest worden, indien deze ontwikkeltools niet naar behoren zouden gaan werken.

5. Is het mogelijk om met de programmeertaal C# te kunnen programmeren/debuggen op het desbetreffende besturingssysteem?

Dit uitgangspunt is voortgekomen uit de niet-functionele eisen. De bedrijfsmentoren willen met de programmeertaal C# ontwikkelen.

6. Heeft het besturingssysteem support van een bedrijf/community?

Dit uitgangspunt is indirect voortgekomen uit de niet-functionele eisen. Het besturingssysteem moet 24 uur per dag lang draaien, dus het besturingssysteem wat op het prototype draait moet ook stabiel draaien. De kans is groot als er support wordt verleend van een bedrijf dat de 'bugs' ook snel opgelost kunnen worden.

7. Voorkeur bedrijfsmentoren?

Dit uitgangspunt spreekt voor zich. De bedrijfsmentoren willen voor het prototype gebruik maken van Windows IoT besturingssysteem met als ontwikkeltool Visual Studio en C# als programmeertaal.

Uit verdere analyse en de bovenstaande uitgangspunten is de volgende tabel uitgekomen:

Tabel 1: Analyse besturingssystemen, ontwikkeltools

	Windows IoT	Raspbian
1. Ondersteunt het besturingssysteem de GPIO, SPI , I ² C, USB interface communicatie van de Raspberry Pi 2?	Door middel van de Windows.Devices.SPI /I ² C/ Serial,Communication en/of Windows.Devices.GPIO libraries die standaard geleverd wordt met Windows IoT kunnen met de communicatie interfaces communicatie opgebouwd worden. USB ondersteuning is ook mogelijk.	Er moet eerst in de raspi-config de I ² C, SPI enz. aangezet worden. Daarna kan er door middel van RaspiSharp/ RaspberryPi .net/ Raspberry# IO communicatie opgebouwd worden met de desbetreffende communicatie interface. USB ondersteuning is ook mogelijk.
2. Is het mogelijk om automatisch updates te kunnen uitvoeren op het desbetreffende besturingssysteem?	Windows IoT voert de automatische updates altijd uit. Deze kan jammergenoeg niet uitgezet worden. Handmatig kan er wel gecheckt worden op updates.	Het is mogelijk om met scripts en cronjobs “unattended updates” periodiek te kunnen uitvoeren.
3. Welke applicatie types ondersteunt desbetreffend besturingssysteem?	Windows IoT heeft ondersteuning voor 'Headless' en 'Headed' applicatie types die in formaat UWP (Universal Windows App) zijn.	Raspbian ondersteunt verschillende typen applicaties. (van achtergrond applicaties tot applicaties met gebruikersinterface) Het hangt van de ontwikkeltools en Framework af welke gebruikt kunnen worden. Om C# programma's te kunnen draaien, moet open-source Mono. NET Framework geïnstalleerd worden.
4. Is er ontwikkelsoftware beschikbaar voor het desbetreffende besturingssysteem?	Ja, Visual Studio om te bouwen en te debuggen.	Ja, Monodevelop om C# code te kunnen bouwen en te kunnen debuggen. Daarnaast kan er ook andere ontwikkeltools geïnstalleerd worden voor alle andere programmeertalen.
5. Is het mogelijk om met de programmeertaal C# te kunnen programmeren/debuggen op het desbetreffende besturingssysteem?	Windows IoT ondersteunt VB.net, C#, C++, Python en Javascript	Op de Raspbian kan er ontwikkeld worden met allerlei verschillende programmeertalen. Het hangt van de gebruiker af met welke taal hij wil gaan ontwikkelen. Het is ook mogelijk om met C# programmeertaal te kunnen programmeren. Hiervoor moeten wel extra Frameworks geïnstalleerd worden.
6. Heeft het besturingssysteem support van een bedrijf/community?	Support van Microsoft. Ze brengen ook regelmatig updates uit.	Support van open community en klein beetje ook van Raspberry Pi Foundation.
7. Voorkeur bedrijfsmentoren?	Bedrijfsmentoren hebben een voorkeur voor Windows IoT	Bedrijfsmentoren hebben een voorkeur voor Windows IoT

Uit bovenstaande tabel 1 blijkt dat een oplossing voor het prototype met beide besturingssystemen gerealiseerd kan worden. Met beide besturingssystemen kunnen de communicatie interfaces van de Raspberry Pi 2 aangesproken worden. Beide besturingssystemen bevatten ontwikkeltools en de mogelijkheid om in de programmeertaal C# te kunnen bouwen en te kunnen debuggen. Daarnaast kunnen er op beide besturingssystemen soortgelijke type applicaties gebouwd worden.

De niet-functionele eisen laten zien dat de bedrijfsmentoren een voorkeur hebben voor Microsoft omgeving en ze willen dan ook graag Windows IoT met Visual Studio als ontwikkelomgeving gebruiken voor het prototype. Vanuit dit oogpunt is bij dit prototype gekozen om gebruik te maken van het Windows IoT besturingssysteem. Daarnaast is het ook handig om alleen Visual Studio te gebruiken als ontwikkeltool, omdat de bedrijfsmentoren hiermee al werken. Ze hoeven hiermee niet MonoDevelop te leren.

Tevens is er gekozen om gebruik te maken van een 'Headless Universal App' applicatie type voor het prototype, omdat het prototype zonder tussenkomst van personen en klimaatcomputer leveranciers zijn werking moet doen en hiervoor heeft de applicatie dus geen gebruikersinterface nodig. Daarnaast brengt 'Universal App' voordelen met zich mee. De ontworpen applicatie kan later op andere platformen gedraaid worden, zonder dat er veel applicatie code veranderd hoeft te worden.

6.8 Vervolg analyse shields

Het tweede stuk van het prototype bestond uit een shield dat mogelijk moest maken om de digitale gegevens te kunnen omzetten naar analoge voltages/milli ampère om deze door te kunnen sturen naar een klimaatcomputer. Uit de eerste analyse bleek dat er 8 shields beschikbaar waren om de klimaatcomputers te kunnen beïnvloeden.

Om verder te kunnen analyseren, moesten uit deze beschikbare shields geselecteerd worden om ingekocht te worden. Om te kunnen selecteren is de afstudeerder uitgegaan van de niet-functionele eisen waar het prototype aan moest voldoen. Uit de niet-functionele eisen en de analyse kwamen de onderstaande zaken naar voren:

1. Shield moet minimaal één analoge uitgang hebben van 0-2V of 4-20mA[Schipper, 2015].
2. Shield moet maximaal binnen 5 werkdagen leverbaar kunnen zijn.
3. Shield mag niet meer dan 100 euro kosten.
4. Shield moet zo simpel mogelijk gekoppeld kunnen worden aan de Raspberry Pi 2 (met een USB kabel of door middel van pinnen).
5. Shield moet van een bekende (te vertrouwen) leverancier komen.
6. Shield mag de volgende interfaces hebben SPI , I²C, UART, GPIO of USB.

De uitgangspunten 1 en 4 spraken voor zich. Voor uitgangspunt 2 was de levertijd genomen van de officiële leverancier. Er is geen gebruik gemaakt van wederverkopers omdat hiervan de levertijden in de regel onbetrouwbaar waren. Immers waren deze wederverkopers op hun beurt weer afhankelijk van de hoofd leverancier. Deze levertijd is bepaald door het desbetreffende shield in het winkelmandje te zetten en te kijken wat er als levertijd op het scherm komt te staan. Voor het uitgangspunt 3 gold ook een soortgelijke beredenering. De prijzen konden namelijk per wederverkopers verschillen. Daarnaast golden de genoemde prijzen voor alleen de shields. Dit hield in dat als er een 'tussenshield' nodig was, dat de totale prijs per shield kon verschillen. Ook speelde de wisselkoers een grote rol bij de prijsbepaling. Er is dus uiteindelijk uitgegaan van de officiële leverancier prijzen.

Bij uitgangspunt 5 was er uitgegaan van de Hoogendoorn 'hardware' medewerker waarbij hij onder andere verantwoordelijk voor selectie en inkoop van hardware was. Hij had een lijst met allerlei leveranciers. Deze lijst was met de jaren mee samengesteld om te kijken welke leverancier te vertrouwen was en welke niet te vertrouwen was. Er is voor dit uitgangspunt een controle uitgevoerd aan de hand van deze lijst.

Bij uitgangspunt 6 was uitgegaan dat er om de 5 minuten wordt gecommuniceerd met Lets-Grow waarbij er weggeschreven data opgehaald wordt. Deze data moet minimaal een analoge waarde voor een analoge uitgang bevatten. De verkregen data kan paar seconden vertraging hebben. De applicatie hoeft niet 'real-time' te kunnen beïnvloeden. Hierdoor maakt het niet uit in hoeveel seconden de spanning weggeschreven kunnen worden. Uit de analyse blijkt dat de snelheden van SPI, I²C, UART en USB hieraan voldoen. Daarnaast is er bij SPI een limiet aan aantal 'apparaten' dat de SPI bus kan accepteren, alleen is deze ook niet van toepassing voor het prototype, omdat er maar een apparaat met een shield gebruikt gaat worden. Er gaat wel bekeken worden welke interface het desbetreffende shield heeft, maar deze interface zal geen invloed hebben op de uiteindelijke shield keuze.

Aan de hand van de 6 uitgangspunten en verdere selectie is de volgende tabel samengesteld:

Tabel 2: Analyse beschikbare shields

	1	2	3	4	5	6
4-20 mA T click	1 analoge uitgang van 4-20 mA	Leverbaar in 2-3 dagen	26.00 Dollar	Klikbaar, maar vereist wel een tussenshield	MikroElektronica - bekend	SPI
ADC-DAC Pi	2 analoge uitgangen van 0-2V	Leverbaar in 1-2 dagen	9.98 Pond	Klikbaar op de GPIO	Abelectronics - bekend	SPI
Expander PI	2 analoge uitgangen van 0-2V	Levertijd onbekend	19.99 Pond	Klikbaar op de GPIO	Abelectronics - bekend	SPI
Phidget Analog 4-Output	4 analoge uitgangen van 0-10V	Leverbaar in 3-5 dagen	90.00 Dollar	Door middel van USB kabel	Phidgets.com - bekend	USB
4-20 mA sensor board	4 analoge uitgangen van 4-20 mA.NOTE De analoge uitgangen vereisen externe spanningsbron op de Pi 2	Leverbaar in 5-7 dagen	46.00 Euro	Klikbaar, maar vereist wel een tussenshield	Cooking Hacks - bekend	I2C
UniPi board	1 analoge uitgang van 0-10V	Leverbaar in 1 dag	149 Euro	Door middel van PIN connectoren (flat kabels)	UniPi technology - onbekend	I2C
Open Source Analog Input/Output Expansion Board	8 analoge uitgangen van 0-10V of 0-24V	Nog niet leverbaar.	Nog in ontwikkeling.	Door middel van PIN connectoren. (flat kabels)	Osexpansionboard.de - onbekend	SPI
Custard Pi 2	2 analoge uitgangen van 0-3.3V	Levertijd onbekend	19 Pond	Klikbaar op de GPIO	SF Innovations - onbekend	SPI

Groen gekleurd: voldoet volledig aan het uitgangspunt; Geel: voldoet niet aan het uitgangspunt

In bovenstaande tabel 2 is af te lezen dat de volgende 3 shields aan de uitgangspunten voldoen. Deze 3 shields zijn ingekocht om verder geanalyseerd te worden:

- 4-20 mA T click shield van Mikroe Elektronika.
- ADC-DAC Pi van AB electronics UK.
- Phidget Analog 4-Output van Phidgets.com.

In de verdere analyse zijn de volgende uitgangspunten door de afstudeerder vastgesteld die relevant waren voor het te realiseren doel:

1. Heeft de leverancier zelf een library/voorbeeld applicatie uitgegeven in een C/C# of Python programmeertaal en kunnen met deze library de analoge uitgangen aangestuurd worden?

De afstudeerder heeft dit uitgangspunt opgesteld om tijd te besparen. Als er een library voorhanden is scheelt dit immers programmeer- en testwerk.

2. Indien er geen libraries beschikbaar zijn, kan er door standaard library die het desbetreffende besturingssysteem bezit, communicatie opgebouwd worden met het desbetreffende shield om de analoge uitgangen aan te sturen?

Dit uitgangspunt moet worden uitgevoerd, indien het eerste uitgangspunt niet realiseerbaar is. Anders kon het desbetreffende shield niet aangestuurd en getest worden. Om de shields te kunnen testen in de praktijk is er voor dit uitgangspunt gebruik gemaakt van de standaard SPI library met een eigen opgestelde voorbeeld code/test programma. Dit test programma werd op de Raspbian (met de een van de C# sharp IO libraries) en Windows IoT (met de Windows.Devices libraries) gedraaid. Per shield werd er gekeken of er een waarde weggeschreven kon worden naar de desbetreffende shield. Uiteindelijk werd er met een multimeter gecontroleerd of het ook daadwerkelijk was weggeschreven.

3. Is er een officiële ondersteuning voor Windows IoT/Raspbian?

Dit uitgangspunt is voortgekomen uit de niet-functionele eisen. Voor het prototype ging Windows IoT gebruikt worden, alleen was er wel een ondersteuning van het shield leverancier voor IoT?. Raspbian is ook gelijk meegenomen met dit uitgangspunt om het alternatieve besturingssysteem ook te analyseren op dit criteria.

4. Haalt het desbetreffende shield ook in praktijk de minimale 8-bit resolutie?

Dit uitgangspunt is voortgekomen uit de analyse. Om regelingen te kunnen beïnvloeden moest er minimaal 8-bit resolutie vereist zijn. Hierdoor moet in praktijk gekeken worden of het shield daadwerkelijk ook minimaal 8-bit resolutie spanning kan produceren. De nauwkeurigheid voor dit uitgangspunt wordt bepaald aan de hand van een multimeter. Deze multimeter wordt op 4000 millivolt (mV) gezet, zodat er bepaald kan worden of de shield daadwerkelijk een resolutie heeft van minimaal 8-bit.

Aan de hand van de 4 uitgangspunten en verdere selectie is de volgende tabel eruit gekomen:

Tabel 3: Analyse ingekochte shields

	4-20 mA T click	ADC-DAC Pi	Phidget Analog 4-Output
1. Heeft de leverancier zelf een library en/of voorbeeld applicatie uitgegeven in een C/C# of Python programmeertaal en kunnen met deze library de analoge uitgangen aangestuurd worden?	Voor zijn eigen software, 'Mikroc Pro for Pic' zijn libraries uitgegeven. Deze libraries waren onbruikbaar op de PI, waardoor er geen spanning gegenereerd konden worden	Abelektronics heeft libraries voor Windows IoT in C# en Raspbian in Python programmeertaal. Daarnaast heeft de leverancier ook voorbeeld applicaties beschikbaar voor allebei de besturingssystemen. Er kon met de library een voorbeeld spanning verstuurd worden naar de uitgangen.	Phidgets heeft verschillende libraries voor allerlei besturingssystemen in verschillende programmeertalen behalve voor IoT. Er kon op de Raspbian met behulp van C# library een voorbeeld spanning verstuurd worden naar de uitgangen.
2. Indien er geen libraries beschikbaar zijn, kan er door standaard library die het desbetreffende besturingssysteem bezit, communicatie opgebouwd worden met het desbetreffende shield om de analoge uitgangen aan te sturen?	Tijdens het testen is het gebleken dat er op geen enkel manier gecommuniceerd kan worden met dit shield. Nadat het shield opnieuw is besteld, is er weer geprobeerd maar ook met de 2e shield kon er geen communicatie opgebouwd worden.	Er kon met de standaard SPI library van Windows IoT als Raspbian een communicatie opgebouwd worden. Op de Raspbian moest er wel de Python 2/3 libraries gebruikt worden. Deze kon in C# gebruikt worden door "dllimport" functie.	Er kon met de standaard SPI library in Raspbian een communicatie opgebouwd worden. Op Windows IoT lukte het niet om communicatie op te kunnen bouwen.
3. Is er officiële ondersteuning voor Windows IoT/Raspbian.	Geen officiële ondersteuning voor Windows IoT en Raspbian. NOTE: Dit is wel opvallend, omdat ze zelf het tussenshield Pi Click Shield verkopen om het te laten werken met de Pi 2.	Er is officiële ondersteuning aanwezig voor Windows IoT als Raspbian	Er is een officiële ondersteuning aanwezig voor Raspbian, maar geen officiële ondersteuning voor Windows IoT.
4. Haalt het desbetreffende shield ook in praktijk de minimale 8-bit resolutie?	Kon niet getest worden.	Er is gebleken uit het testen dat er een nauwkeurigheid van 2 cijfers achter de komma waargenomen kon worden.	Er is gebleken uit het testen dat er een nauwkeurigheid van 2 cijfers achter de komma waargenomen kon worden.

Groen gekleurd: voldoet volledig aan het uitgangspunt; Geel: voldoet gedeeltelijk aan het uitgangspunt; Rood: Voldoet niet aan het uitgangspunt.

6.9 De keuze voor het shield

Uit de bovenstaande tabel 3 kon er geconstateerd worden dat ADC-DAC Pi het enige shield was dat op beide besturingssystemen werkte en waarvoor ook voorbeeld applicaties beschikbaar waren. Daarnaast kon er ook door middel van SPI standaard libraries met het shield gecommuniceerd worden.

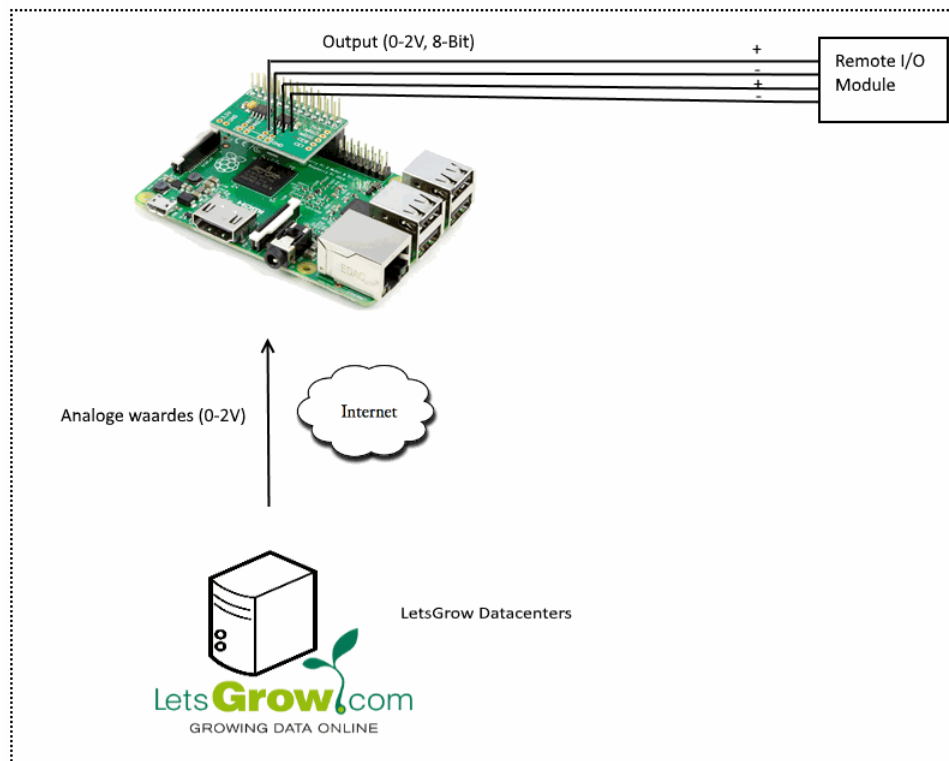
Phidget Analog 4-Output shield werkte niet op de Windows IoT besturingssysteem wegens het ontbreken van “ARM” libraries voor de Windows IoT. Onder Raspbian werkte dit shield goed. Er kon een spanning geproduceerd worden tussen de -10 en 10 volt.

4-20 mA T click van Micro Elektronika werkte op geen van beide besturingssystemen. Dit shield is wel opnieuw besteld om de hardware matig defect buiten te sluiten. Alleen is er weer gebleken dat het op geen enkele manier communicatie opgebouwd kon worden met dit shield. Hierdoor is dit shield voor 'nu' voor dit prototype onbruikbaar verklaard.

In hoofdstuk 6.7 op bladzijde 24 was er al besloten om gebruik te maken van het Windows IoT besturingssysteem met Visual Studio als ontwikkeltool. Vanuit dit oogpunt is bij dit prototype gekozen om gebruik te maken voor het shield ADC-DAC Pi. Dit shield werkt onder Windows IoT goed. Tevens had de leverancier ook libraries beschikbaar gesteld voor Windows IoT en Windows IoT wordt ook officieel ondersteund. Het is gebleken dat de nauwkeurigheid in werkelijkheid ook minimaal 8-bit is. Het gekozen shield voldoet zowel aan alle functionele eisen als niet-functionele eisen die de bedrijfsmentoren hadden vastgesteld.

6.10 Architectuur nieuwe situatie

De afstudeerder heeft na de hardware, besturingssystemen en ontwikkeltools selectie, de architectuur van de nieuwe situatie in kaart gebracht.



Figuur 9: Nieuwe situatie

Door middel van het internet ontvangt de Raspberry Pi 2 de analoge waarden die tussen de 0-2V zijn van de servers. Deze waarden worden doorgegeven aan het ADC-DAC Pi shield die 2 analoge uitgangen heeft. Deze 2 uitgangen zijn door middel van 4 bedradingen van 75mm^2 gekoppeld aan de 'plus-kant' en aan de 'grond-kant' van 'remote input/output modules'.

6.11 UML diagrammen en beschrijvingen

In deze fase is ook het ontwerp van de systeemarchitectuur door middel van UML modelleertaal gerealiseerd. Er is gebruik gemaakt van de UML modelleertaal, omdat het voor de opdrachtgever beter te begrijpen is, zonder dat hij verstand heeft van programmeren. Daarnaast hebben bedrijfsmentoren ervaring met UML. Voor hen is het dan ook duidelijk en kunnen ze de gehele opbouw van de applicatie in een opslag zien. De afstudeerder heeft voor het opstellen van UML gebruik gemaakt van het boek 'Praktisch UML'[Warmer and Kleppe, 2011].

Niet alle UML diagrammen zullen behandeld worden, omdat de diagrammen op soortgelijke manier gerealiseerd zijn. Dit hoofdstuk is gebaseerd op het hoofdstuk 11 op bladzijde 49 (in de bijlage op de bladzijde 130) van het definitierapport.

Use-case diagram cliënt applicatie

Voor het prototype moest er een applicatie opgebouwd worden en de LetsGrow Web API server applicatie uitgebreid moeten worden, omdat er sprake was van een cliënt-server architectuur werking. Dit hield in dat er een cliënt applicatie opgebouwd moest worden die connectie moet maken met de LetsGrow.com Web API server. Daarnaast moest de server uitgebreid worden om de binnenkomende berichten van het prototype cliënt te kunnen ontvangen en te kunnen bewerken. Om deze te realiseren is eerst een use-case diagram samengesteld aan de hand van de functionele eisen.

Voor het use-case diagram van het prototype Pi cliënt applicatie zijn de use-cases gebaseerd op de volgende functionele eisen:

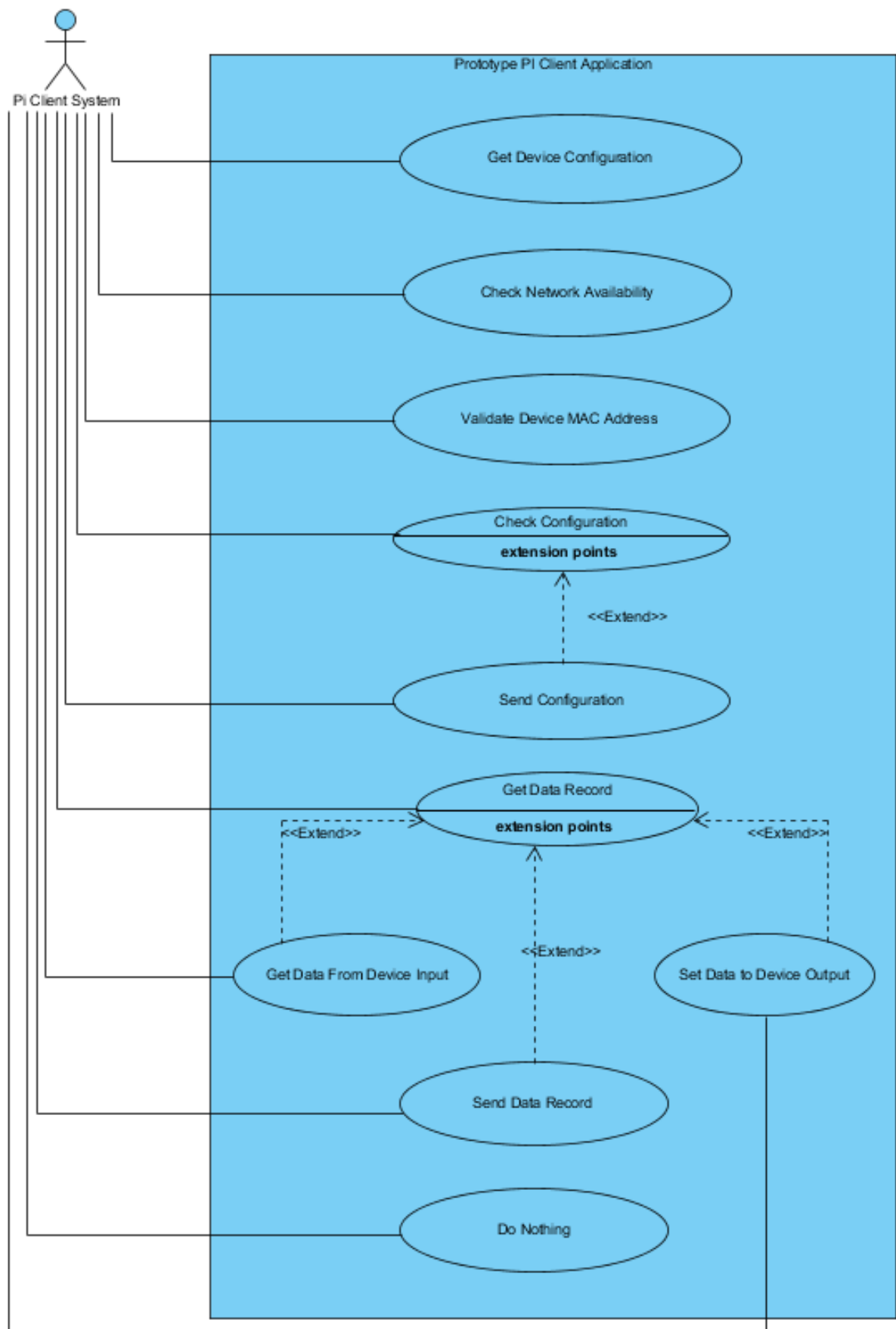
- Get Device Configuration: Functionele eis 1, 7
- Check Network Availability: Functionele eis 3
- Validate Device MAC address: Functionele eis 4
- Check Configuration: Functionele eis 7
- Send Configuration: Functionele eis 7
- Get Data Record: Functionele eis 5 en 6
- Set Data To Device Output: Functionele eis 8
- Do Nothing: Functionele eis 3

De functionele eis 2 wordt elke use-case gerealiseerd. Het prototype initieert per use-case de verbinding.

Note: 'Get Data From Device Input' en 'Send Data Record' use-cases zijn gebaseerd op de uitbreidbaarheid in de toekomst. Deze komen niet voort uit de eisen, maar zijn door de afstudeerder geïmplementeerd voor toekomstig gebruik.

De pijlen met 'extend' beschrijvingen houden in dat de betreffende use-case een uitbreiding met optioneel of uitzonderlijk gedrag bevat, oftewel: De use-case 'Get Data From Device Input', 'Set Data To Device Input' en 'Send Data Record' zijn 'extended' van de 'Get Data Record'. Dit houdt in dat de use-case 'Get Data Record' wordt uitgebreid met drie uitzonderlijk gedrag.

Op de volgende pagina in figuur 10 is het use-case diagram te zien van de Pi cliënt applicatie



Figuur 10: Use case diagram cliënt applicatie

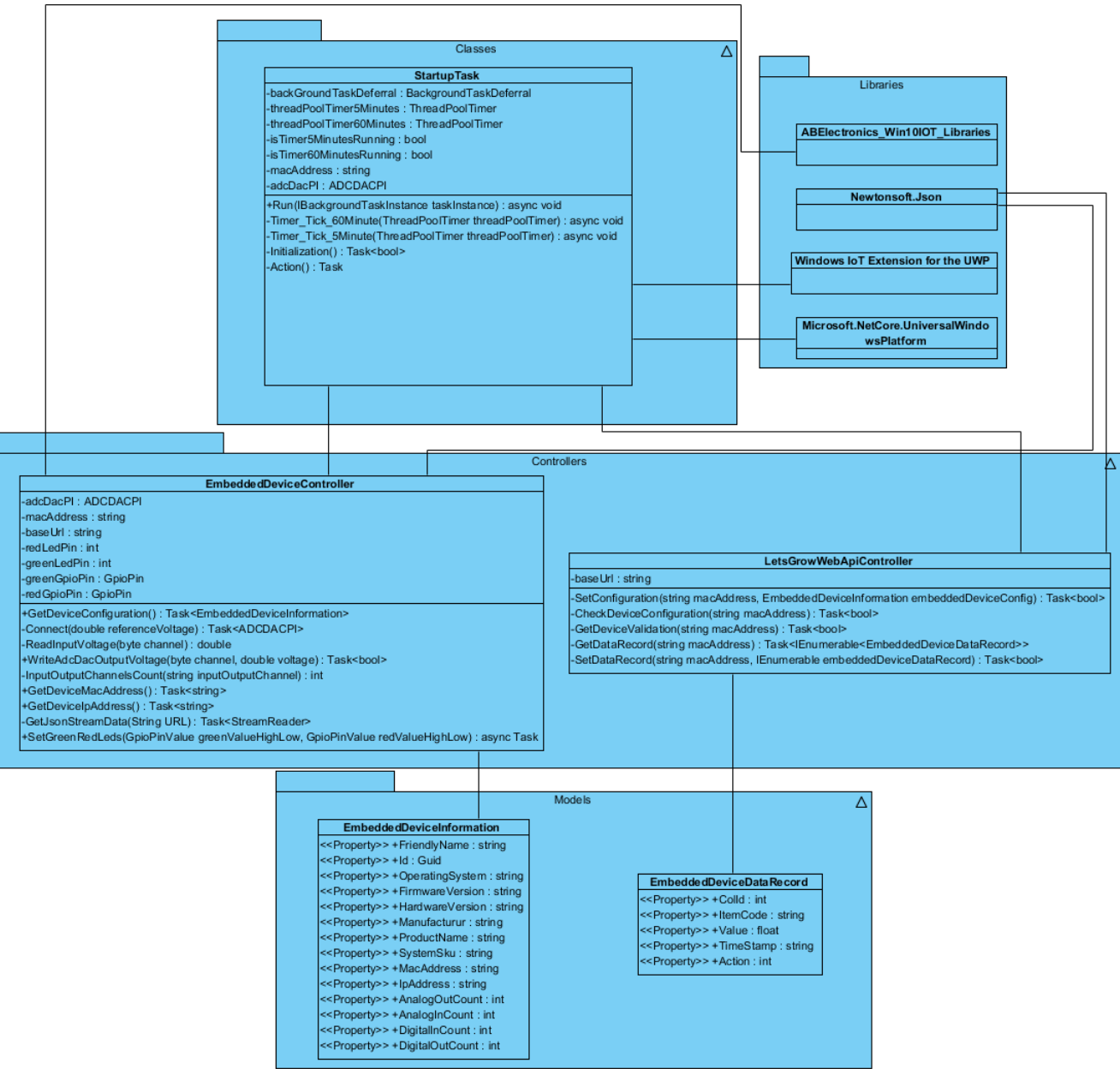
Use-case beschrijving

Alle use-cases zijn ook beschreven. De use-case 'Get Data Record' zal hieronder afgebeeld worden om te laten zien hoe het beschreven is. De template voor de use-case beschrijvingen is opgesteld uit de theorie afkomstig uit hoofdstuk 9 op bladzijde 97 uit het boek 'Praktisch UML, vijfde editie' van Jos Warmer en Anneke Kleppe.

Get Data Record	
Naam	Get Data Record
Samenvatting	Het Pi cliënt systeem gaat een datarecord ophalen van de WebServer.
Actoren	Het Pi cliënt systeem.
Aannamen	Het prototype moet een actieve internetverbinding hebben. Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server. Het Pi cliënt systeem heeft zijn configuratie check uitgevoerd. De Web API server moet online zijn.
Beschrijving	<ol style="list-style-type: none"> (1) Pi cliënt systeem vraagt aan de Webserver of hij een lijst van datarecord kan krijgen. (2) Web API server ontvangt dit aanvraag bericht. (3) Web API server stelt een lijst van datarecord op. (4) Web API server stuurt deze lijst van datarecord naar het Pi cliënt systeem. (5) Pi cliënt systeem ontvangt het antwoord van de Webserver (6) Pi cliënt systeem gaat de ontvangen lijst van data record processen en kijkt wat hij als actie moet ondernemen. Indien de actie wegschrijven is, dan treedt de [Data wegschrijven naar device uitgangen] uitzondering op. Indien de actie ophalen is, dan treedt de [Data ophalen van de device ingangen] uitzondering op. Indien de actie niets doen is, dan treedt de uitzondering [Doe niets] op. (7) Pi cliënt systeem is klaar om verder te gaan met de volgende werking.
Uitzondering	<p>[Data wegschrijven naar device uitgangen]</p> <ol style="list-style-type: none"> (1) Pi cliënt systeem voert de "Set Data To Device Output" use-case beschrijving uit. <p>[Data ophalen van de device ingangen]</p> <ol style="list-style-type: none"> (1) Pi cliënt systeem voert de "Get Data From Device Input" use-case beschrijving uit. <p>[Doe niets]</p> <ol style="list-style-type: none"> (1) Pi cliënt systeem voert de "Do Nothing" use-case beschrijving uit.
Resultaat	Het Pi cliënt systeem heeft succesvol een lijst van datarecord ontvangen van de Web API server en gaat deze procedure na 5 minuten weer herhalen.

Figuur 11: Use case 'Get Data Record' beschrijving cliënt applicatie

Klassendiagram cliënt applicatie



Figuur 12: Klassendiagram cliënt applicatie

Voor de architectuur van de cliënt applicatie en de LetsGrow.com Web API server applicatie is er door de afstudeerder gekozen om de architectuur in meerdere lagen te realiseren, zodat de functionaliteiten duidelijk gescheiden werden. Hierdoor had elke laag zijn 'eigen' verantwoordelijkheid. Daarnaast kunnen de lagen in de toekomst vervangen of uitgebreid worden zonder dat er een groot stuk van de applicatie gewijzigd hoeft te worden.

De afstudeerder heeft ook onderzocht welke ontwerppatronen [Gamma et al., 1994] toepasbaar zouden zijn voor de te bouwen applicaties. Zo is onder andere gekeken naar het Model-View-Controller ontwerppatroon. De te bouwen applicaties zullen geen gebruikersinterface bevatten. Er is daarom gekozen om de principes van het meerdere lagen ontwerppatroon aan te houden.

De Pi cliënt applicatie bestond uit de volgende lagen:

- Classes laag voor de 'business logic' van de cliënt applicatie
- Controllers laag voor het afhandelen van de HTTP verzoeken en apparaat handelingen zoals het communiceren met de analoge uitgangen van het gebruikte shield.
- Models laag voor het representeren van domein specifieke data om de configuratie informatie en de datarecords samen te stellen.
- Libraries laag voor te gebruiken bibliotheken binnen de applicatie

In deze lagen zijn de volgende klassen ingebouwd:

StartupTask klasse: De 'StartupTask' zorgt voor de "business logic" van het gehele prototype applicatie systeem. Hierin wordt eerst een 'BackgroundTask' aangemaakt en ook gestart. Daarna wordt deze 'Task' altijd in een actieve stand gehouden, zodat het prototype voor altijd in de achtergrond kan blijven werken. De methodes die in deze klasse voorkomen, zorgen ervoor dat er 'Timer' aangemaakt kan worden van 5 en 60 minuten, het gehele prototype proces opgestart kan worden en dat er een netwerkverbinding check uitgevoerd kan worden. Tevens wordt de verkregen een lijst van het type 'data record set' gesorteerd om te kijken wat het prototype als actie kan uitvoeren.

EmbeddedDeviceController klasse: Deze klasse zorgt ervoor dat er een communicatie opgebouwd kan worden met het desbetreffende shield. De methodes in deze klasse zorgen ervoor dat er apparaat configuratie opgehaald kan worden, geconnecteerd kan worden met het shield, analoge gegevens weggeschreven kunnen worden en opgehaald kunnen worden, MAC-adres en IP-adres opgehaald kunnen worden, shield aantal ingangen/uitgangen kanalen uitgelezen kunnen worden, een 'Logging' mapje op het prototype zelf aangemaakt kan worden en de logging naar een bestand weggeschreven kan worden. Note: Laatste functie is voor testdoeleinden geïmplementeerd, zodat de afstudeerder het prototype goed kon testen. Na het testen is deze functionaliteit uitgeschakeld.

LetsGrowWebApiController klasse: Deze klasse is verantwoordelijk voor het opzetten van de verbinding tussen het prototype applicatie en de LetsGrow.com Web Server door middel van het REST architectuur principe. De methodes in deze klasse zorgen ervoor dat het apparaat bij het LetsGrow server kan valideren, nieuwe lijst van het type 'data recordset' opgehaald en verzonden kunnen worden, configuratie informatie check uitgevoerd kan worden en configuratie van het apparaat verzonden kan worden.

Het gehele klassendiagram van de cliënt applicatie is in figuur 12 op de vorige pagina te vinden. Wegens leesbaarheid is dit klassendiagram opgedeeld in stukken en in de bijlagen A,B en C op de bladzijden 54,55 en 56 toegevoegd.

Om inzicht te geven in hoe een klasse opgebouwd en beschreven is, zal hieronder de klasse LetsGrowWebApiController behandeld worden.

LetsGrowWebApiController
-baseUrl : string
-SetConfiguration(string macAddress, EmbeddedDeviceInformation embeddedDeviceConfig) : Task<bool>
-CheckDeviceConfiguration(string macAddress) : Task<bool>
-GetDeviceValidation(string macAddress) : Task<bool>
-GetDataRecord(string macAddress) : Task<IEnumerable<EmbeddedDeviceDataRecord>>
-SetDataRecord(string macAddress, IEnumerable embeddedDeviceDataRecord) : Task<bool>

Figuur 13: LetsGrowWebApiController klasse

De in bovenstaand figuur 13 afgebeelde klasse is verantwoordelijk voor het opzetten van de verbinding tussen de cliënt applicatie en de LetsGrow.com Web Server door middel van het REST architectuur principe. De methodes in deze klasse zorgen ervoor dat het apparaat gevalideerd kan worden bij de LetsGrow.com server, lijst van data record opgehaald kunnen worden, nieuwe lijst van data record verzonden kan worden, configuratie check uitgevoerd kan worden en configuratie van het apparaat verzonden kan worden.

Door de afstudeerder is gekozen om de methode benamingen zo in te richten, zodat de toekomstige ontwikkelaar het snel kan oppakken. Daarnaast zorgen deze benamingen dat er in de code weinig commentaar genoteerd hoeft te worden en dat het gelijk aan de methode waargenomen kan worden wat zijn functie is.

Zowel de bedrijfsmentoren als de opdrachtgever een duidelijk beeld te geven over de functies van de methodes, zijn de klassen beschreven. In de onderstaande tabel 4 wordt de beschrijving van de LetsGrowWebApiController klasse getoond om te laten zien hoe de klassen beschreven zijn.

Tabel 4: Beschrijving methodes LetsGrowWebApiController

Methodes	Uitleg
SetConfiguration(string macAddress, EmbeddedDeviceInformation embeddedDeviceConfig)	Deze methode is verantwoordelijk voor het sturen van de configuratie informatie van het apparaat naar de LetsGrow.com server.
CheckDeviceConfiguration(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van de configuratie informatie van het apparaat van de LetsGrow.com server.
GetDataRecord(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van een lijst van het type “data record” van de LetsGrow.com server.
SetDataRecord(string macAddress, IEnumerable embeddedDeviceDataRecord)	Deze methode is verantwoordelijk voor het sturen van een lijst van het type “data record” naar de LetsGrow.com server.
GetDeviceValidation(string macAddress)	Deze methode is verantwoordelijk voor het valideren van een device door middel van een MAC-adres bij de LetsGrow.com server.

Klassendiagram LetsGrow.com Web API server applicatie

Het server gedeelte is een uitbreiding van de bestaande LetsGrow Web API applicatie. Deze Web API was gebaseerd op ASP.NET Web API[Microsoft, 2016a]. De afstudeerder had eerst deze applicatie bestudeerd. Uit een korte analyse kwam eruit dat deze applicatie ook uit verschillende lagen waren opgebouwd. Models map bevatte verschillende modellen. Deze hielden de data in de applicatie in. Controllers mapje zorgde ervoor dat de 'HTTP' verzoeken behandeld werden door de verschillende controllers. Het Classes mapje bezat de "Business logic".

Dit klassendiagram is ook op soortgelijke manier samengesteld. Daarnaast waren de klassen beschreven en de methodes ook uitgelicht wat ze allemaal voor dienden.

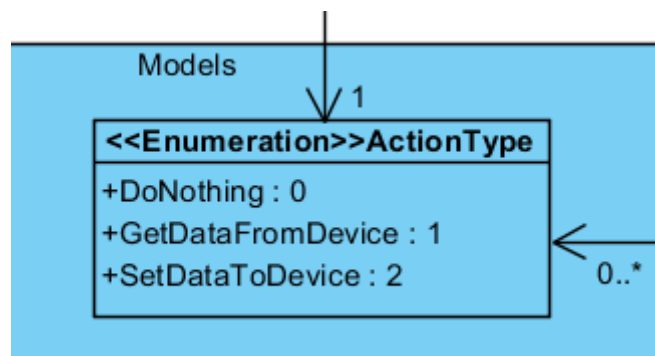
Dit klassendiagram is te vinden in de bijlage G op bladzijde 60 van dit verslag.

7 Ontwikkelfase

In de ontwikkelfase is het architectuur van het ontwerp gerealiseerd. Daarnaast zijn de UML elementen van de definitiefase verder uitgewerkt. Nadat deze afgerond zijn, is er begonnen met het realiseren van de applicaties. Deze applicaties zijn dan 'intern' door de afstudeerder getest en tevens is er een testplan opgesteld om de gehele systeem te laten testen door een 'externe' tester. Uiteindelijk is er ook een handleiding samengesteld en zijn de benodigde documenten en prototype opgeleverd. De gehele ontwerp van de applicaties zijn gebaseerd op het ontwerprapport die in de bijlage K op bladzijde 154 te vinden is.

7.1 Ontwerp architectuur

De UML elementen van de vorige fase zijn in deze fase door de afstudeerder allemaal 'definitief' afgerond en tevens zijn er ook sequentiediagrammen opgesteld. Sommige zaken aan de cliënt klassendiagram en server klassendiagram moesten worden gewijzigd. Deze wijzigingen waren meer bedoeld om de bruikbaarheid en de overzichtelijkheid te verbeteren. Er is bijvoorbeeld in allebei de applicaties in de 'Models' laag de klasse ActionType (Enumeratie) bijgekomen. Deze overweging was eerst bij de definitiefase ook bedacht alleen toen was die er uit gelaten. Later is het toch weer ingezet, omdat de code er overzichtelijker van werd. Daarnaast is ook een klasse van benaming veranderd om de verwarringen uit de weg te ruimen. Deze verwarring kon ontstaan, omdat er in LetsGrow Web API server applicatie een soortgelijke benaming al bestond. Het waren kleine veranderingen en dit hadden meer invloed op de overzichtelijkheid dan op de functionaliteit.



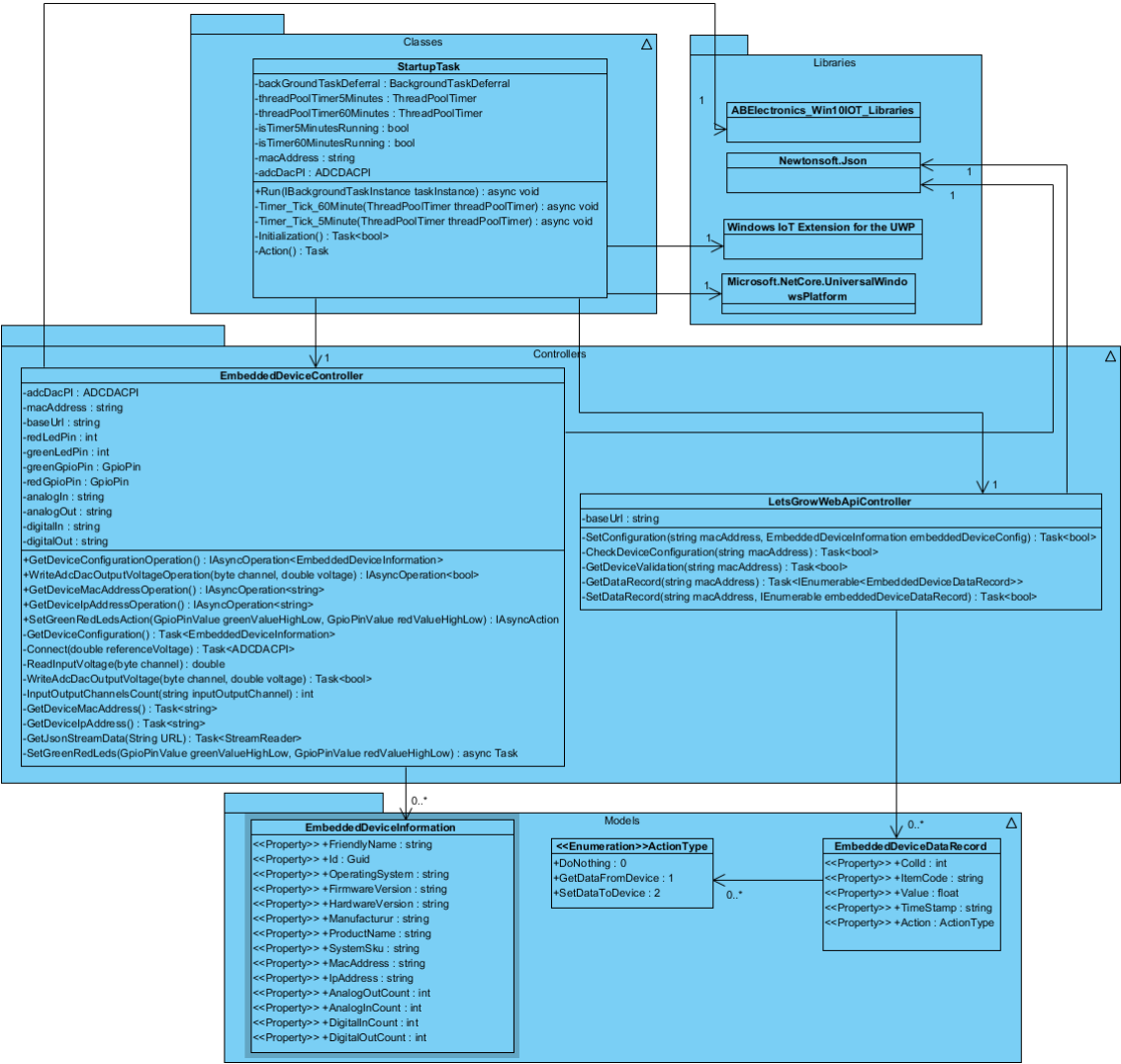
Figuur 14: ActionType klasse

Op de volgende pagina in figuur 15 is het uiteindelijke klassendiagram voor de Pi cliënt applicatie te vinden.

Wegens leesbaarheid is dit klassendiagram opgedeeld in stukken en in de bijlagen D,E en F op de bladzijden 57,58 en 59 bijgevoegd.

Het uiteindelijke klassendiagram voor de LetsGrow Web API server applicatie is in de bijlage H op bladzijde 61 te vinden.

Klassendiagram cliënt applicatie



Figuur 15: Klassendiagram cliënt applicatie

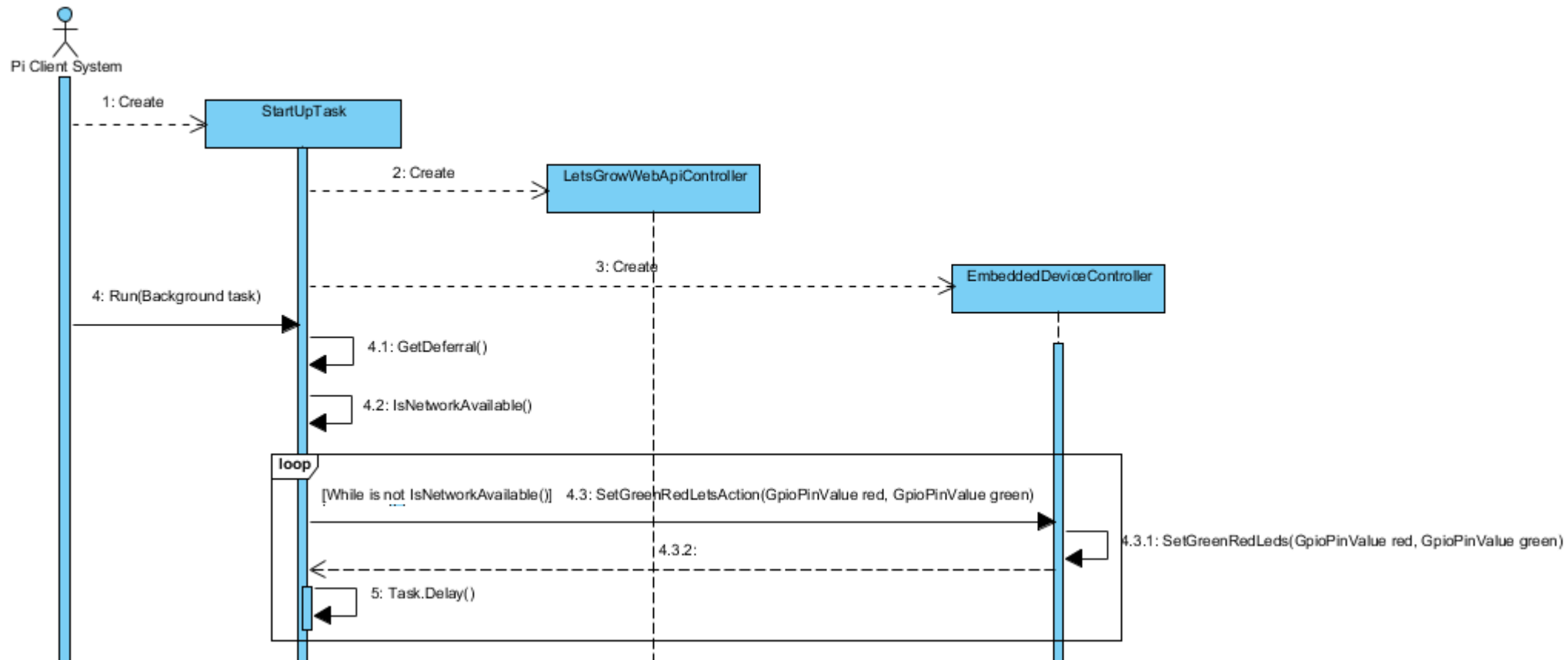
Sequentiediagrammen

In deze fase zijn er sequentiediagrammen opgebouwd om de interactie in de tijd zichtbaar te maken. Normaal is het gewenst om er per use-case een sequentiediagram op te stellen, alleen de bedrijfsmentoren wilden dat de gehele applicatie interactie in één sequentiediagram getoond werd, zodat ze beter overzicht hadden. Hierdoor is het besloten om de use-cases in één sequentiediagram te implementeren.

Op de volgende pagina in figuur 16 wordt een stukje van het sequentiediagram van de Pi cliënt applicatie getoond om te laten zien hoe het sequentiediagram opgebouwd is. De beschrijving die hierbij hoort:

Nadat het Pi cliënt systeem opgestart is, wordt de applicatie automatisch in de achtergrond opgestart. De eerste actie die wordt uitgevoerd is het controleren of er internet toegang aanwezig is. Indien er geen netwerk aanwezig is, dan wordt de applicatie tijdelijk gepauzeerd en wordt op een later tijdstip nogmaals geprobeerd of er netwerktoegang mogelijk is. Als er geen netwerktoegang is wordt het groene en rode ledje van de Pi 2 aangezet (`SetGreenRedLedsAction`) en na 30 seconden (`Task.Delay`) wordt weer gekeken of er een internetverbinding aanwezig is.

De boven behandelde stukje beschrijving is ook gerealiseerd voor alle sequentiediagrammen.

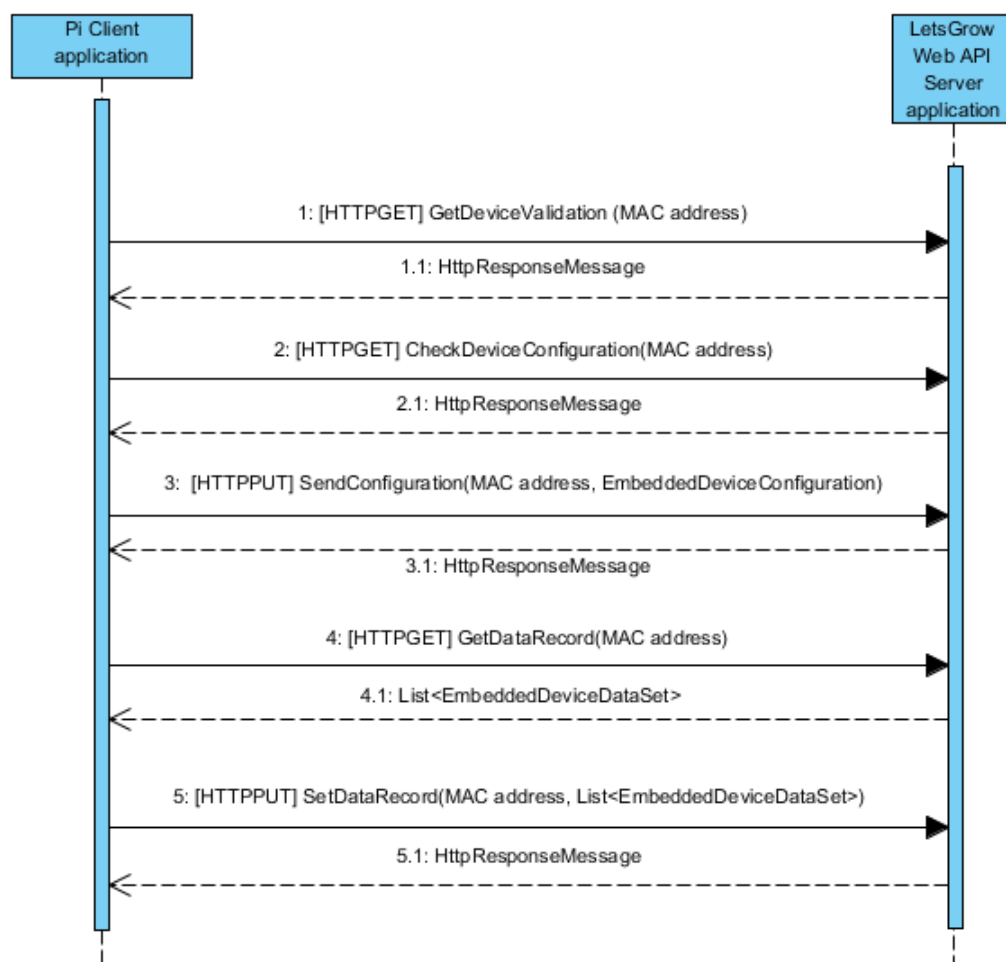


Figuur 16: Use case sequentie 'Check Network Availability' stukje van het sequentiediagram Pi cliënt systeem

Daarnaast was er ook een sequentiediagram (figuur 17) opgesteld om te laten zien hoe de communicatie wordt opgebouwd tussen de Pi cliënt applicatie en de LetsGrow.com Web API server applicatie. Deze communicatie wordt opgebouwd door middel van het REST API protocol. Alternatieven voor de REST API was SOAP en deze is ook bestudeerd door de afstudeerder. De voordelen[Dhingra, 2013] die naar voren kwamen:

- REST is efficiënter dan SOAP (SOAP maakt gebruik van XML voor alle berichten, REST maakt gebruik van kleiner bericht formaten (zoals JavaScript Object Notation (json)).
- Makkelijk te bouwen. Er hoeven voor REST geen speciale 'toolkits' gebruikt te worden.

LetsGrow.com maakt ook zelf gebruik van de REST API en de voorkeur gaat dan ook uit naar het gebruik van deze REST API bij de ontwikkeling van het prototype. Daarom is er voor REST API methode gekozen.



Figuur 17: Sequentiediagram cliënt-server communicatie

7.2 Prototype realisatie

Het realiseren van het prototype gebeurde in meerdere stappen. Deze stappen zullen hieronder beschreven worden.

Eerst is er hardware matig allerlei bekabeling en het ADC-DAC Pi shield door middel van GPIO pinnen op de eerste 8 pinnen van de Raspberry Pi 2 geplaatst. Daarna is er door van IP-adres benadering met het Windows IoT besturingssysteem connectie opgebouwd om 'debug' mode aan te zetten. Uiteindelijk is er op de ontwikkel-pc Visual Studio 2015 editie geopend om applicatie te bouwen tegelijkertijd te debuggen naar de Raspberry Pi 2.

Dit ging helaas niet zonder tegenslagen. De volgende problemen kwamen tijdens het debuggen/bouwen tevoorschijn:

- Visual Studio 2015 debugger werkte eerst niet goed. Er kon geen verbinding gemaakt worden met het prototype. De afstudeerder had met Microsoft contact opgenomen en Microsoft raadde aan om een 'inside preview versie' van het besturingssysteem te gebruiken. Dit had de afstudeerder gedaan en daarna lukte het om succesvol te kunnen verbinden.
- Tijdens het bouwen van de 'achtergrond' applicatie bleek dat in praktijk de 'Headless' mode van Windows IoT onstabiel was. Dit hield in dat er in de achtergrond het proces (`input_service.exe`) elke keer stopte met werken. Dit had niets te maken met de te bouwen applicatie. Door de afstudeerder zijn de error-logs bekeken waarna geconcludeerd kon worden dat het systeem in 'Headless mode' nog steeds zocht naar 'inputs', zoals een HDMI kabel, USB toetsenbord of een muis. Een 'work-around' was om het Windows IoT besturingssysteem te draaien in 'Headed-mode' maar de gebouwde applicatie in de achtergrond(als 'headless') te laten draaien. Uiteindelijk was door middel van deze aanpak het probleem opgelost. Tevens was dit ook de enige oplossing, omdat Microsoft op dat moment geen oplossing had. Deze oplossing is 'gemaild' met de 'error-logs' naar het Windows Dev Center om Microsoft op de hoogte te brengen van de bug met een mogelijke 'work-around'.

Applicatie-code

De applicaties zijn door de afstudeerder gebouwd met C# programmeertaal. Er is door de afstudeerder gekozen om de applicatie duidelijk en volgens de richtlijnen te ontwerpen. Daarnaast is er ook gebruik gemaakt van C# LINQ expressies voor sommige stukken code. Met deze expressies had de afstudeerder weinig ervaring. Het was voor de afstudeer ook beter om door middel dit project nieuwe kennis en ervaring op te doen. Technisch gezien zorgt de LINQ expressies ervoor dat er door middel van 1 regel code een gehele functie uitgevoerd kon worden, in plaats van meerdere regels code. Daarnaast was het belangrijk om exceptie handeling te gebruiken middels 'try-catch' constructies (dit houdt in dat er in de code excepties afgehandeld kunnen worden). De gerealiseerde applicatie moest zonder de tussenkomst van personen kunnen draaien. Hierdoor moest hij indien er fouten optreden, deze fouten goed kunnen afvangen en doorgaan met zijn werking. Hieronder zal interessante code getoond worden:

```

var dataSet = await letsGrowWebApiController.GetDataRecord(macAddress);
var outputList = dataSet.Where(d => d.Action == ActionType.SetDataToDevice).ToList();
var inputList = dataSet.Where(d => d.Action == ActionType.GetDataFromDevice).ToList();

if (outputList.Count != 0)
{

```

Figuur 18: Applicatie code: LINQ expressies

De in bovenstaand figuur 18 afgebeelde code laat de gebruikte LINQ expressie zien. Aan de hand van 1 regel code kon gelijk de actie bepaald worden en een lijst met desbetreffende gegevens samengesteld worden. Deze lijst kon dan weer verder behandeld worden om de ontvangen gegevens te kunnen wegschrijven naar de desbetreffende kanalen. Deze wordt in de volgende figuur getoond:

```

foreach (var data in outputList)
{
    //Math.Max(First value is max zero, second value length)
    //Math.Round(First is value in double and second is the number of digits after the decimal place).
    var channel = data.ItemCode.Substring(Math.Max(0, data.ItemCode.Length - 2));
    var succesfullyWriteOutputVoltage =
        await embeddedDeviceController.WriteAdcDacOutputVoltageOperation(Convert.ToByte(channel),
            Math.Round(data.Value, 4));
}

```

Figuur 19: Applicatie code: Afhandeling actie

De bovenstaand figuur 19 laat zien hoe door middel van de functie Math.Max de desbetreffende kanaal nummering verkregen kan worden. Deze wordt dan ook met de desbetreffende analoge waarde in 4 decimalen achter de komma wegeschreven naar de ADC-DAC Pi shield uitgangen.

De onderstaand figuur 20 laat de methode voor het opvragen van het apparaat validatieproces zien. Er is voor deze methode gebruik gemaakt van de eerder genoemde 'try-catch' constructie. Dit houdt dat als er een fout optreedt bij de validatie dat deze dan wordt afgevangen, wordt gelogd en verder wordt doorgegeven.

```

internal async Task<bool> GetDeviceValidation(string macAddress)
{
    string url = string.Format("{0}/api/EmbeddedDevice/{1}/Actions/GetDeviceValidation", baseUrl, macAddress);
    var request = (HttpRequest)WebRequest.Create(new Uri(url));
    request.Method = "GET";
    request.Accept = "application/json";
    request.ContentType = "application/json";

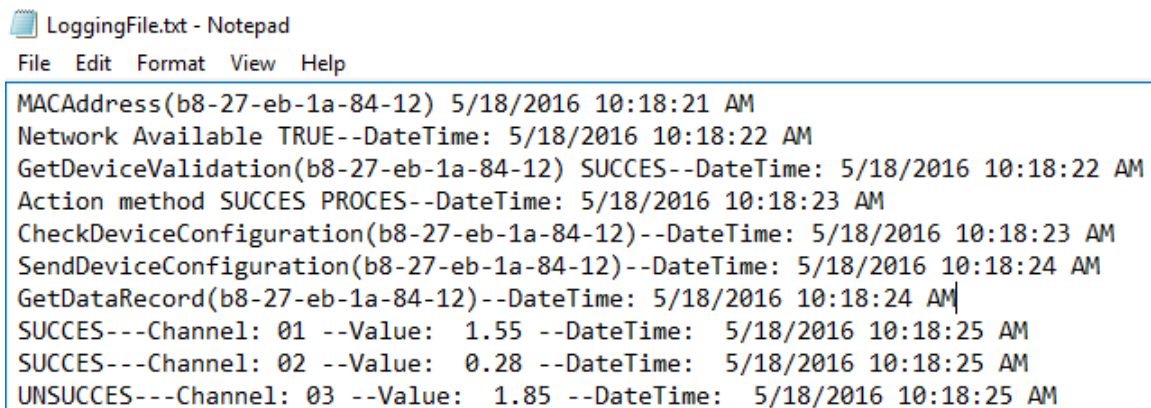
    try
    {
        HttpResponseMessage httpResponse = (HttpResponseMessage)await request.GetResponseAsync();
        if (httpResponse.StatusCode == HttpStatusCode.OK)
            return true;
        return false;
    }
    catch (Exception ex)
    {
        embeddedDeviceController.WriteLoggingToFile(ex.ToString());
        throw new SecurityException("Failed device validation", ex);
    }
}

```

Figuur 20: Applicatie code: 'try-catch' constructie

7.3 Testen van het prototype

Het testen van het prototype is gebeurd in 3 stappen. De eerste stap was dat de afstudeerder per gebouwde functie in de code (inline debugging) ging testen of de uitkomst ook daadwerkelijk goed was waarvoor hij gebouwd was. Zo werd de code stuk voor stuk getest aan de hand van de specificaties in het ontwerp. De tweede stap was dat de afstudeerder een 'logging' functie inbouwde in de applicatie. Deze hield in dat per uitgevoerde functie in een tekst bestand weggeschreven werd wanneer het uitgevoerd was en of het succesvol uitgevoerd was. Hierdoor wist de afstudeerder precies welke tijdstip welke functie uitgevoerd werd en of het ook daadwerkelijk succesvol uitgevoerd was. De afstudeerder heeft het prototype anderhalve week lang getest door middel van deze 'logging' functie. Hieronder is een stukje van een logbestand afgebeeld ter illustratie van de opbouw van de 'logging' bestand.



```
MACAddress(b8-27-eb-1a-84-12) 5/18/2016 10:18:21 AM
Network Available TRUE--DateTime: 5/18/2016 10:18:22 AM
GetDeviceValidation(b8-27-eb-1a-84-12) SUCCES--DateTime: 5/18/2016 10:18:22 AM
Action method SUCCES PROCES--DateTime: 5/18/2016 10:18:23 AM
CheckDeviceConfiguration(b8-27-eb-1a-84-12)--DateTime: 5/18/2016 10:18:23 AM
SendDeviceConfiguration(b8-27-eb-1a-84-12)--DateTime: 5/18/2016 10:18:24 AM
GetDataRecord(b8-27-eb-1a-84-12)--DateTime: 5/18/2016 10:18:24 AM
SUCCES---Channel: 01 --Value: 1.55 --DateTime: 5/18/2016 10:18:25 AM
SUCCES---Channel: 02 --Value: 0.28 --DateTime: 5/18/2016 10:18:25 AM
UNSUCCES---Channel: 03 --Value: 1.85 --DateTime: 5/18/2016 10:18:25 AM
```

Figuur 21: Applicatie: Logging bestand

Het bovenstaande figuur 21 laat zien dat er in elke functie 'logging' ingebouwd was om het prototype uitgebreid te kunnen testen. In het logbestand zijn tijdsaanduidingen opgenomen om hiermee te kunnen bepalen hoe lang de applicatie doorbrengt in de verschillende functies. Dit proces heeft de gedurende anderhalve week met succes gedraaid. Note: In het bovenstaande figuur kan er gezien worden dat 'Channel 3' niet succes is. Dit klopt ook, omdat het shield twee analoge uitgangen had. Alleen was het wel handig om te kijken wat er zou gebeuren met meer dan twee waardes tegelijk.

De derde stap was dat het prototype door middel van 'black box' testmethode door de tester van LetsGrow.com te laten testen. Deze methode is gekozen, omdat de tester bijna helemaal niets wist van de interne structuur van de applicatie. Tevens had deze tester geen toegang tot de code zelf en was 'black box' testen beter geschikt dan 'white box' testen. De bedrijfsmentor reviewde en testte de code aan de hand van zijn test regels. Hij deed dus een soort 'white box' testen. Tevens had de afstudeerder al een soort 'white box testen' gedaan. Het was dus beter om het prototype met verschillende test methodes te testen. Op de volgende pagina is er in het figuur 22 een voorbeeld van een 'black box' testgeval uitgebeeld die door de tester uitgevoerd is.

7.4 Handleiding prototype

Voordat het project afgesloten kon worden, moest ook een handleiding opgeleverd worden voor de servicemonteur. Dit is dan ook gerealiseerd. In de bijlage R op bladzijde 198 is de handleiding te vinden.

Minstens een analoge waarde kunnen sturen naar de klimaatcomputer.

De volgende eisen worden met dit testgeval getest:

Het prototype moet minstens een analoge waarde kunnen sturen naar een klimaatcomputer

Pre condities	Test-scenario	Beschrijving	Uitvoer-voorspelling	Werkelijke uitvoer	Opmerking
<ul style="list-style-type: none"> • Het prototype is opgestart. • LetsGrow.com Web API server applicatie is opgestart. • SQL Server 2014 opgestart en de benodigde tabellen zijn geopend. • Tester heeft de Hoogendoorn klimaatcomputer voor zich met de Hoogendoorn software. • De sensor-metingen tabblad is geopend en zichtbaar voor de tester. 	Het prototype moet minstens een analoge waarde kunnen sturen naar een klimaatcomputer	<p>Het prototype gaat een analoge waarde naar de klimaatcomputer sturen. Het prototype gaat het groene ledje laten branden.</p> <p>De tester ziet aan de hand van Hoogendoorn Klimaatcomputer (tabblad sensor-metingen) dat er een analoge waarde het systeem binnen is gekomen.</p>	Het prototype heeft succesvol een analoge waarde gestuurd naar de klimaatcomputer. Het prototype heeft het groene ledje laten branden.		Dit testgeval kan ook op de volgende manier getest worden. Door middel van een multimeter kan er aan de 1 ^e en 2 ^e analoge uitgangen van het shield gemeten worden of er ook daadwerkelijk de desbetreffende spanning waar te nemen is.

Figuur 22: Testgeval

8 Evaluatie

In deze hoofdstuk zal ik de gehele afstudeerperiode evalueren.

8.1 Producten

8.1.1 Plan van Aanpak

Het Plan van Aanpak (Pva) was van te voren ingepland om opgeleverd te worden. Het voldeed aan de verwachtingen van zowel de opdrachtgever als de bedrijfsmentoren. Ze hadden dankzij dit document een duidelijk beeld over hoe het project gerealiseerd zou worden. Uiteindelijk is dit document met succes opgeleverd.

8.1.2 Definitierapport

Het definitierapport was ook een van de rapporten die ik van te voren had ingepland om op te leveren. Dit is dan ook met succes tijdens het definitiefase opgeleverd. Doel van dit rapport was om de opdrachtgever en de bedrijfsmentoren te informeren over de analyse die ik gedaan had. Laatste stukje van dit rapport bevatte systeemarchitectuur ontwerpen. Al deze zaken waren opgeleverd en voldeed aan de verwachtingen van zowel de opdrachtgever als de bedrijfsmentoren.

Wat ik minder goed heb gedaan is om het rapport niet op te splitsen in twee delen. Het definitierapport bestaat nu uit twee gedeeltes: het eerste gedeelte is de analyse en het tweede gedeelte het systeemontwerp. Dit was dan overzichtelijker voor zowel de bedrijfsmentor als de opdrachtgever. Uiteindelijk maakte het inhoudelijk niets uit, omdat dit rapport alle informatie bevat wat ook de analyse en architectuurdocument zou bevatten.

8.1.3 Ontwerprapport

Het ontwerprapport bevatte het gehele ontwerp van het prototype en de LetsGrow.com Web API server applicatie. Er was eerst per use-case een sequentiediagram opgesteld, alleen de bedrijfsmentor wilde één sequentiediagram per applicatie zien, omdat de functies en methodes niet ingewikkeld waren. Dankzij deze aanpak kon hij de 'flow' van de applicatie beter zien. Dit is dan ook aangepast. Uiteindelijk is het ontwerprapport met succes afgerond en opgeleverd. Dit rapport voldeed tevens aan de verwachtingen van zowel de opdrachtgever als de bedrijfsmentoren. Dankzij dit rapport wisten de bedrijfsmentoren gelijk hoe het gehele prototype in elkaar zat.

8.1.4 Testplan

Het testplan was samengesteld om het prototype te kunnen testen. Er is eerst door mij code-technisch getest. Daarna was door de bedrijfsmentor de gehele code doorgenomen, gereviewd en ook getest. Alleen deze testen waren meer 'white-box-testen' perspectief en moest het prototype ook extern getest worden door de officiële tester van LetsGrow.com. Dit rapport was voor hem bedoeld en dankzij dit rapport heeft hij het prototype kunnen testen op de functionaliteiten. Uiteindelijk is er uit de tests geen problemen naar voren gekomen.

8.1.5 Handleiding

De handleiding was voor de servicemonteur samengesteld zodat hij hiermee het prototype aan een klimaatcomputer kon aansluiten. De handleiding is eveneens getest door er een servicemonteur mee aan het werk te zetten. De praktijk leerde dat de handleiding voldeed en alle informatie bevatte om een succesvolle aansluiting te kunnen doen.

8.1.6 Prototype

Het prototype bestond uit 2 delen. Het eerste gedeelte bestond uit de hardware componenten. Deze hardware componenten hebben uiteindelijk voldaan aan de wensen van zowel de opdrachtgever als de bedrijfsmentoren. Al deze hardware componenten waren op tijd besteld en ook door mij in elkaar gezet. De gehele project is zorgvuldig omgegaan met de hardware van het prototype. Het prototype werd na gebruik veilig opgeborgen in een kast.

Het tweede gedeelte bestond uit de realisatie van de Pi cliënt applicatie (prototype applicatie) en de LetsGrow.com Web API server applicatie. Deze heb ik met succes gerealiseerd en opgeleverd. De opgeleverde applicaties werden gereviewed door de bedrijfsmentor, omdat procedure binnen LetsGrow.com is dat alle code eerst wordt gereviewed door een collega programmeur vervolgens wordt de applicatie gebouwd en wordt deze getest door de tester van LetsGrow.com. Bij reviews werd er gelet op potentiële fouten, code constructies, style, code-guidelines en ook ifSQ richtlijnen. Uiteindelijk is er uit deze reviews en testen goede resultaten naar voren gekomen.

Het prototype is anderhalve week lang getest en kwam met succes de duurttest uit. De probleem van de opdrachtgever is met het prototype opgelost en de doelstelling voor het project is dan ook behaald. Indien ze willen kunnen ze dit prototype bij de klant verder gaan inzetten voor verschillende toepassingen.

8.2 Proces

Om het gehele proces te kunnen beschrijven ga ik eerst beginnen met de gekozen methodiek. In de praktijk is gebleken dat de gekozen waterval methodiek inderdaad de juiste aanpak was voor dit project. De behoefte van de opdrachtgever is gedurende het project stabiel gebleven. De waterval methode legt de nadruk op documentatie wat in het kader van mijn afstuderen ook goed uitkwam. Door gebruik te maken deze methode was voor mij en ook voor het afstudeerstagebedrijf de voortgang goed te monitoren. Qua voortgang weet je precies waar je zit en wat je nog moet doen en hoeveel tijd je nog hebt. Het proces wordt hierdoor overzichtelijker. De fasering bleek in praktijk dat het voor dit project ook erg handig was. Eerst moest er grondig een analyse gedaan worden, voordat er begonnen kon worden met het ontwerpen.

Als ik in de toekomst weer te maken krijg met een soortgelijk project, dan zou ik deze aanpak weer met plezier toepassen, omdat het voor dit soort projecten geschikt is.

8.2.1 Oriëntatiefase:

De oriëntatiefase was belangrijk om te de opdracht te verhelderen. Het inzicht is onder andere verkregen door middel van gesprekken met de opdrachtgever. Tevens is in deze fase de risicoanalyse opgesteld en een wekelijkse planning gemaakt. De oriëntatiefase liep volgens de verwachtingen en het verwachte deelproduct (Pva) zijn ook in deze fase succesvol gerealiseerd.

8.2.2 Definitiefase:

De definitiefase was de rode draad voor mijn project. In deze fase had ik eerst door middel van benodigde (relevante) literatuur mijn kennis verbreed. Dit was benodigd om meer kennis te krijgen over bepaalde zaken, zodat deze kennis later in het project meegenomen kon worden. Daarna heb ik door middel van een hoofdvraag een analyse uitgevoerd om te kijken of het wel mogelijk was om door middel van het prototype de klimaatcomputer regelingen te kunnen beïnvloeden. Na deze uitgebreide analyse heb ik de functionele eisen en niet functionele eisen achterhaald. Daarna ben ik in een valkuil gestapt. Ik had deze eisen omgezet naar user-stories, maar dit hoefde helemaal niet. User-Stories worden veel gebruikt voor 'Agile' methoden, terwijl

ik geen 'Agile' methode hebt gebruikt. Dit heb ik ook weer snel teruggedraaid. Uiteindelijk heb ik ook de systeemarchitectuur opgesteld. Al de genoemde zaken voldeden aan de verwachtingen van de opdrachtgever.

8.2.3 Ontwikkelfase:

De ontwikkelfase was de laatste fase voor het project en tevens ook de fase waarin het prototype gerealiseerd en uitgebreid getest werd. Het bouwen van de applicatie en het debuggen op de Raspberry Pi 2 bleek geen makkelijke opgave. De eerste paar pogingen zijn dan ook mislukt. Hiervoor had ik met Microsoft contact opgenomen om dit op te kunnen lossen. Daarna was dit probleem opgelost. Het testen van het prototype verliep goed. Het prototype is door mij anderhalve week lang getest en kwam met succes uit de duurttest. Daarnaast heeft de tester van LetsGrow.com het product getest. Deze fase verliep volgens de planning en uiteindelijk voldeden de gerealiseerde tussen- als eindproducten aan de verwachtingen.

8.3 Competenties

De onderstaande competenties zijn door mij uitgevoerd volledig uitgevoerd en met succes afgerond. Eerst heb ik alle competenties doorgenomen en gekeken wat de criteriums waren. Deze criteriums (die te vinden is op het Blackboard portal van de HHS, in de course 'Afstuderen Faculteit IT & Design' onder het mapje 'Gemeenschappelijk voor deeltijd/volgtijd', onder hoofdstuk 'Beroepstaken' link: HTI wiki 2014-2015.html) zijn ook uitgevoerd om te voldoen aan de competenties.

- G1: Praktische aspecten hanteren in (internationale) projecten.

Deze competentie is door mij volledig uitgevoerd en met succes afgerond. Er is door mij een opdrachtoomschrijving samengesteld, een fasering opgesteld en een risicoanalyse opgesteld. Al deze zaken zijn beschreven in dit verslag in hoofdstuk 4 aanpak en hoofdstuk 5 oriëntatiefase.

- A1: Analyseren van het probleemdomain

Deze competentie is door mij volledig uitgevoerd en met succes afgerond. In de oriëntatiefase heb ik eerst een probleemanalyse uitgevoerd aan de hand van de 6-W formule[Verhoeven, 2010]. Daarna heb ik in de definitiefase eerst de huidige situatie geanalyseerd. Daarnaast heb ik door middel van een hoofdvraag geanalyseerd hoe de regelingen in klimaatcomputers beïnvloed kunnen worden, zonder de tussenkomst van de betreffende klimaatcomputer leveranciers. Al deze zaken zijn beschreven in hoofdstuk 5 oriëntatiefase en hoofdstuk 6 definitiefase.

- C10: Ontwerpen van een systeemarchitectuur

In de definitiefase is het systeemarchitectuur ontwerp gerealiseerd. Dit ontwerp is gedaan aan de hand van UML modelleertaal[Warmer and Kleppe, 2011]. Er is rekening gehouden met de architectuur, zodat hij passend was voor de gehele project. Daarnaast zijn ook alle klassen beschreven en is bij de keuzes nagedacht over de toekomstvastheid van de opdrachtgever. Al deze zaken zijn beschreven in het tweede gedeelte van hoofdstuk 6 definitiefase. Deze competentie is door mij volledig uitgevoerd en met succes afgerond.

- C8: Ontwerpen van een technisch informatiesysteem

In de ontwikkelfase is voortgegaan vanuit de vorige competentie (C10) en is het gehele systeem technisch ontworpen. Al deze zaken zijn beschreven in het tweede gedeelte van hoofdstuk 6 definitiefase en hoofdstuk 7 ontwikkelfase beschreven. Deze competentie is door mij volledig uitgevoerd en met succes afgerond.

- D17: Testen van softwaresystemen

Er is voor de tester een testplan opgesteld. Aan de hand van dit plan heeft de tester het gehele prototype kunnen testen op functionaliteiten. Daarnaast heb ik ook zelf een 'interne test' uitgevoerd. Deze competentie is door mij volledig uitgevoerd en met succes afgerond. Het uiteindelijke testrapport is in de bijlage L op bladzijde 173 te vinden.

8.4 Kwaliteitsniveau project

Uiteindelijk heb ik aan de hand van het Competentiemodel Academie ICT en Media September 2006 document (die te vinden is op het Blackboard portal van de HHS, in de course 'Afstuderen Faculteit IT & Design' onder het mapje 'Gemeenschappelijk voor deeltijd/voltijd', onder hoofdstuk 'Beroepstaken') het kwaliteitsniveau van mijn project bepaald.

Eerst heb ik de volgende stellingen met een ja beantwoord:

- Je werkt onder een behoorlijke tijdsdruk.
- Je hebt meerdere (informele) opdrachtgevers die allen andere wensen/eisen kunnen hebben.
- Er zullen veel mensen te maken krijgen met mijn werk.
- Er zal veel veranderen op de afdeling/bij de klant als mijn werk af is.
- De resultaten van mijn opdracht heeft het management nodig om te beslissen hoe het in de toekomst verder moet.

Hierboven zijn er 5 stellingen met een 'ja' beantwoord. Hierdoor is de indicatie van de moeilijkheidsgraad 'complex'. Daarna heb ik gekeken bij de taakrol. Ik had de gehele afstudeerproject in mijn eentje, zelfstandig uitgevoerd. Dus mijn taakrol was: zelfstandig. Aan de hand van de onderstaande tabel 5 komt de volgende uit:

Tabel 5: Matrix Context/Taakrol

		Taakrol		
		Geleid	Zelfstandig	Sturend
Context	Simpel	1	2	3
	Lastig	2	3	4
	Complex	3	4	5

Het gehele afstudeerproject is uitgevoerd op een niveau van 4.

8.5 Doel en Probleem

Concluderend kan ik zeggen dat het probleem is opgelost en het doel gehaald is. Het gerealiseerde prototype en bijbehorende applicaties hebben voor gezorgd dat LetsGrow.com nu een oplossing heeft om de klimaatcomputers regelingen kunnen beïnvloeden, zonder de tussenkomst van de klimaatcomputer leveranciers. LetsGrow.com kan nu los gaan en als ze het willen, kan het prototype ook ingezet worden bij de klant.

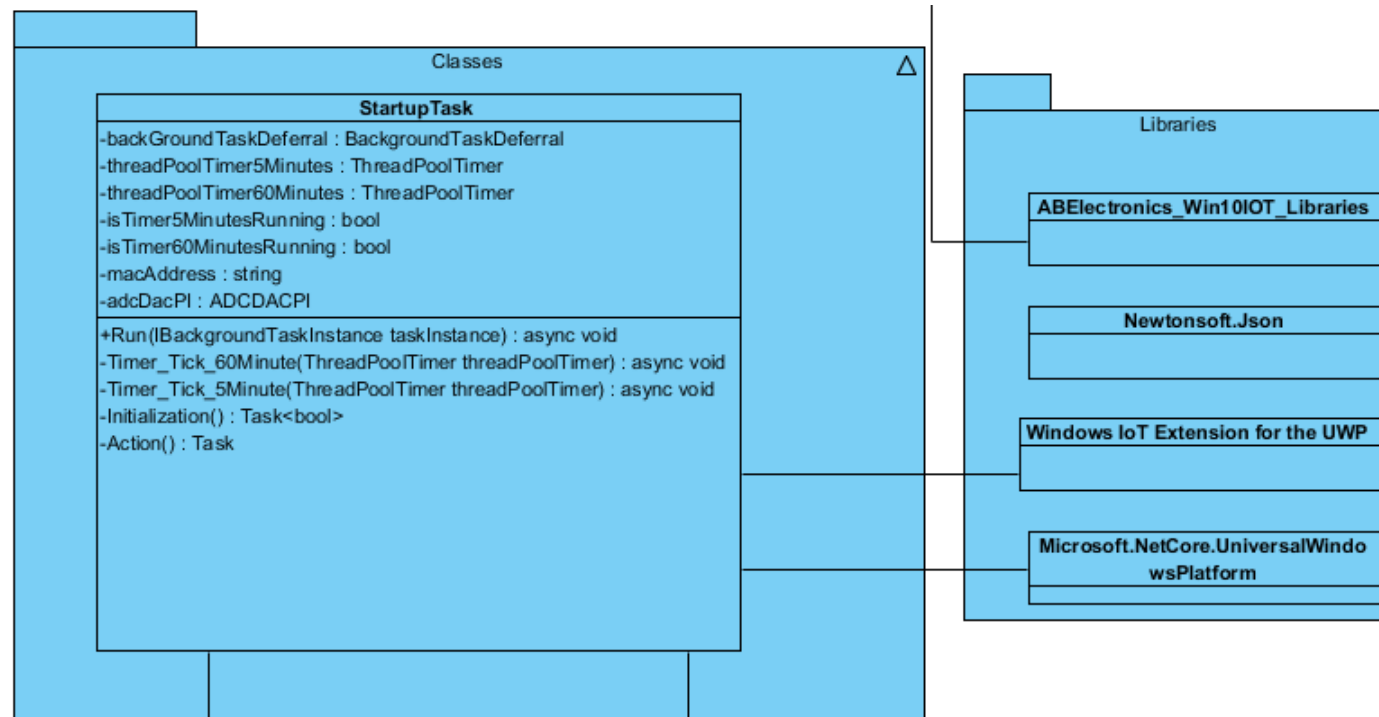
Referenties

- [ADC-DAC-PI-shield, 2016] ADC-DAC-PI-shield, A. (2016). ADC-DAC Pi shield. Website geraadpleegd op 2 mei 2016, <https://www.abelectronics.co.uk/p/39/ADC-DAC-Pi-Raspberry-Pi-ADC-and-DAC-expansion-board>.
- [Cooking-Hacks, 2016] Cooking-Hacks (2016). 4-20 mA Sensor Board tutorial for Arduino, Raspberry Pi and Intel Galileo. Website geraadpleegd op 2 mei 2016, <https://www.cooking-hacks.com/documentation/tutorials/4-20ma-sensor-board-shield-tutorial-for-arduino-raspberry-pi-intel-galileo/>.
- [Dhingra, 2013] Dhingra, S. (2013). Rest vs soap: How to choose the best web service. Website geraadpleegd op 14 maart 2016, <http://searchsoa.techtarget.com/tip/REST-vs-SOAP-How-to-choose-the-best-Web-service>.
- [Expander-Pi-shield, 2016] Expander-Pi-shield, A. (2016). Expander Pi shield. Website geraadpleegd op 2 mei 2016, <https://www.abelectronics.co.uk/p/50/Expander-Pi>.
- [Foundation, 2015a] Foundation, R. P. (2015a). Faq: 4. what hardware does it have? Website geraadpleegd op 2 maart 2016, <https://www.raspberrypi.org/help/faqs/#performanceInterfaces>.
- [Foundation, 2015b] Foundation, R. P. (2015b). The Magpi Magazine issue 31; Raspberry Pi 2. (31):70.
- [Foundation, 2015c] Foundation, R. P. (2015c). The Magpi Magazine issue 34; Windows 10 on Raspberry Pi. (34):70.
- [Foundation, 2015d] Foundation, R. P. (2015d). The Magpi Magazine issue 35; Amazing Pi Projects. (35):70.
- [Foundation, 2016] Foundation, R. P. (2016). Rasbian Frontpage. Website geraadpleegd op 2 mei 2016, <https://www.raspbian.org/>.
- [Gamma et al., 1994] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, Boston, 1 edition.
- [Harrington, 2015] Harrington, W. (2015). *Learning Raspbian*. Packt Publishing, Birmingham, 1 edition.
- [Hoogendoorn, 2016] Hoogendoorn (2016). Hoogendoorn Growth Managment. Website geraadpleegd op 14 februari 2016, <http://www.hoogendoorn.nl/over-ons>.
- [Jimb0, 2015] Jimb0 (2015). What is a shield? Website geraadpleegd op 2 maart 2016, <https://learn.sparkfun.com/tutorials/arduino-shields>.
- [Laukien, 2015] Laukien, D. (2015). Open Source Expansion Board For Raspberry Pi. Website geraadpleegd op 2 mei 2016, <http://www.sf-innovations.co.uk/custard-pi-2.html>.
- [LetsGrow.com, 2015] LetsGrow.com (2015). Letsgrow.com. Website geraadpleegd op 14 februari 2016, <http://www.letsgrow.com/nl/wat-is-letsgrowcom>.
- [Microsoft, 2016a] Microsoft (2016a). Learn About ASP.NET Web Api. Website geraadpleegd op 20 april 2016, <http://www.asp.net/web-api>.
- [Microsoft, 2016b] Microsoft (2016b). Windows IoT Page. Website geraadpleegd op 2 mei 2016, <https://developer.microsoft.com/en-us/windows/iot>.

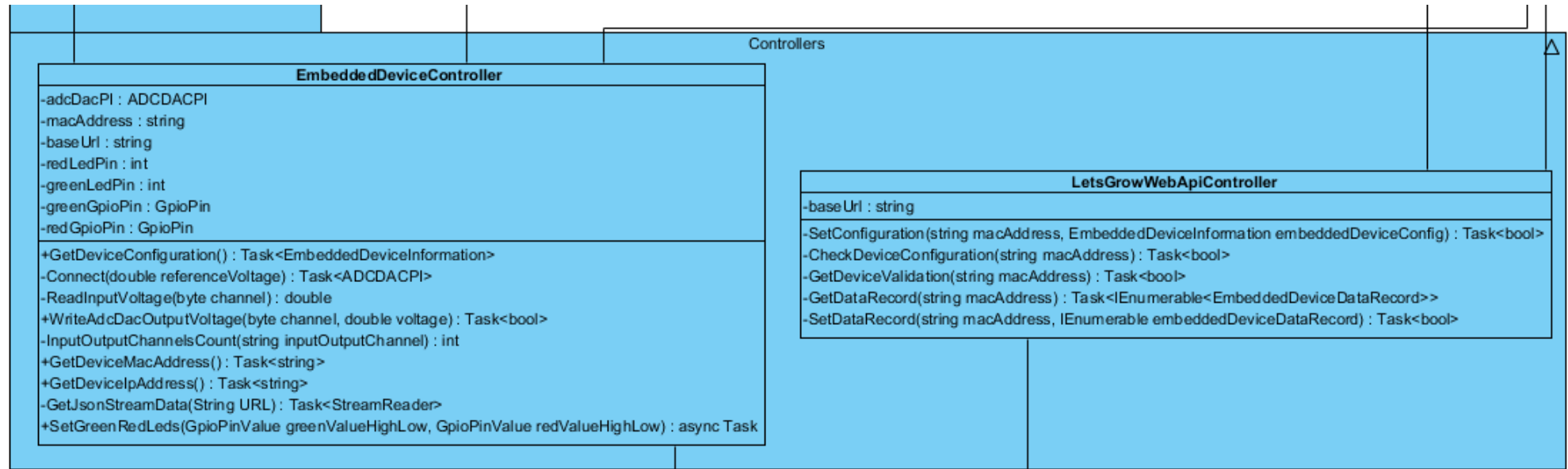
- [Mikroe.com, 2016] Mikroe.com (2016). Mikroe Elektronika 4-20 ma t click shield. Website geraadpleegd op 2 mei 2016, <http://www.mikroe.com/click/4-20ma-t/>.
- [Phidgets.com, 2012] Phidgets.com (2012). Phidgetanalog 4-Output. Website geraadpleegd op 2 mei 2016, http://www.phidgets.com/products.php?product_id=1002.
- [Sargent,] Sargent, J. K. *Windows 10 for the Raspberry Pi 2: Getting Started with Windows 10: Beginners guide to getting started with Windows 10 for the Raspberry Pi 2*. Amazon Digital Services LLC, Seattle.
- [Schipper, 2015] Schipper, J. (2015). *Commissioning values various sensor*. Hoogendoorn Vlaardingen. Geraadpleegd op 9 maart 2016, Interne PDF documentatie, <https://...../Products/Commissioningvaluesvariousensenso.pdf>.
- [sf innovations, 2016] sf innovations (2016). Custard Pi 2. Website geraadpleegd op 2 mei 2016, <http://www.sf-innovations.co.uk/custard-pi-2.html>.
- [Soy, 2016a] Soy, E. (2016a). Gesprek met Dhr. Peter Hendriks over de afstudeeropdracht. Geraadpleegd op 13 februari 2016.
- [Soy, 2016b] Soy, E. (2016b). Gesprek met Dhr.Christiaan Posthumus Meyjes over de regelingen en klimaatcomputers. Geraadpleegd op 13 maart 2016.
- [Soy, 2016c] Soy, E. (2016c). Gesprek met Dhr.Edwin Vis over de communicatie. Geraadpleegd op 13 maart 2016.
- [Soy, 2016d] Soy, E. (2016d). Interview Dhr.Jack Schipper over werking van de klimaatcomputers. Geraadpleegd op 17 maart 2016, Klimaatcomputers.
- [Soy, 2016e] Soy, E. (2016e). Vergadering met de bedrijfsmentoren over Eisen/Wwensen. Geraadpleegd op 7 april 2016, Klimaatcomputers.
- [Technology, 2016] Technology, U. (2016). UniPi Board Technology. Website geraadpleegd op 2 mei 2016, <http://unipi.technology/product/unipi/>.
- [van Houtum, 2013] van Houtum, I. (2013). Agile versus traditionele waterval aanpak. Website geraadpleegd op 15 februari 2016, <http://www.iep.nu/2013/09/11/agile-versus-traditionele-waterval-aanpak/>.
- [van Linden, 2014] van Linden, A. (2014). *Construction EtherCan*. Hoogendoorn Vlaardingen. Geraadpleegd op 9 maart 2016, Interne PDF documentatie, <https://...../Products/ConstructionEtherCan.pdf>.
- [Verhoeven, 2010] Verhoeven, N. (2010). *Wat is onderzoek*. Boom Lemma Uitgevers, Amsterdam, 5 edition.
- [Vis, 2016] Vis, E. (2016). Interface description letsgrow web api (version 1.5). Geraadpleegd op 14 maart 2016, <http://api.letsgrow.com/Download/LetsGrowWebAPI.pdf>.
- [Warmer and Kleppe, 2011] Warmer, J. and Kleppe, A. (2011). *Praktisch UML*. Pearson Benelux B.V., Amsterdam, 5 edition.

Bijlagen

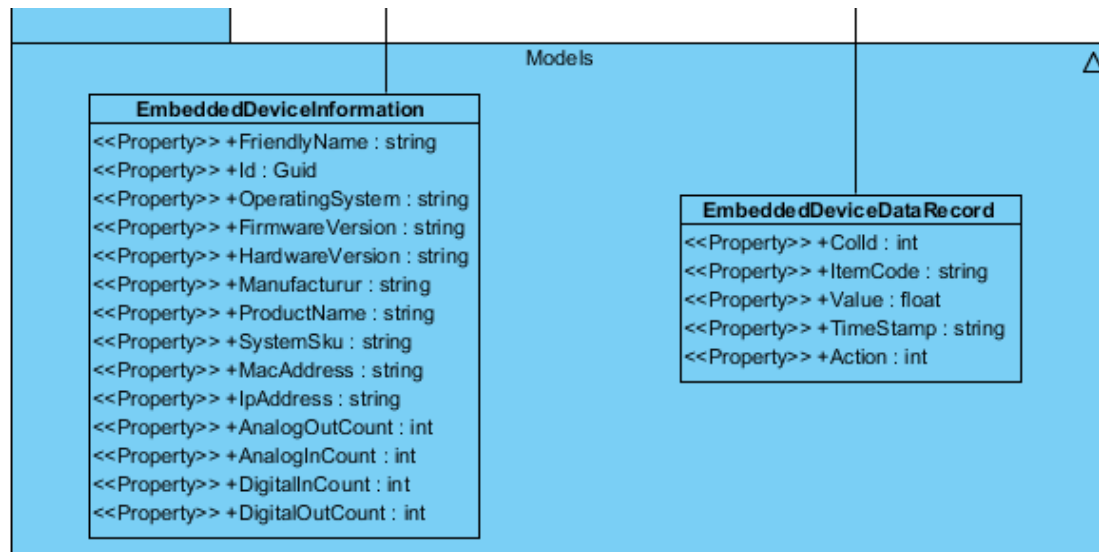
A Klassendiagram Clïent applicatie deel 1



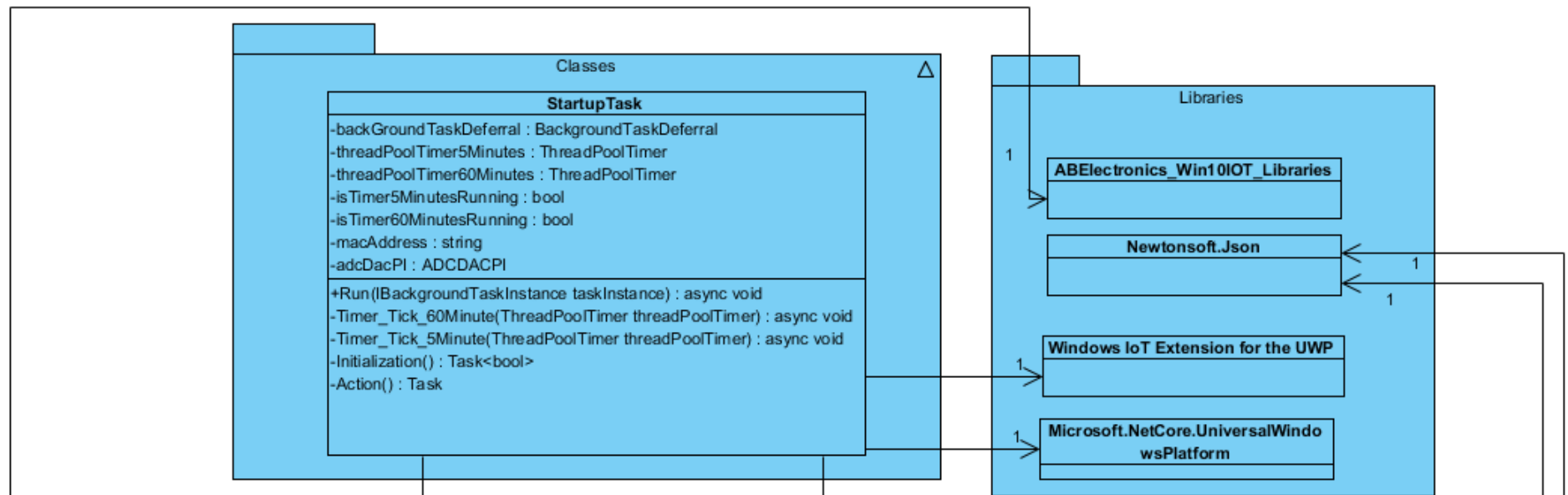
B Klassendiagram Cliënt applicatie deel 2



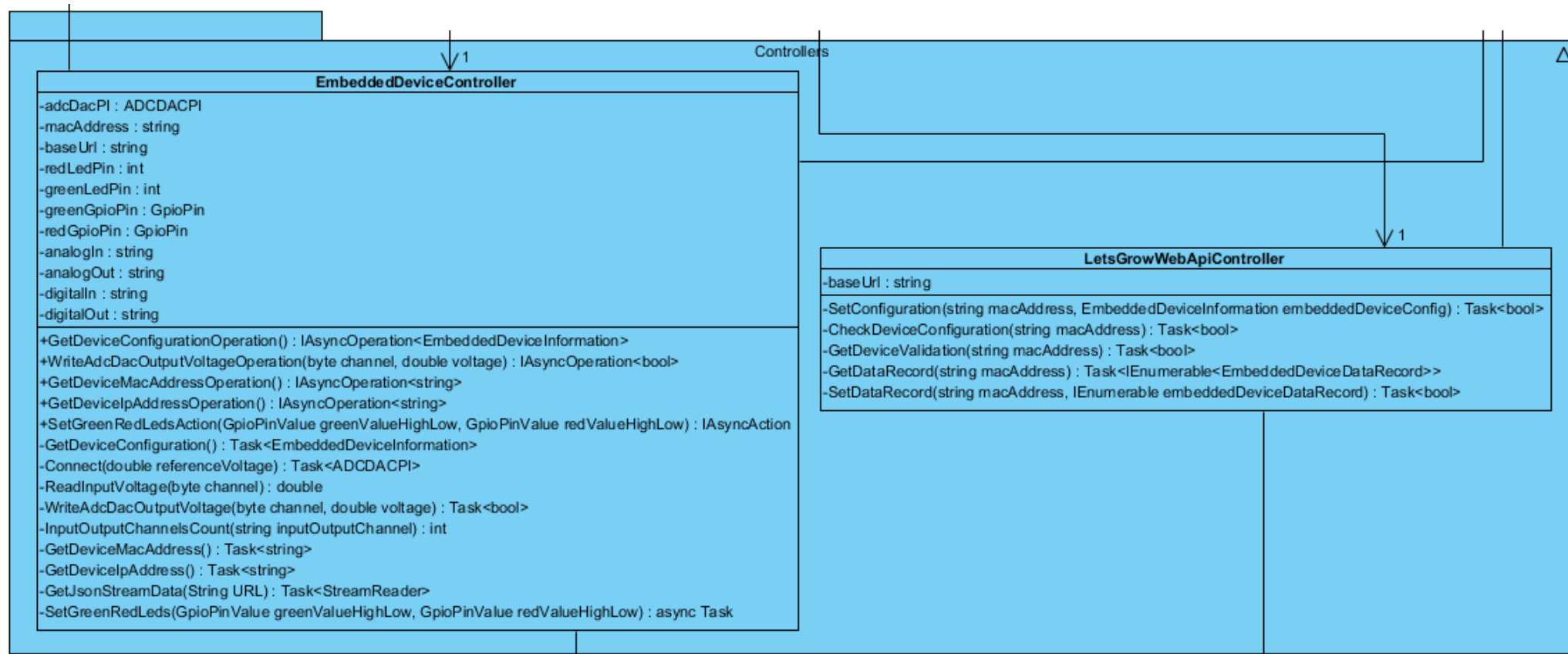
C Klassendiagram Cliënt applicatie deel 3



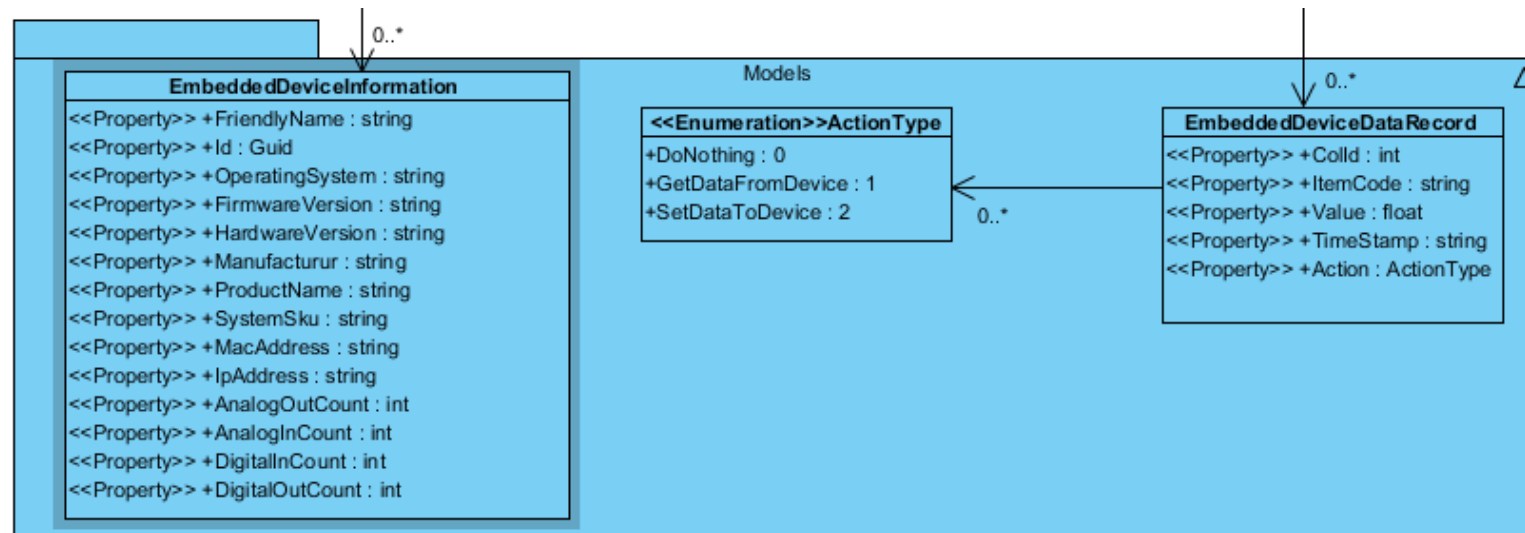
D Ontwerp Klassendiagram Cliënt applicatie deel 1



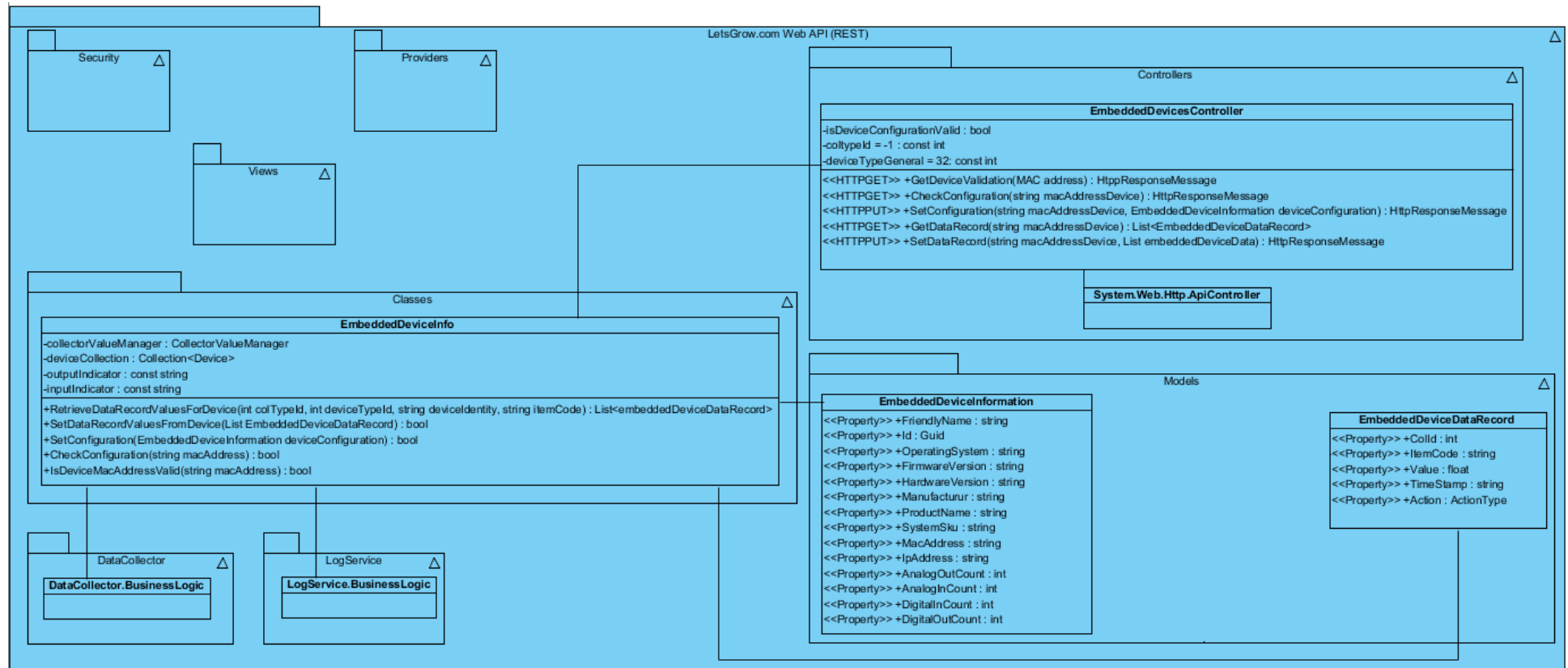
E Ontwerp Klassendiagram Cliënt applicatie deel 2



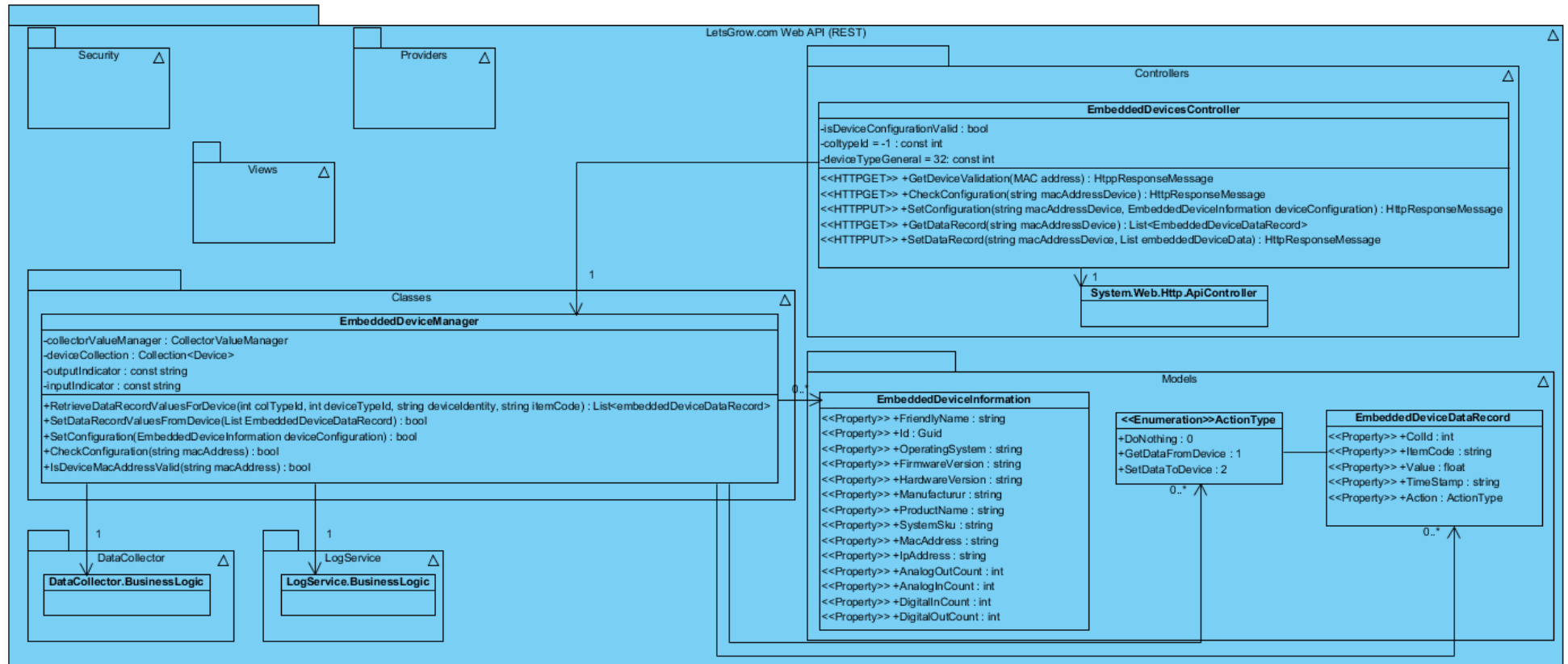
F Ontwerp Klassendiagram Cliënt applicatie deel 3



G Klassendiagram LetsGrow.com Web API server applicatie



H Ontwerp Klassendiagram LetsGrow.com Web API server applicatie



I Plan Van Aanpak

Plan van aanpak

Data wegschrijven naar klimaatcomputers

Bedrijf: LetsGrow.com

Versie: 1.0

Datum: 3-03-2015

Plaats: Vlaardingen

Opdrachtgever:

Peter Hendriks

Bedrijfsmentoren:

Leon Batta/Edwin Vis

Documenthistorie:

Datum	Versie	Beschrijving
24-2-2016	0.5	Concept
3-3-2016	1.0	Definitief

Distributie:

Naam	Versie Concept	Versie Definitief
Peter Hendriks		X
Leon Batta		
Edwin Vis	X	X

Akkoord opdrachtgever en projectleider.

Peter Hendriks	Eray Soy
Handtekening:	Handtekening:

Inhoudsopgave

1.	Inleiding	1
2.	Opdrachtschrijving	2
2.1	Aanleiding	2
2.2	Probleemstelling	2
2.3	Doelstelling van de afstudeeropdracht	2
2.4	Resultaat	2
2.5	Afbakening	2
3.	Projectorganisatie	3
3.1	Methodiek	3
3.2	Tussenresultaten	4
3.3	Eindresultaten	4
3.4	Rolverdeling	5
3.5	Afspraken opdrachtgever, bedrijfsmentoren/projectmanagers en de afstudeerder	5
4	Hardware en Software	6
4.1	Hardware	6
4.2	Software	6
5.	Kwaliteit	7
6.	Planning	8
7.	Kosten en Baten	9
7.1	Kosten	9
7.2	Baten	9
8.	Risicoanalyse	10

1. Inleiding

LetsGrow.com is een bedrijf dat zijn klanten de mogelijkheid biedt om hun teelt gerelateerde gegevens online te registeren en te analyseren. Via het online platform zijn de gegevens van onder andere het kasklimaat, het gewas, arbeid en het energieverbruik in te zien. Hier kunnen de klanten gemakkelijk grafieken en rapporten creëren om de gegevens van hun teelt inzichtelijk te krijgen. LetsGrow.com streeft ernaar om wereldwijd de online service te zijn voor tuinbouwondernemers, adviseurs, onderzoekers en andere belanghebbenden in de tuinbouwbranche.

In de toekomst wil LetsGrow.com data door middel van een minicomputer “direct” door kunnen sturen naar de klimaatcomputers van klanten onafhankelijk van het type of merk. Hiermee willen ze de regelingen in de klimaatcomputers beïnvloeden.

2. Opdrachtomschrijving

2.1 Aanleiding

In de toekomst wil LetsGrow.com data door middel van een minicomputer “direct” door kunnen sturen naar de klimaatcomputers van klanten onafhankelijk van het type of merk. Hiermee willen ze zelf de regelingen in de klimaatcomputers beïnvloeden.

2.2 Probleemstelling

LetsGrow heeft momenteel niet de mogelijkheid om de regelingen in de klimaatcomputers van derden (Hortimax, SerCom en Priva) te kunnen beïnvloeden zonder de tussenkomst van de klimaatcomputer leveranciers.

2.3 Doelstelling van de afstudeeropdracht

Een oplossing(prototype) met bijbehoren applicatie bouwen voor LetsGrow.com, zodat met behulp van dit prototype regelingen in de klimaatcomputers kunnen worden beïnvloed, zonder de tussenkomst van de klimaatcomputer leveranciers.

2.4 Resultaat

Een prototype waarmee door LetsGrow.com op afstand regelingen kunnen worden beïnvloed zonder de tussenkomst van de klimaatcomputer leveranciers.

2.5 Afbakening

- Dit project wordt uitgevoerd volgens de waterval methode (fasering).
- De documentatie en de gemaakte software worden geschreven voor het samengesteld prototype(Raspberry PI 2 + Shield)
- Het prototype wordt getest op een Hoogendoorn Klimaatcomputer.
- Het prototype wordt door een deskundige (service monteur) aangesloten aan een klimaatcomputer.
- Voor het prototype ‘minicomputer’ gaat de Raspberry PI 2 gebruikt worden.
- Behuizing voor het prototype wordt gerealiseerd indien er tijd over is.
- De besturingssystemen en ontwikkeltools die geanalyseerd gaan worden, zijn de Windows IoT met Visual Studio en de Raspbian met MonoDevelop.

3. Projectorganisatie

3.1 Methodiek

Dit project zal uitgevoerd worden volgens de waterval methodiek waarbij er gebruik gemaakt wordt van 3 fases: de oriëntatiefase, definitiefase en de ontwikkelfase. Er is voor deze methode gekozen, omdat het een duidelijk methode is met een duidelijke onderscheiding van fases. Hierdoor is alles van te voren overzichtelijk en goed ingepland. Daarnaast moet er ook gedocumenteerd worden.

Oriëntatiefase:

In de oriëntatiefase gaat de afstudeeropdracht verhelderd worden. Daarnaast gaan de risico's geanalyseerd worden en er wordt een planning gemaakt voor het gehele project. Deze zaken zullen in dit Plan van aanpak vinden zijn.

Definitiefase:

In de definitiefase zal er eerst de benodigde literatuur ingelezen moeten worden. Aan de hand van een onderzoeksvraag en deelvragen zullen de besturingssystemen en ontwikkeltools geanalyseerd worden. Nadat deze geanalyseerd zijn, dan worden de hardware componenten geanalyseerd. Nadat deze allemaal geanalyseerd zijn, worden de eisen achterhaald. Aan de hand van de analyse en eisen zal er selectie op de hardware en software gebied plaatsvinden. Uiteindelijk wordt in deze fase de systemarchitectuur en het definitierapport opgesteld.

Ontwikkelfase:

In de ontwikkelfase zal eerst de architectuur ontwerp gerealiseerd worden. Daarnaast wordt de software code geschreven en een testplan opgesteld. Aan de hand van dit testplan zal het gehele prototype op functionaliteiten getest worden. Daarnaast wordt er in deze fase ook allerlei documenten opgeleverd.

3.2 Tussenresultaten

Definitierapport:

In dit rapport zal de analyse gedeeltes komen van de hardware en de software. Daarnaast wordt er een analyse uitgevoerd om tot aan systeemeisen te kunnen komen. De hardware en software selectie zal in dit rapport ook beschreven worden en uiteindelijk zal er een globaal ontwerp (klassendiagrammen) opgesteld worden.

Ontwerprapport:

In dit rapport zal het ontwerp van het prototype uitgewerkt en beschreven worden.

Testplan:

Het testplan beschrijft hoe het prototype getest dient te worden en wat de resultaten moeten zijn. Dit zal gebeuren aan de hand van een stappenplan, waarbij de tester alle stappen moet doorlopen om het gehele prototype te kunnen testen. Alle eisen zullen behandeld worden in het stappenplan.

Handleiding:

Voor de servicemonteur wordt er een handleiding geschreven, zodat een monteur van Hoogendoorn het prototype snel en goed kan aansluiten op een klimaatcomputer en in bedrijf kan stellen.

3.3 Eindresultaten

Prototype:

Uiteindelijk wordt het prototype met de gekozen shield en ontworpen applicatie opgeleverd.

3.4 Rolverdeling

Hieronder wordt de rolverdeling van alle betrokken beschreven.

Contactgegevens opdrachtgever, bedrijfsmentoren/projectmanagers en projectleider:

Naam:	Functie:	Email:
Dhr. Peter Hendriks	Opdrachtgever	phe@letsgrow.com
Dhr. Leon Batta	Bedrijfsmentor/Projectmanager	lba@letsgrow.com
Dhr. Edwin Vis	Bedrijfsmentor/Projectmanager	ev@letsgrow.com
Eray Soy	Afstudeerder/Projectleider	eray_075@msn.com

3.5 Afspraken opdrachtgever, bedrijfsmentoren/projectmanagers en de afstudeerder.

Afspraken met de opdrachtgever:

De communicatie met de opdrachtgever Dhr. Peter Hendriks zal zowel via de mail als mondeling plaatsvinden. De opdrachtgever bevindt zich namelijk een kamer naast de afstudeerder. Indien nodig kan er op elk gewenst moment overleg plaatsvinden.

Afspraken met de bedrijfsmentoren /projectmanagers:

De communicatie met de bedrijfsmentoren Dhr. Leon Batta en Dhr. Edwin Vis zullen zowel via de mail als mondeling plaatsvinden. Dhr. Leon Batta is op de dinsdagen en woensdagen aanwezig. Dhr. Edwin Vis is van maandag tot donderdag aanwezig. Op overige werkdagen zal er elektronisch contact opgenomen kunnen worden met de bedrijfsmentoren.

Afspraken met de afstudeerder/projectleider:

De afspraken die met de afstudeerder gemaakt zijn:

- De afstudeerder moet voor 9:30 's ochtend in het gebouw van Hoogendoorn in Vlaardingen aanwezig zijn.
- De afstudeerder moet 5 dagen per week, 8 uur per dag aan zijn afstudeeropdracht werken.
- Afwezigheid wegens vooraf ingeplande zaken moeten altijd van te voren bij zowel de opdrachtgever als bedrijfsmentor door middel van een e-mail gemeld worden.
- De afstudeerder moet elke week zijn uren bijhouden. Deze urenregistratie moet de week daarop bij de receptie ingeleverd worden.
- Een onvoorziene afwezigheid moet de afstudeerder voor 9:00 telefonisch of via mail bij de opdrachtgever en bedrijfsmentoren melden.

4 Hardware en Software

Dit hoofdstuk zal de hardware en de software behandelen die voor het prototype benodigd zijn.

4.1 Hardware

De hardware voor het prototype bestaat uit 2 delen. Het eerste gedeelte (de minicomputer) was al van te voren vastgesteld met de opdrachtgever en de bedrijfsmentoren. Voor dit project wordt de Raspberry PI 2 gebruikt. De Raspberry PI 2 is al voor het afstuderen geïnterpreteerd, maar er gaat wel een uitgebreide analyse plaatsvinden.

Om de software van het prototype te kunnen ontwikkelen, zijn de volgende hardware componenten benodigd voor de PI 2:

- Een voeding van 5 volt (2 ampère) voor de PI.
- HDMI kabel voor het kunnen instellen van het besturingssysteem.
- Een monitor om aan de Raspberry PI te kunnen aansluiten voor het initieel configureren van de Raspberry PI.
- Micro SD kaart van 16 GB voor het besturingssysteem en data opslag.
- Netwerkkabel voor het op afstand te kunnen verbinden met de PI.
- Actieve netwerkverbinding met het internet.
- Toetsenbord en een muis voor de input.
- Een USB HUB voor extra spanning, indien het shields dat nodig zullen hebben.

Het tweede gedeelte (de minicomputer) gaat uit een shield bestaan die gekozen moet worden. Om dit goed te kunnen kiezen gaat er eerst geanalyseerd worden welke shields beschikbaar zijn. Al deze zaken zullen in de volgende fase geanalyseerd worden.

4.2 Software

De software gedeelte van het prototype bestaat uit een besturingssysteem en de te ontwikkelen applicatie. Van te voren is het de voorkeur voor het besturingssysteem en ontwikkeltool/programmeertaal uitgesproken. LetsGrow.com wil gebruik maken van de Windows IoT besturingssysteem met de ontwikkeltool Visual Studio en C# als programmeertaal. Eerst worden deze geanalyseerd of ze haalbaar zijn voor dit project en aan de hand van de analyse zal er conclusie getrokken worden of het ook daadwerkelijk toegepast gaan worden. Daarnaast wordt als alternatief Raspbian met MonoDevelop geanalyseerd of het mogelijk is om C# applicaties te kunnen bouwen en te kunnen draaien.

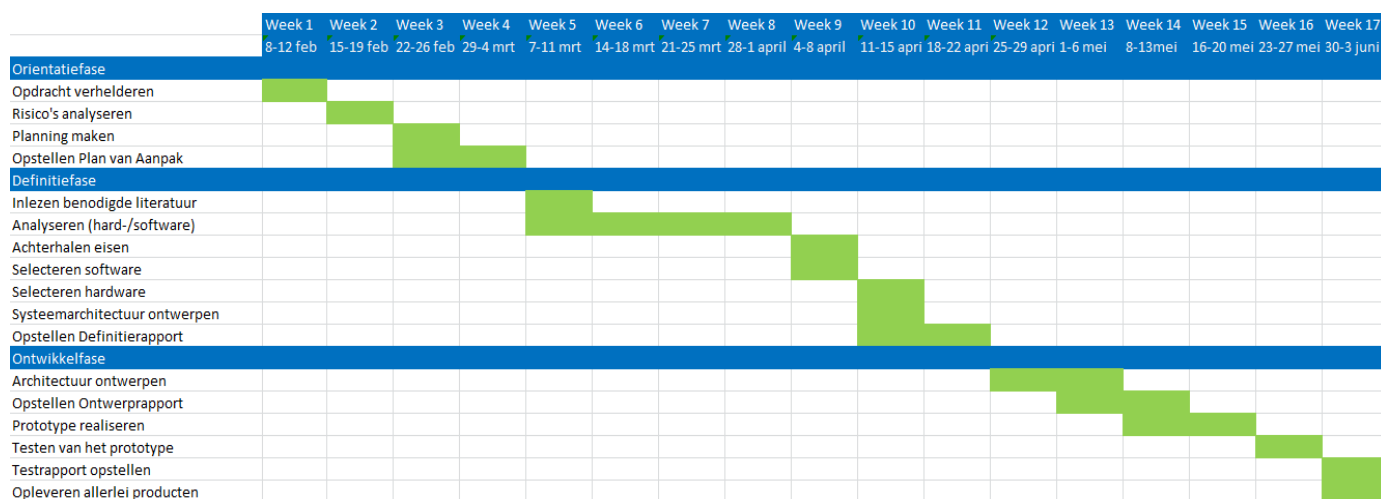
5. Kwaliteit

De kwaliteit van het project wordt gerealiseerd aan de hand van het testrapport die in de laatste fase gaat plaatsvinden. Het prototype gaat getest worden op functionaliteiten door Dhr. Joost Simons (de tester van LetsGrow.com).

De ontwikkelde software code wordt gecontroleerd door de bedrijfsmentor. Dhr. Edwin Vis. Om deze code te reviewen houdt hij vast aan de richtlijnen van de ifSQ. De ontwikkelde software code moet minimaal aan de richtlijnen van de ifSQ Level-1 voldoen. Daarnaast hanteert hij de “naming-guidelines” van Microsoft. Tevens wordt er naar de “code style” en mogelijke applicatie fouten gekeken. Deze zaken worden uiteindelijk gecontroleerd om te kijken of de ontworpen applicatie aan de kwaliteitsnormen van LetsGrow.com voldoet.

6. Planning

De planning voor het gehele afstudeerproject zal in totaal 17 weken in beslag nemen. De planning ziet er als volgt uit.



Figuur 1: Fase Planning

7. Kosten en Baten

Dit hoofdstuk zal de kosten en de baten behandelen.

7.1 Kosten

De maximale kosten mogen voor het onderzoek aan het prototype niet meer dan 300 euro zijn. Dit is inclusief alle benodigde extra hardware voor het onderzoek.

De kosten voor de software mogen voor het prototype geen “extra” licentiekosten hebben. Alles moet zoveel mogelijk in open-source of demo/gratis software gerealiseerd worden.

De besturingssystemen die voor dit project onderzocht en gebruikt gaan worden zijn voornamelijk open-source en closed-source. Dit houdt in dat er geen licentiekosten aan verbonden zijn. Windows IOT is een closed-source besturingssysteem. Ondanks dit licentie model is ook dit besturingssysteem gratis te gebruiken.

Voor het programmeerwerk wordt er gebruik gemaakt van de gratis editie van Visual Studio en/of van MonoDevelop op de Raspberry PI. Deze zijn gratis te gebruiken. Voor in productie(klant versie) moet er wel een licentie aangeschaft worden voor Visual Studio. LetsGrow heeft al een licentiepack genaamd Action Pack van Microsoft, waarin onder andere licenties voor Visual Studio's zitten. Voor de klant versie hoeven er dus geen “extra” kosten gemaakt te worden op het gebied van software.

7.2 Baten

De baten voor LetsGrow.com zijn dat dit prototype een aanvullende dienst kan gaan aanbieden naast de mogelijkheden die LetsGrow al heeft. Klanten krijgen de mogelijkheid om via het prototype te kunnen regelen op extra parameters die extern kunnen worden berekend. Hierdoor zullen de klanten het LetsGrow.com platform intensiever gebruiken en zullen klanten hun abonnement eerder verlengen/longer aanhouden. Nieuwe mogelijkheden en tevreden klanten zorgen tevens voor mond-op-mond reclame. Het prototype draagt daarbij aan de innovatie uitstraling van LetsGrow en kan worden gebruikt voor de persberichten.

8. Risicoanalyse

Dit hoofdstuk zal de risico's beschrijven die tijdens dit project voor kunnen komen.

Omschrijving	De afstudeerder is tijdens de gehele periode meer dan 5 werkdagen afwezig wegens ziekte/ongeval.
Risico	De afstudeerder is tijdens de gehele periode meer dan 5 werkdagen afwezig wegens ziekte/ongeval.
Kans	Klein
Kansverlagende maatregel	Geen
Impact	Groot
Impactverlagende maatregel	Spoedig mogelijk afspraak inplannen met de opdrachtgever en bedrijfsmentoren om verder te bespreken.
Plan B	Geen.

Omschrijving	De opdrachtgever is tijdens de gehele periode meer dan 14 werkdagen afwezig wegens ziekte/ongeval.
Risico	De opdrachtgever is tijdens de gehele periode meer dan 5 werkdagen afwezig wegens ziekte/ongeval.
Kans	Klein
Kansverlagende maatregel	Geen
Impact	Middel
Impactverlagende maatregel	Met de opdrachtgever afspreken wie als alternatieve opdrachtgever beschikbaar kan zijn, indien de opdrachtgever meer dan 5 werkdagen afwezig is.
Plan B	Met de alternatieve opdrachtgever contact opnemen.

Omschrijving	De bedrijfsmentor(en) is/zijn tijdens de gehele periode meer dan 14 werkdagen afwezig wegens ziekte/ongeval.
Risico	De bedrijfsmentor(en) is/zijn tijdens de gehele periode meer dan 14 werkdagen afwezig wegens ziekte/ongeval.
Kans	Klein
Kansverlagende maatregel	Geen
Impact	Middel
Impactverlagende maatregel	Voor dit project zijn er 2 bedrijfsmentoren beschikbaar. Als de een uitvalt, dan kan er met de andere contact opgenomen worden. Indien ze allebei uitvallen, dan moet er met de opdrachtgever contact opgenomen worden.
Plan B	Met de 2 ^e bedrijfsmentor contact opnemen en/of met de opdrachtgever.

Omschrijving	De bestelde hardware komt niet op tijd aan/duurt meer dan 5 werkdagen.
Risico	De bestelde hardware komt niet op tijd aan en/of het duurt meer dan 5 werkdagen om geleverd te worden.
Kans	Middel
Kansverlagende maatregel	De hardware bestellen bij een leverancier die het snelst kan bezorgen. Indien het kan, ophalen bij een fysieke winkel.
Impact	Groot
Impactverlagende maatregel	Kijken wanneer de bestelling ontvangen zal worden. Als het langer duurt dan normaal, gelijk een alternatieve winkel zoeken waar dezelfde hardware besteld kan worden en de lopende bestelling/pakket annuleren.
Plan B	Met de bedrijfsmentoren en opdrachtgever bespreken en zo spoedig mogelijk een alternatieve winkel benaderen voor de nieuwe bestelling.

Omschrijving	De bestelde hardware is defect/beschadigd bij aankomst
Risico	De bestelde hardware is defect/ beschadigd bij aankomst
Kans	Middel
Kansverlagende maatregel	Geen
Impact	Groot
Impactverlagende maatregel	De hardware die ontvangen wordt, zo nauwkeurig mogelijk uitpakken en voorzichtig mee zijn.
Plan B	De hardware zo snel mogelijk retour sturen/langs brengen naar de winkel en zo snel mogelijk nieuwe bestellen. De opdrachtgever en de bedrijfsmentoren hierover informeren.

Omschrijving	De hardware loopt schade op.
Risico	De hardware loopt schade op.
Kans	Middel
Kansverlagende maatregel	De handleiding van de hardware goed lezen voor bijvoorbeeld hoeveel voltage hij aankan. Tevens na gebruik de hardware goed opbergen in een schermhoesje en in de kast leggen, zodat anderen er niet eraan kunnen zitten. Ook is het aangeraden om niet te eten of te drinken in de buurt van de hardware.
Impact	Groot
Impactverlagende maatregel	Vervangende hardware bestellen bij de leverancier.
Plan B	De hardware zo snel mogelijk retour sturen/langs brengen naar de leverancier en zo snel mogelijk een nieuwe bestellen. De opdrachtgever en de bedrijfsmentoren hierover informeren.

J Definitierapport

Definitierapport

Data wegschrijven naar klimaatcomputers

Bedrijf: LetsGrow.com

Versie: 1.0

Datum: 22-04-2015

Plaats: Vlaardingen

Opdrachtgever:

Peter Hendriks

Bedrijfsmentoren:

Leon Batta/Edwin Vis

Documenthistorie:

Datum	Versie	Beschrijving
14-04-2016	0.3	Concept
22-04-2016	1.0	Definitief

Distributie:

Naam	Versie Concept	Versie Definitief
Peter Hendriks		X
Leon Batta		
Edwin Vis	X	X

Akkoord opdrachtgever en projectleider.

Peter Hendriks	Eray Soy
Handtekening:	Handtekening:

Inhoudsopgave

1. Inleiding	1
2. Analyse huidige opstelling.....	2
3. Analyse hardware, besturingssystemen en ontwikkeltools	4
3.1 Wat is een klimaatcomputer?	6
3.1.1 Wat is een regeling?	7
3.1.2 Hoe kan een regeling worden beïnvloed?.....	8
3.2 Wat is een Raspberry Pi 2?	9
3.2.1 Welke communicatie interfaces zijn er mogelijk met de Raspberry Pi 2?	10
3.3 Wat is een shield?	12
3.3.1 Hoe wordt de communicatie opgebouwd tussen de shields en de Pi 2?	12
3.3.2 Welke shields zijn beschikbaar voor de Pi 2 om de regelingen te kunnen beïnvloeden in klimaatcomputers?	13
3.4 Hoe kan er communicatie opgebouwd worden tussen het prototype en LetsGrow.com?..	13
3.5 Hoe kan er communicatie opgebouwd worden tussen het prototype en de klimaatcomputer?	14
3.6 Hoe kan er middels de ontwikkelomgeving Visual Studio en de programmeertaal C# een applicatie gebouwd worden voor de Raspberry Pi 2 draaiend onder Windows IoT die een aangesloten shield aanstuurt?	14
3.6.1 Wat is Windows IoT	14
3.6.2 Wat is Visual Studio?	16
3.6.3 Welke typen applicaties zijn er mogelijk op Windows IoT?	17
3.6.4 Welke communicatie interfaces ondersteunt de Raspberry Pi 2 met Windows IoT?...	18
3.6.5 Hoe kunnen er door middel van Visual Studio applicaties gebouwd en gedebugd worden op het Windows IoT besturingssysteem?.....	18
3.6.6 Hoe kan er een applicatie gebouwd worden om de Input/Output van de Raspberry Pi 2 aan te kunnen sturen op het Windows IoT besturingssysteem?	20
3.7 Hoe kan er middels de ontwikkelomgeving MonoDevelop en de programmeertaal C# een applicatie gebouwd worden voor de Raspberry Pi 2 draaiend onder Raspbian die een aangesloten shield aanstuurt?	24
3.7.1 Wat is Raspbian	24
3.7.2 Wat is MonoDevelop?	26
3.7.3 Welke typen applicaties zijn er mogelijk op Raspbian?	26
3.7.4 Welke communicatie interfaces ondersteunt de Raspberry Pi 2 met Raspbian?	26
3.7.5 Hoe kan er door middel van MonoDevelop applicaties gebouwd en gedebugged worden op de Raspbian?	27

3.7.6	Hoe kan er een applicatie gebouwd worden om de Input/Output van de Raspberry Pi 2 aan te kunnen sturen op het Raspbian besturingssysteem?	27
4.	Conclusie Analyse hardware, besturingssystemen en ontwikkeltools.	28
5.	De wensen van de opdrachtgever:.....	29
6.	De Requirements.....	30
6.1	Functionele eisen:	30
6.2	Niet-functionele eisen:.....	30
7.	De keuze voor het besturingssysteem, ontwikkeltool en applicatie type.	31
8.	Vervolg Analyse shields	33
8.1	Verdere analyse geselecteerde shields.	35
8.1.1	4-20 mA t click shield.....	37
8.1.2	ADC-DAC Pi shield.....	40
8.1.3	Phidget Analog 4-Output.....	42
8.2	Conclusie Shields	44
9.	De keuze voor het shield.	46
10.	Architectuur nieuwe situatie	47
11.	UML Diagrammen	49
11.1	Use-case diagrammen en beschrijvingen.....	49
11.2	Use-case beschrijvingen	52
11.3	Klassendiagrammen	61
11.4	Klassen beschrijvingen.	63
11.5	Klassendiagram beschrijvingen.	64
11.6	Sequentiediagram.	67
	Bibliografie	69

1. Inleiding

In dit definitie rapport zal eerst de huidige situatie geanalyseerd worden. Daarna zal er een analyse plaatsvinden aan de hand van een hoofdvraag. Na het analyseren zullen zowel de functionele- als niet-functionele eisen opgesteld worden. Aan de hand van deze analyse en de eisen wordt er door de afstudeerder een keuze gemaakt welke componenten er gebruikt gaan worden voor het prototype. Deze gemaakte keuze zal tevens in dit rapport beschreven worden. Uiteindelijk wordt er een globaal ontwerp van de architectuur opgesteld.

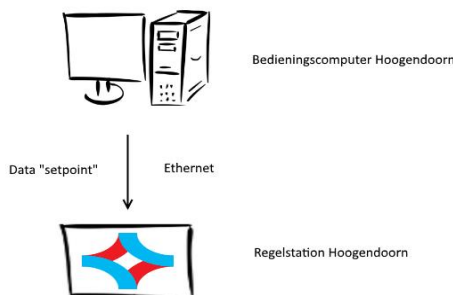
2. Analyse huidige opstelling.

In de Nederlandse kassenbouw wordt gebruik gemaakt van klimaatcomputers. Er zijn verschillende leveranciers van klimaatcomputers. Op de Nederlandse markt zijn Hoogendoorn, Priva en Hortimax de meest toonaangevende. Een Hoogendoorn klimaatcomputer bestaat uit 2 onderdelen, de bedieningscomputer en het regelstation. Deze componenten zorgen ervoor dat de klant zijn klimaat, water, energie in zijn kas(sen) kan controleren en besturen. Om al deze zaken goed te kunnen sturen en controleren, moeten er data aangeleverd worden. Hoogendoorn zorgt ervoor dat er data bij de klimaatcomputers aangeleverd wordt. Echter via de koppeling met LetsGrow.com kunnen bestaande regelingen uitgebreid worden. Data die opgehaald wordt bij LetsGrow.com, kan dan als invloed gebruikt waarop de regelingen gestuurd wordt. LetsGrow.com zorgt ervoor dat er data bij de Hoogendoorn klimaatcomputers aangeleverd kan worden (Posthumus, 2016). Een Hoogendoorn klimaatcomputer wordt eerst door middel van een software via internet gekoppeld aan LetsGrow.com. Nadat er een koppeling plaats heeft gevonden, treedt er data-uitwisseling plaats. Deze data komt via het internet als een digitale waarde de bedieningscomputer binnen.



Figuur 1: LetsGrow bedieningscomputer koppeling schema

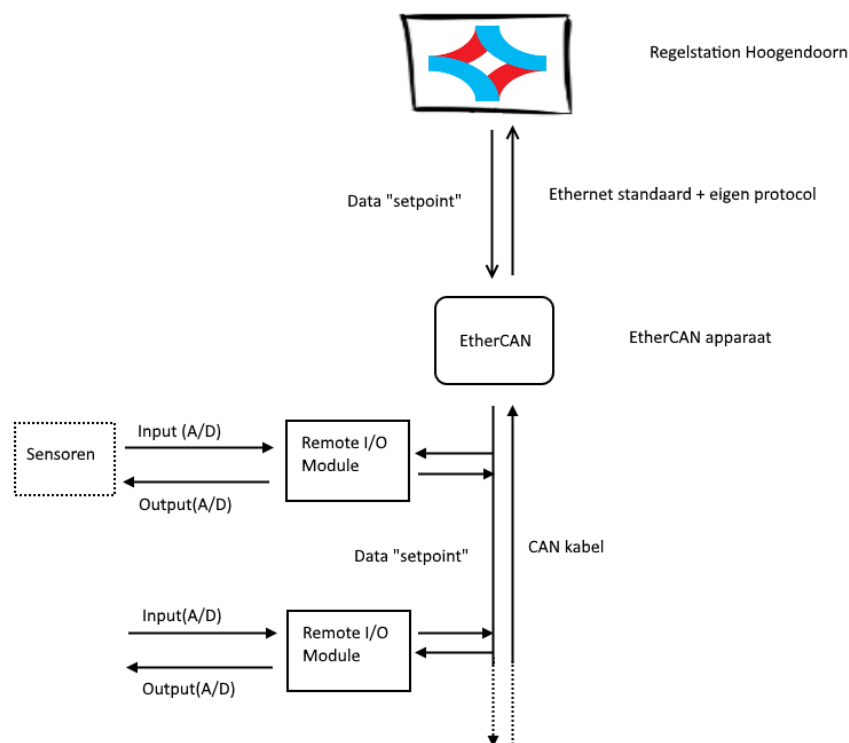
De bedieningscomputer is een PC waarop het Windows 7 besturingssysteem van Microsoft draait met de Hoogendoorn software erop. Verder heeft de PC een speciale behuizing om de temperatuur onder controle te kunnen houden. De Hoogendoorn software zorgt ervoor dat de klant zijn klimaat, watergift, warmtekrachtkoppeling(WKK's) ed. kan controleren en indien nodig kan sturen. De configuratie die de klant instelt, wordt doorgestuurd naar het regelstation. Dit gebeurt door de bedieningscomputer die alle data door middel van een ethernetverbinding doorstuurt naar het regelstation.



Figuur 2: Bedieningscomputer, regelstation koppeling schema

Het regelstation is een “Embedded mini-pc” die draait op Windows 7 Embedded besturingssysteem. De functie van het regelstation is het uitvoeren van de regelingen. Om te kunnen regelen zijn er metingen en sturingen nodig. Hiervoor zorgt het I/O-systeem. Het I/O-systeem bestaat uit een EtherCan welke de gateway is tussen ethernet en CAN-bus, vervolgens zijn er “Remote I/O modules” die decentraal opgesteld worden en aan de CAN-bus hangen (Linden, Construction EtherCan, 2014). Aan de Remote I/O modules zijn de sensoren en actuatoren verbonden.

Metingen komen via het I/O-systeem bij het regelstation binnen, het regelstation bepaalt de actie aan de hand van de meting en het setpoint. De actie wordt in een sturing omgezet die via het I/O-systeem naar het regelorgaan gaat. Bijvoorbeeld een motor om de ramen open te sturen.



Figuur 3: Regelstation, remote input/output koppeling schema

De “remote input/output” modules worden op de Hoogendoorn computer zelf geconfigureerd. Dit houdt in dat wanneer een input of een output nodig is een servicemonteur de desbetreffende module aan de CAN-kabel koppelt. Als er bijvoorbeeld een 0-10V 12-bit input nodig is, dan koppelt de servicemonteur een Beckhoff bus terminal met de desbetreffende input-voltage. Dit verschilt dus per gekoppelde sensor (Posthumus, 2016).

Omdat de specificaties van een Hoogendoorn klimaatcomputer 100% bekend zijn bij LetsGrow.com en andere klimaatcomputer leveranciers liever geen openheid van zaken geven kan LetsGrow.com alleen een “setpoint” oftewel een sturing waarde sturen naar de Hoogendoorn klimaatcomputer (Posthumus, 2016).

3. Analyse hardware, besturingssystemen en ontwikkeltools

In dit hoofdstuk worden de hardware, besturingssystemen en de ontwikkeltools geanalyseerd. Deze analyse is benodigd om eerst te kijken hoe de regelingen in klimaatcomputers beïnvloed kunnen worden, zonder de tussenkomst van de betreffende klimaatcomputer leveranciers, voordat er eisen opgesteld worden en hardware en software geselecteerd kunnen worden. De gehele analyse wordt uitgevoerd aan de hand van een bronnenonderzoek en indien nodig zullen er interviews/gesprekken gevoerd worden met deskundig personeel over het onderwerp.

Om een goede analyse te kunnen verrichten, moet er natuurlijk een hoofdvraag opgesteld worden. Deze hoofdvraag is aan de hand van de doelstelling geformuleerd.

Deze hoofdvraag luidt:

Hoe kunnen regelingen in klimaatcomputers beïnvloed worden zonder de tussenkomst van de betreffende klimaatcomputer leverancier?

Uitgangspunt voor deze hoofdvraag is dat er gebruik gemaakt wordt van een stukje hardware in de vorm van een Raspberry Pi 2 met een specifiek shield (prototype). Om deze hoofdvraag goed te kunnen beantwoorden, is deze hoofddeelvraag opgedeeld in deelvragen en zijn deze deelvragen weer opgedeeld in verschillende sub-deelvragen. Er zal door middel van een analyse antwoord gegeven worden op deze deel- en sub-deelvragen.

- Deelvraag: Wat is een klimaatcomputer?
 - Sub-deelvraag: Wat is een regeling?
 - Sub-deelvraag: Hoe kan een regeling worden beïnvloed?
- Deelvraag: Wat is een Raspberry Pi 2
 - Sub-deelvraag: Welke communicatie interfaces zijn er mogelijk met de Raspberry Pi 2?
- Deelvraag: Wat is een shield
 - Sub-deelvraag: Hoe wordt de communicatie opgebouwd tussen de shields en de Pi 2?
 - Sub-deelvraag: Welke shields zijn beschikbaar voor de Pi 2 om regelingen te kunnen beïnvloeden in klimaatcomputers?
- Deelvraag: Hoe kan er communicatie opgebouwd worden tussen het prototype en LetsGrow.com?
- Deelvraag: Hoe kan er communicatie opgebouwd worden tussen het prototype en de klimaatcomputer?

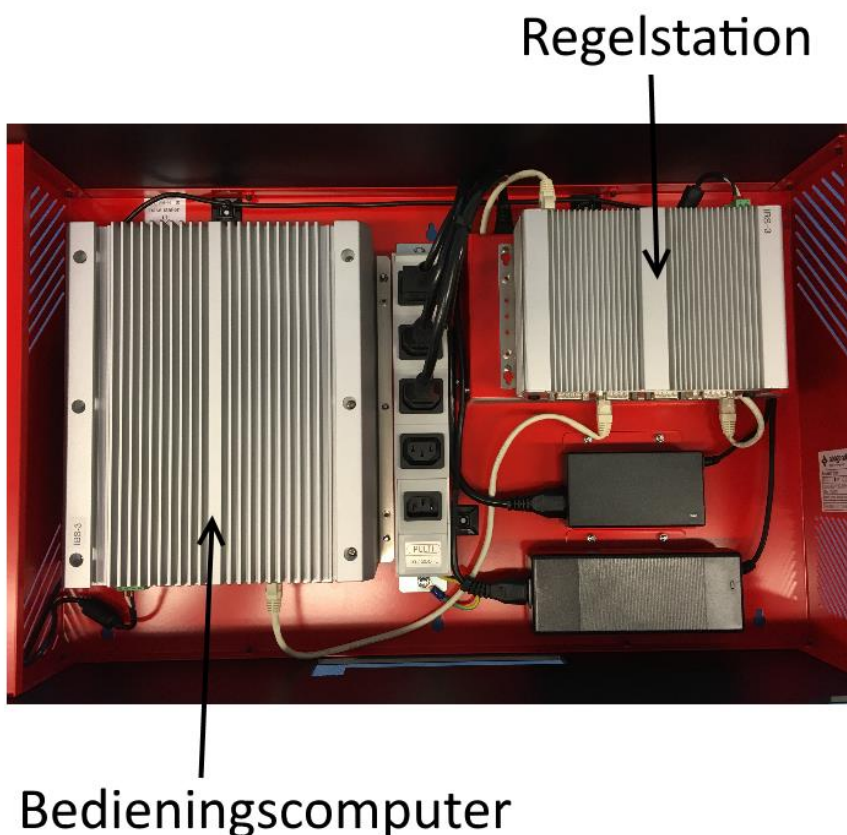
- Deelvraag: Hoe kan er middels de ontwikkelomgeving Visual Studio en de programmeertaal C# een applicatie gebouwd worden voor de Raspberry Pi 2 draaiend onder Windows IoT die een aangesloten shield aanstuurt?
 - Sub-deelvraag: Wat is Windows IoT
 - Sub-deelvraag: Wat is Visual Studio
 - Sub-deelvraag: Welke typen applicaties zijn er mogelijk op Windows IoT?
 - Sub-deelvraag: Welke communicatie interfaces ondersteunt de Raspberry Pi 2 met Windows IoT?
 - Sub-deelvraag: Hoe kunnen er door middel van Visual Studio applicaties gebouwd en gedebugd worden op het Windows IoT besturingssysteem?
 - Sub-deelvraag: Hoe kunnen er een applicatie gebouwd worden om de Input/Output van de Raspberry Pi 2 aan te kunnen sturen op het Windows IoT besturingssysteem.

- Deelvraag: Hoe kan er middels de ontwikkelomgeving MonoDevelop en de programmeertaal C# een applicatie gebouwd worden voor de Raspberry Pi 2 draaiend onder Raspbian die een aangesloten shield aanstuurt?
 - Sub-deelvraag: Wat is Raspbian
 - Sub-deelvraag: Wat is MonoDevelop
 - Sub-deelvraag: Welke typen applicaties zijn er mogelijk op Raspbian?
 - Sub-deelvraag: Welke communicatie interfaces ondersteunt de Raspberry Pi 2 met Raspbian?
 - Sub-deelvraag: Hoe kan er door middel van MonoDevelop applicaties gebouwd en gedebugd worden op de Raspbian?
 - Sub-deelvraag: Hoe kan er een applicatie gebouwd worden om de Input/Output van de Raspberry Pi 2 aan te kunnen sturen op het Raspbian besturingssysteem.

3.1 Wat is een klimaatcomputer?

Een klimaatcomputer is een computer die binnen een tuinbouwbedrijf verantwoordelijk is voor de aansturing en controleren van allerlei processen zoals het klimaat (temperatuur, RV, CO₂, verwarming, koeling), watergift maar ook warmte-kracht-koppeling (WKK's) (Posthumus, 2016).

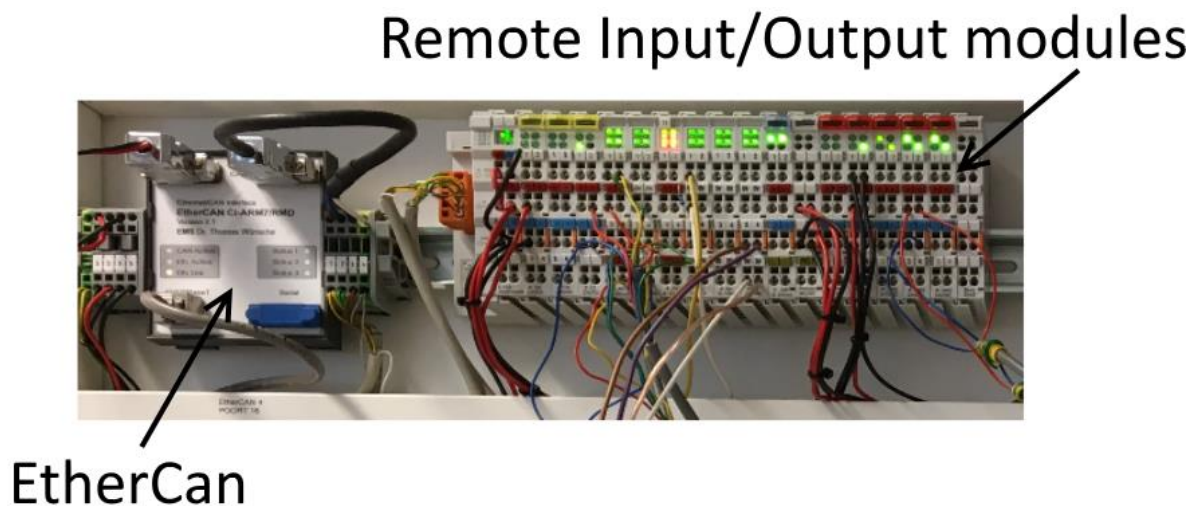
Een klimaatcomputer bestaat uit 2 onderdelen, de bedieningscomputer en het regelstation. De bedieningscomputer is een pc met een i7 processor, 8GB geheugen ram met een SSD opslag voor het besturingssysteem Microsoft Windows 7 en een normale harddisk voor dataopslag. Op deze pc draait de Hoogendoorn software (Schipper, Werking Klimaatcomputers, 2016). Verder heeft de pc een speciale behuizing om de temperatuur onder controle te kunnen houden. Het regelstation is een "Embedded mini-pc" die draait op Windows 7 Embedded besturingssysteem met een 2GB ram geheugen en een SSD opslagtype. De functie van het regelstation is het uitvoeren van de regelingen. De bedieningscomputer en het regelstation zijn met een ethernet kabel aan elkaar verbonden en wisselen voortdurend gegevens met elkaar uit. De bedieningscomputer waar de Hoogendoorn software op draait, haalt van allerlei (Hoogendoorn/LetsGrow) servers de noodzakelijke gegevens (klimaat gegevens) op en stuurt die naar het regelstation. Vanuit het regelstation worden deze gegevens weer verstuurd naar de ingangen/uitgangen modules, de "remote input/output modules" (Schipper, Werking Klimaatcomputers, 2016).



Figuur 4: Hoogendoorn Klimaatcomputer

3.1.1 Wat is een regeling?

Een regeling is een model van berekeningen waarbij uiteenlopende zaken geregeld kunnen worden. Om te kunnen regelen zijn er metingen (meting waardes) en sturingen (sturing waardes) nodig (Posthumus, 2016). Hiervoor zorgt het Input/Output-systeem (Ingang/Uitgang-systeem). Het I/O-systeem bestaat uit een EtherCan welke de gateway is tussen ethernet en CAN-bus, vervolgens zijn er “Remote Input/Output modules” die decentraal opgesteld worden en aan de CAN-bus hangen (Linden, Construction EtherCan, 2014). Aan de Remote I/O modules zijn de sensoren en actuatoren verbonden.



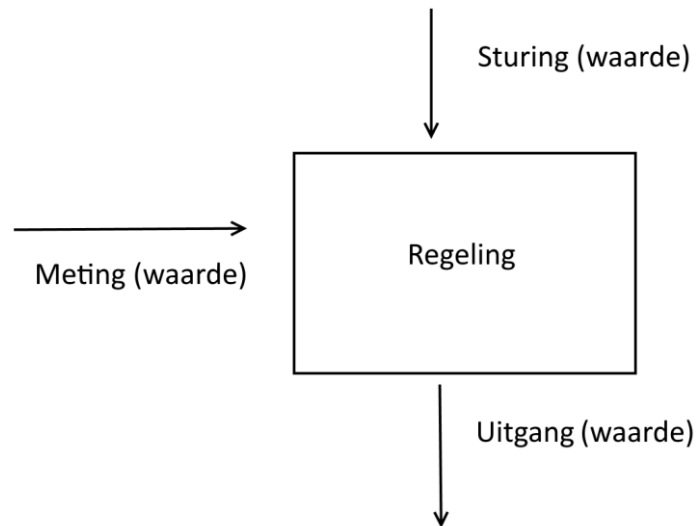
Figuur 5: EtherCan en Remote Modules

Een Remote Input/Output modules van de Hoogendoorn klimaatcomputer is een ‘bus connector’ van het merk ‘BeckHoff’ waarbij er door middel van bekabeling allerlei ingangen en uitgangen gekoppeld kunnen worden aan deze connectoren.

Metingen komen via het I/O-systeem bij het regelstation binnen, het regelstation bepaalt de actie aan de hand van de meting en de sturing (ook wel een “setpoint” genoemd) (Posthumus, 2016). De actie wordt in een uitgang sturing omgezet die via het I/O-systeem naar het regelorgaan gaat. Bijvoorbeeld een motor om de ramen open te sturen.

3.1.2 Hoe kan een regeling worden beïnvloed?

Een regeling kan aan de hand van de verschillende metingen en sturingen beïnvloed worden. Dit houdt in dat er digitale data afkomstig vanuit de klimaatcomputer en “data” afkomstig vanuit de ‘remote input/output module’ de regeling kunnen beïnvloeden (Posthumus, 2016). Door klimaatcomputer enkele digitale data te sturen(sturing) en door bijvoorbeeld een sensor aan de ingangen van de ‘remote input/output module’ te koppelen en een analoge spanning te produceren(meting), kunnen de regelingen beïnvloedt worden.

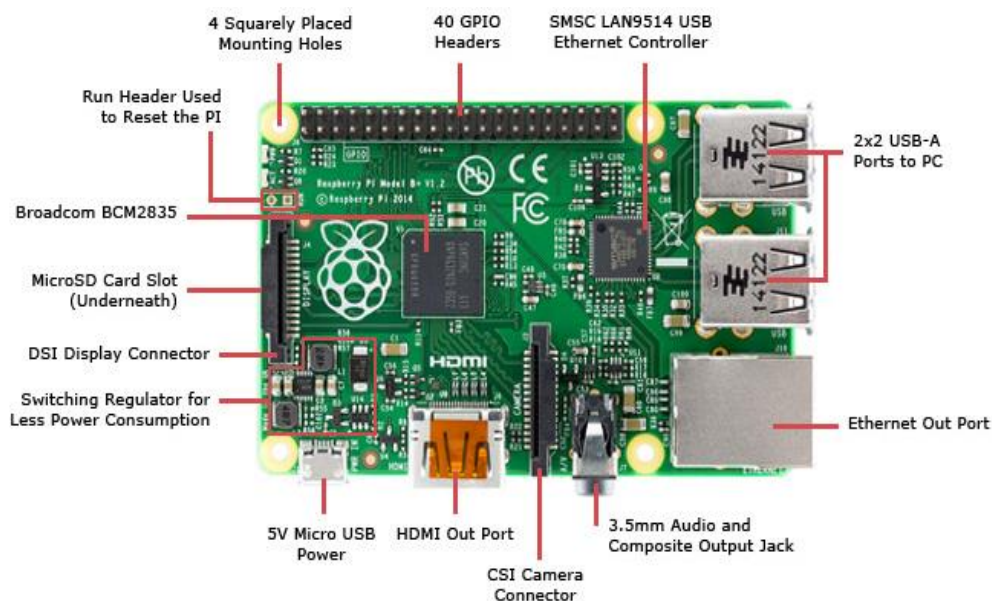


Figuur 6: Regeling model

De sensoren die aan de ‘remote input/output module’ hangen moeten een minimale spanning tussen de 0-2V of 4-20mA produceren met minimaal 8-bit resolutie om de regelingen te kunnen beïnvloeden (Schipper, Commissioning values various senso, 2015). Resolutie minder dan 8-bit beïnvloedt de regelingen niet. Resolutie heeft in deze context betrekking op de conversie van een analoge waarde naar een digitale waarde. 8-bit betekent dat de uitgang spanning in minimaal (2^8) 256 stappen te realiseren moet. Als er bijvoorbeeld een spanning tussen de 0-10V geproduceerd word in 8-bit, dan betekend dat 10V opgedeeld kan worden in 256 stukken. LetsGrow heeft wel de mogelijkheid om een “sturing (waarde)” via de bedieningscomputer naar de regeling te kunnen sturen. Alleen om de gehele regeling te kunnen beïnvloeden moet er ook een “meting (waarde)” gestuurd worden. Deze waarde kan LetsGrow op dit moment niet sturen (Hendriks, 2016).

3.2 Wat is een Raspberry Pi 2?

De Raspberry Pi 2 is een minicomputertje die ontwikkeld wordt als development platform voor scholen om studenten ermee te leren programmeren. Verder wordt het ook veel gebruikt door hobbyisten. Maar het kan ook heel goed gebruikt worden voor het besturen van allerlei processen uiteenlopend van heel simpel tot heel complex (Richard, 2014). Op de Pi 2 kan er een besturingssysteem gedraaid worden in tegenstelling tot een Arduino. Er zijn verschillende besturingssystemen beschikbaar variërend van Windows IoT (Raspberry Pi Foundation, The MagPi Magazine Issue 34, 2015) tot allerlei Linux varianten. Het officieel door de Raspberry Pi Foundation ondersteunde besturingssysteem Pi is Raspbian (Upton, What operating system (os) does it use?, 2016). De processor bestaat uit een Broadcom BCM2835 ARM-microprocessor die op 800 mHz wordt aangedreven. Daarnaast heeft hij 1 GB ram aan boord. Verder zijn er 4 usb-controllers en een 100 MBit netwerkaansluiting aanwezig. Voor opslag maakt hij gebruik van een Micro-SD geheugenkaart en als voeding kan elke 5 Volt-adapter met een Micro-USB aan de Pi aangesloten worden (Raspberry Pi Foundation, The MagPi Magazine Issue 31, 2015). Er kan ook een monitor gekoppeld worden aan de Pi door middel van de HDMI-aansluiting of door Display Serial Interface (DSI) Display Connector. Camera modules kunnen aan de hand van Camera Serial interface (CSI) connector gekoppeld worden. Ook zijn er 2 'leds' aanwezig die de status van het systeem kunnen laten zien. Deze kunnen handig zijn voor het prototype om door middel van de 'leds' een status te laten zien van de geïnstalleerde software.



Figuur 7: Raspberry Pi 2 (Robillard, 2015)

Om op de Raspberry Pi 2 minimaal te laten werken, zijn de volgende zaken benodigd: 4GB micro SD-kaart met een besturingssysteem, voeding van minimaal 5V micro USB (Raspberry Pi Forums, 2016).

Bron:<https://www.raspberrypi.org/documentation/installation/sd-cards.md>

Indien gewenst kan er door middel van HDMI een monitor gekoppeld worden en via een van de USB-poorten een muis en/of toetsenbord. Voorts beschikt de Pi over een Ethernet port zodat er een internet connectie opgezet kan worden.

Bron:<https://www.raspberrypi.org/help/quick-start-guide/>

3.2.1 Welke communicatie interfaces zijn er mogelijk met de Raspberry Pi 2?

Op de Raspberry Pi 2 zijn 40 "General Purpose Input Output" (GPIO) pinnen aanwezig (Raspberry Pi Foundation, Usage GPIO pins, 2016). Hiermee kan er hardwarematig een extensie bordje (shield) aan de Raspberry Pi gekoppeld worden. Op deze manier kan de Raspberry Pi 2 communiceren met allerlei extra toepassingen, zoals sensoren, schermen enz. Voor het prototype zijn deze pinnen van groot belang, omdat de shields door middel van deze pinnen aan de Raspberry Pi aangesloten kunnen worden en data kunnen uitwisselen door middel van standaarden zoals Serial Peripheral Interface (SPI), Inter-integrated Circuit (I²C), Universal Asynchronous Receiver Transmitter (UART) (Upton, What hardware interfaces does it have?, 2016) en Pulse Width Modulation (PWM). Tevens is er de mogelijkheid om door middel van de USB-poorten data uit te kunnen uitwisselen.

Serial Peripheral Interface (SPI) is een interface bus (synchrone seriële data bus) dat meestal gebruikt wordt voor het sturen van gegevens tussen microcontrollers en kleine randapparatuur zoals sensoren. Het maakt gebruik van aparte klok- en datalijnen, samen met een select lijn naar het apparaat waarmee communicatie opgebouwd wordt.

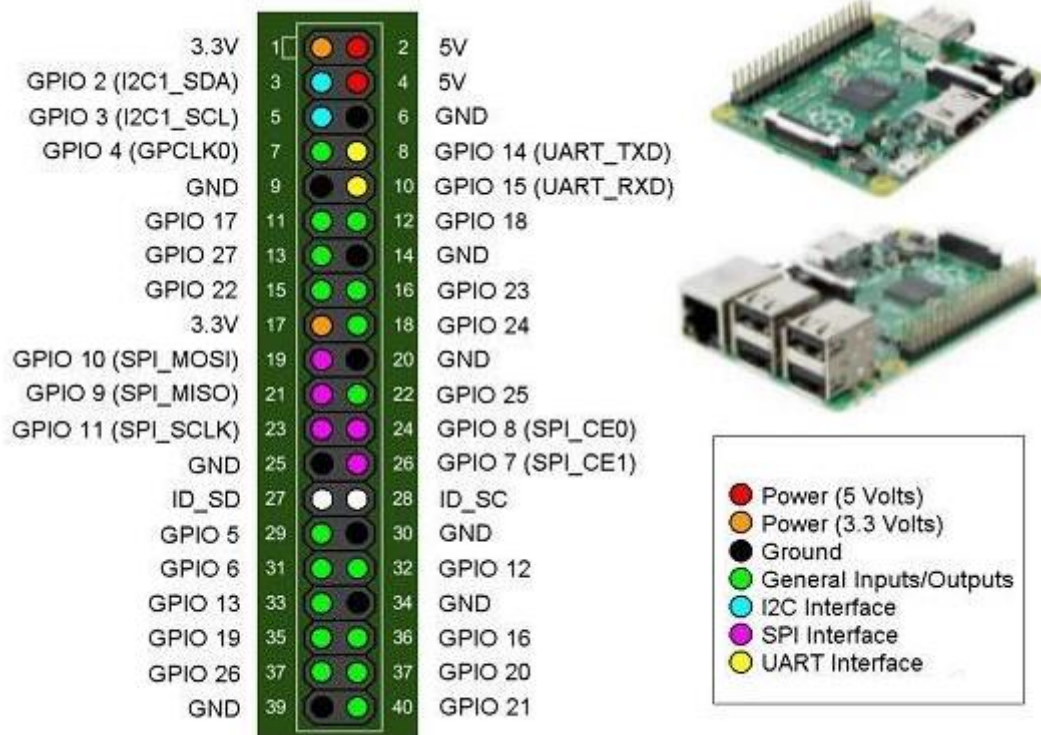
Het Inter-Integrated Circuit (I²C) protocol is een protocol dat bedoeld is om meerdere "Slaves" digitale geïntegreerde schakelingen ("chips"), met één of meer "master" chips te laten communiceren. Net als de Serial Peripheral Interface (SPI) is het alleen bedoeld voor korte afstand communicatie binnen een apparaat. I²C bestaat uit een serieel clock (SCL) signaal en een serieel data-lijn (SDA) signaal. I²C werkt ook volgens het Master/Slave principe (Sparkfun Tutorials I2C, 2016).

SPI (data-overdracht: 1Mbits/sec) is vergeleken met een I²C in normaal modus (data-overdracht: "normal-mode" 100 Kbits/sec) sneller maar of dit voor het prototype uitmaakt, zal na het opstellen van de eisen en de wensen duidelijk worden (Hanabusa, 2007).

Bron: <http://codeandlife.com/2012/07/03/benchmarking-raspberry-pi-gpio-speed/>

Een UART (Universal Asynchronous Receiver / Transmitter) is de microchip die geprogrammeerd is om asynchrone communicatie te regelen in desbetreffende microcontroller of computer. Het maakt gebruik van een enkele datalijn voor het zenden en voor het ontvangen van gegevens (Sparkfun Tutorials Serial Communication, 2016).

PWM is een modulatietechniek die gebruikt wordt als vorm van elektrische voeding of als manier van digitale informatieoverdracht. Er kan een elektrisch signaal in de vorm van een blokgolf met een vaste frequentie gestuurd worden (Monk, 2015).



Figuur 8: GPIO pinnen lay-out van de Pi 2 (Herbison, 2014)

De GPIO zijn de pinnen die ervoor zorgen dat er een fysieke verbinding tussen de Pi 2 en buitenwereld gemaakt kan worden. De GPIO pinnummeringen kunnen per gebruikte shield verschillen. Dit hangt af van de applicatie van het shield zelf. Als er in de applicatie gerefereerd wordt naar de GPIO.BOARD pinnen, dan wordt er de pinnummering die voor de poorten gelden genoteerd. Indien er verwezen wordt naar de GPIO.BCM pinnummering, dan wordt de "Broadcom SOC Channel" nummering toegepast. Deze wordt in de bovenstaande figuur als GPIO X genoteerd. (Raspberry Pi Foundation, Usage GPIO pins, 2016)

3.3 Wat is een shield?

Een shield is een modulaire printplaat waarop verschillende microchips, sensoren en allerlei modules geïntegreerd zijn (Learn SparkFun Tutorials/Arduino Shields, 2016). Een shield wordt een “expansion board” oftewel een uitbreidingsbordje genoemd. Een shield wordt gebruikt om een Raspberry Pi of soortgelijke elektronische platformen te kunnen uitbreiden met extra functionaliteiten die normaliter niet aanwezig zijn op de Raspberry Pi.



Figuur 9: LoRawan Shield Voor Raspberry Pi 2 (Cooking Hacks Shields For Raspberry PI, 2016)

3.3.1 Hoe wordt de communicatie opgebouwd tussen de shields en de Pi 2?

De communicatie tussen het shield en de Pi 2 hangt af van de communicatie interface die een shield ondersteunt en of het desbetreffende shield compatible is met de Pi 2. Zoals hierboven in een deelvraag geanalyseerd te hebben, heeft de Pi 2 de mogelijkheid om door middel van de GPIO, SPI, I²C, UART, PWM en USB communicatie op te kunnen bouwen. Indien het shield een van deze communicatie interfaces ondersteunt, dan kan er door middel van deze interfaces communicatie opgebouwd worden met het desbetreffende shield. Het desbetreffende shield wordt dan op de GPIO pinnen van de Pi 2 gezet om er een verbinding mee te kunnen maken. Deze pinnen moeten natuurlijk juist geklikt worden op de desbetreffende GPIO pinnen, anders kan er geen communicatie optreden. Soms komt ook voor dat het shield niet rechtstreeks op de GPIO pinnen van de Pi 2 wordt aangesloten, door middel van een USB-kabel wordt verbonden (Phidgetsinc). Daarnaast moet er ook een stuk software gedraaid worden om het desbetreffende shield aan te kunnen spreken. Hiervoor hebben sommige shield leveranciers voorbeeld code of voorbeeld applicaties beschikbaar. Indien er besloten wordt om zelf een applicatie te ontwerpen om met het shield te communiceren, dan kunnen vaak ook ‘libraries’ gebruikt worden van de shield leveranciers/community of van leveranciers van besturingssystemen voor de Raspberry Pi 2.

3.3.2 Welke shields zijn beschikbaar voor de Pi 2 om de regelingen te kunnen beïnvloeden in klimaatcomputers?

Uit de analyse in hoofdstuk 3 blijkt dat er een minimale spanning tussen de 0-2V geproduceerd moet worden met een minimale resolutie van 8-bit om de regelingen te kunnen beïnvloeden (Schipper, Commissioning values various senso, 2015). Uitgaande van deze informatie moeten shields uitgezocht worden die een minimale spanning tussen de 0-2V kunnen genereren met minimaal 8-bit resolutie. Daarnaast moeten deze shields met de Raspberry Pi 2 kunnen communiceren. Uit een ‘snelle’ inventarisatie blijkt dat de volgende shields beschikbaar zijn om de regelingen te kunnen beïnvloeden:

- 4-20 mA T click shield van Mikroe Elektronika (MikroElektronika, 2016).
- ADC-DAC Pi van AB electronics UK (ABelectronics ADC-DAC Pi, 2016).
- Expander Pi van AB electronics UK (ABelectronics Expander Pi, 2016).
- Phidget Analog 4-Output van Phidgets.com (Phidgets.com, 2012).
- 4-20 mA sensor board voor Arduino, Raspberry Pi en Galileo van Cooking Hacks (Cooking Hacks 4-20mA Sensor Board, 2016).
- UniPi board van UniPi.Technology (UniPi.technology Products, 2016).
- Custard Pi 2 van sf-innovations.co.uk (SF innovations Custard Pi 2, 2016).
- Open Source Analog Input/Output Expansion Board van Osexpansionboard.de (Osexpansionboard.de, 2016).

3.4 Hoe kan er communicatie opgebouwd worden tussen het prototype en LetsGrow.com?

Het prototype heeft een “Ethernet-poort” aan boord en kan met behulp van een “ethernet-kabel” verbonden worden met het Internet. LetsGrow heeft een aantal methoden om met externe systemen te kunnen koppelen en om data beschikbaar te stellen. Varianten zijn SOAP, WCF en REST services. REST heeft de voorkeur. Dit gebeurt op basis van de LetsGrow.com Web API (Vis, 2016). Daarnaast hebben deze hardware componenten die aan het internet verbonden zijn, een cliënt applicatie aan boord die door LetsGrow.com zelf is ontworpen. Aan de hand van de Web API die LetsGrow.com zelf ontworpen heeft, kunnen deze cliënt systemen een verbinding maken met de LetsGrow.com servers en gegevens uitwisselen (Vis, 2016). Deze cliënt applicaties doen hun werk op normale pc’s. Om met LetsGrow communicatie op te kunnen bouwen, moet de LetsGrow.com Web API geïmplementeerd worden in de cliënt applicatie. Daarnaast is er natuurlijk een abonnement benodigd.

3.5 Hoe kan er communicatie opgebouwd worden tussen het prototype en de klimaatcomputer?

Zoals eerder geanalyseerd te hebben, bestaat de klimaatcomputer uit een bedieningscomputer en een regelstation. Het regelstation huisvest de regelingen en met behulp van de bedieningscomputer komen hier sturing waardes naartoe. Vanuit het “remote input/output module” komen er meting waardes het regelstation binnen. Het prototype moet dus een “analoge spanning” aan deze “remote input/output module” doorgeven. De communicatie tussen deze “input/output module” en de Pi 2 wordt gerealiseerd door de uitgangen van het desbetreffende shield te koppelen met een draad aan de ingangen deze “remote input/output module” (Schipper, Werking Klimaatcomputers, 2016). Er moet een draad komen voor de plus kant van de spanning en voor de grond kant. Deze draden kunnen op het desbetreffende shields gesoldeerd worden, indien ze geen “connectoren” ingangen bezitten. Hiermee kan het prototype communicatie opzetten met de klimaatcomputer.

3.6 Hoe kan er middels de ontwikkelomgeving Visual Studio en de programmeertaal C# een applicatie gebouwd worden voor de Raspberry Pi 2 draaiend onder Windows IoT die een aangesloten shield aanstuurt?

Om deze deelvraag goed te kunnen beantwoorden, is deze deelvraag opgesplitst in meerdere sub-deelvragen. Aan de hand van deze sub-deelvragen moet uiteindelijk een antwoord komen op deze deelvraag

3.6.1 Wat is Windows IoT

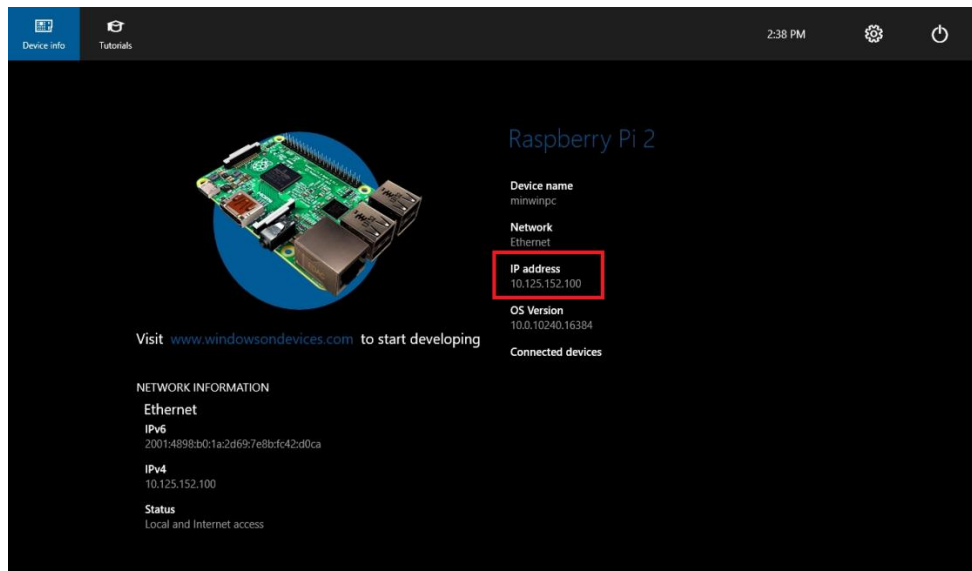
De afstudeerder heeft voor deze sub-deelvraag de volgende uitgangspunten gehanteerd: Wat houdt Windows IoT als een besturingssysteem in, hoe kan dit besturingssysteem geïnstalleerd worden op de Raspberry Pi 2, hoe kan dit besturingssysteem benaderd worden en indien nodig hoe kan de configuratie aangepast worden, voorts nog de vraag hoe het geregeld zit met ‘updates’ van het OS.

Bron: <https://developer.microsoft.com/en-us/windows/iot/win10/adafruitwelcome>

Windows Internet of Things (IoT) is een closed-source besturingssysteem die gebaseerd is op Windows 10 en dat geoptimaliseerd is voor kleinere apparaten. Windows IOT draait op de Raspberry Pi 2 en 3, Dragonboard 410c en MinnowBoard Max (What is Windows IoT Core, 2016). Door middel van Universal Windows Platform (UWP) API kunnen er applicaties gebouwd worden voor deze apparaten. Windows IOT ondersteunt de GPIO van de Raspberry Pi en biedt ook ondersteuning voor WIFI, Bluetooth, Camera, Audio USB apparaten. Daarnaast ondersteunt dit besturingssysteem met “de laatste release build 10.0.10586” ook de SPI en I2C interfaces volledig. Standaard libraries voor SPI en I2C komen mee met het OS. Door gebruik te maken van de Windows.Devices driver kan de SPI of GPIO van de Raspberry Pi aangesproken worden.

Bron: <https://developer.microsoft.com/en-us/windows/iot/win10/adafruitwelcome>

Windows IoT heeft geen volledige OS gebruikersinterface om mee te kunnen werken. Als het opgestart is, zie je de informatie van Raspberry Pi met bijbehorende IP adressen. Het is een minimale OS die ervoor moet zorgen dat er applicaties door middel van Visual Studio ontwikkeld en gedraaid kunnen worden.



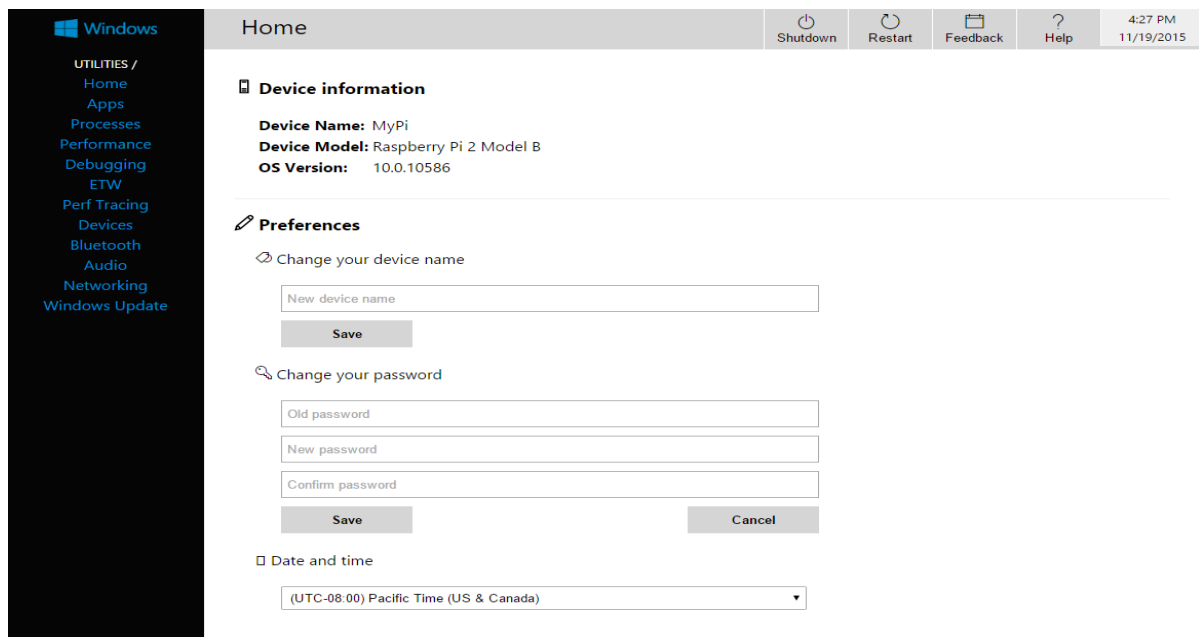
Figuur 10: Windows IoT "Default" Start Applicatie

Bron: <https://developer.microsoft.com/en-us/windows/iot/win10/kitsetupcrpi>

Om Windows IoT te kunnen gebruiken, moet Windows 10 (versie 10.0.10240 of hoger) besturingssysteem geïnstalleerd worden op de desbetreffende pc waar de ontwikkeling van de applicatie plaats gaat vinden (Windows IoT Starter Pack Contents, 2016). Op het Windows 10 besturingssysteem moet de "ontwikkelaar mode" ingeschakeld worden. Voor dit project moet de pc waarop de afstudeerder zijn werkzaamheden verricht, geüpdatet worden naar Windows 10 en moet de ontwikkelaar mode ingeschakeld worden.

Door middel van een online beschikbare tool van Microsoft kan IoT gedownload worden en op een SD-kaart van 8GB gezet worden. Voor de installatie is er minimaal een SD-kaart van 8GB benodigd. Deze SD-kaart wordt vervolgens in de Pi 2 gestopt en pas dan kan de Windows IoT besturingssysteem opgestart worden. Het opstarten van IoT gebeurt automatisch zodra Raspberry Pi een stroomtoevoer heeft (Windows IoT Starter Pack Overview, 2016).

Tevens kan via een browser de Windows Device Portal (WDP) benaderd worden. WDP biedt de mogelijkheid om de basisconfiguratie en management opties te kunnen configureren van de Raspberry Pi. Daarnaast kunnen de applicaties die op de Raspberry Pi draaien opgevraagd worden en "real-time" gedebugd worden. Ook kan hier handmatig gecheckt worden op de Windows Updates. Als er een update aanwezig is, word het automatisch geïnstalleerd. Dit kan niet handmatig "uitgezet" worden. Tevens kan hier ingesteld worden of een applicatie in de achtergrond moet draaien of niet.



Figuur 11: Windows Device Portal (WDP)

3.6.2 Wat is Visual Studio?

Visual Studio is een ontwikkel omgeving (development environment) voor Windows die uitgegeven is door Microsoft. Er kan taalafhankelijk geprogrammeerd worden. Kort samengevat bestaat deze software uit:

- Een editor om software te schrijven en grafische interface voor de ontwikkelde software te kunnen samenstellen.
- Een compiler om de software te controleren en te bouwen.
- Een debugger om de software te kunnen debuggen, indien er fouten zijn in de software.

Inmiddels heeft Microsoft Visual Studio versie 2015 uitgegeven. Deze versie ondersteunt ook het Windows IoT besturingssysteem. Dit houdt in dat met deze versie applicaties gebouwd en gedebugd kunnen worden op de Raspberry Pi 2 draaiend op Windows IoT.

Visual Studio versies onder de 2015 ondersteunen geen Windows IoT ontwikkeling. Daarnaast heeft Visual Studio 2015 ook een community editie die gratis te gebruiken is door studenten/individuele ontwikkelaars voor het bouwen van “niet-zakelijke” applicaties (Visual Studio 2015 product editions, 2016). Deze community editie bevat volledige “features” om applicaties mee te kunnen bouwen. Voor dit project is deze versie geschikt om een applicatie mee te kunnen bouwen.

Bron: <https://www.visualstudio.com/en-us/products/vs-2015-product-editions.aspx>

3.6.3 Welke typen applicaties zijn er mogelijk op Windows IoT?

Op Windows IoT is het mogelijk om een “headed” applicatie en een “headless” applicatie te draaien (Windows IoT Headed and Headless mode, 2016). “Headed” applicaties zijn applicaties waarbij er gebruik gemaakt wordt van een gebruikersinterface. Bij “Headless” applicaties zijn er geen gebruikersinterfaces aanwezig en draait de applicatie in de achtergrond.

“Headless” applicaties worden meestal gebruikt bij toepassingen waarbij de applicatie voortdurend in de achtergrond moet draaien zonder de tussenkomst van een persoon en waarbij een userinterface niet van toepassing is. Hierbij merkt de gebruiker niets van deze applicatie. Deze applicatie draait dan voortdurend als een achtergrond applicatie om bijvoorbeeld bepaalde acties te kunnen verrichten en treedt niet naar de voorgrond. Als het hele besturingssysteem crasht, dan starten deze “headless” applicaties automatisch op. Daarnaast is er bij dit applicatie type geen monitor of een toetsenbord/muis benodigd (Windows IoT Headed and Headless mode, 2016).

“Headed” applicaties worden meestal gebruikt voor systemen waarbij er door middel van een actie informatie getoond moet worden aan de gebruiker of door de ‘input’ (inbreng) van de gebruiker een actie ondernomen moet worden. Hierbij speelt de gebruikersinterface een belangrijke rol. Bij dit applicatie type is er een monitor benodigd en de toetsenbord/muis zijn optioneel afhankelijk van de toepassing.

Bron: <https://developer.microsoft.com/en-us/windows/iot/win10/headlessmode>

Deze applicatie typen kunnen ontwikkeld worden in C#, C++, JavaScript, VB, Node.js, Python programmeertalen. Daarnaast wordt het aangeraden door Microsoft om de applicatie in de Universal Windows App vorm te bouwen om deze te richten op verschillende platformen en niet op een platform te blijven hangen. Deze gebouwde applicatie kan later naar een ander platform omgezet worden (Ekuan, 2016).



Figuur 12: Universal Apps Platforms (Ekuan, 2016)

Bron 1: <https://msdn.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide>

Bron 2: <https://msdn.microsoft.com/en-us/windows/uwp/get-started/whats-a-uwp>

Bron 3: <https://developer.microsoft.com/en-us/windows/iot/faqs> (Which languages can be used to create UWP apps that can run on Windows 10 IoT Core?)

Bron 4: <https://developer.microsoft.com/en-us/windows/iot/faqs> (Are true console apps supported on Windows 10 IoT Core?)

3.6.4 Welke communicatie interfaces ondersteunt de Raspberry Pi 2 met Windows IoT?

Windows IoT besturingssysteem is relatief nieuw voor de Raspberry Pi 2. Dit besturingssysteem werd op 10 augustus uitgebracht voor de Raspberry Pi 2. Hierdoor moet er gekeken worden welke communicatie interfaces van de Raspberry Pi 2 ondersteund worden. De interfaces die officieel ondersteund worden door Microsoft zijn (Windows Dev Center, 2016):

- GPIO pinnen
- SPI
- I2C
- Serial UART
- USB

Bron 1: <https://developer.microsoft.com/en-us/windows/iot/win10/kitsetuppcrpi>

Bron 2: <https://developer.microsoft.com/en-us/windows/iot/win10/samples/PinMappingsRPi2.htm>

Bron 3: <http://tweakers.net/nieuws/104707/microsoft-haalt-windows-10-iot-core-uit-de-testfase.html>

Bron 4: <http://tweakers.net/nieuws/101142/microsoft-kondigt-windows-10-voor-raspberry-pi-2-aan.html>

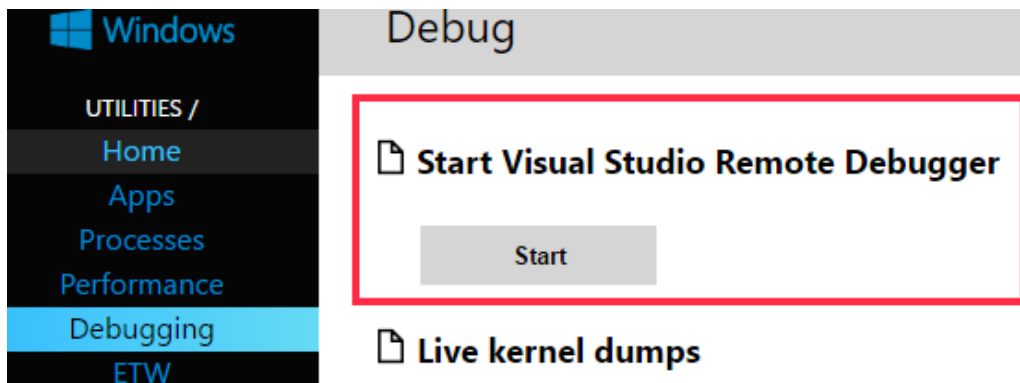
3.6.5 Hoe kunnen er door middel van Visual Studio applicaties gebouwd en gedebugd worden op het Windows IoT besturingssysteem?

Om applicaties te kunnen bouwen en te kunnen debuggen op de Windows IoT met behulp van Visual Studio zijn de volgende zaken alvast van tevoren benodigd (Windows IoT Starter Pack Contents, 2016):

- Een Raspberry Pi 2 draaiend op Windows IoT
- Windows Device Portal (WDP/IoT Dashboard wordt het ook genoemd).
- Een lokale internetverbinding naar de Raspberry Pi 2.
- Windows 10 (versie 10.0.10240 of hoger) op de ontwikkel PC waarbij de ontwikkel mode ingeschakeld moet worden.
- Visual Studio (VS) 2015 Community editie.

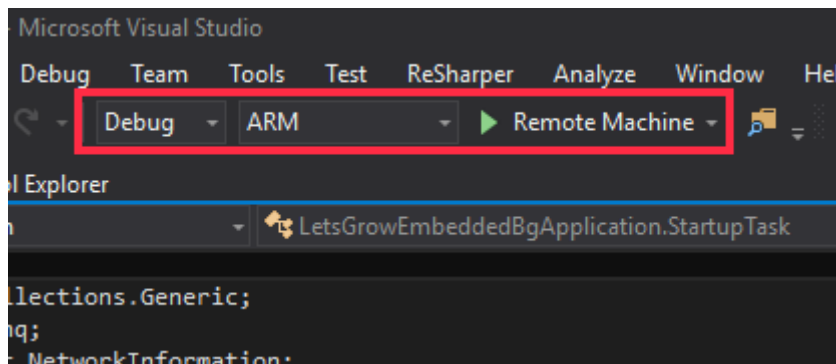
Bron: <https://developer.microsoft.com/en-us/windows/iot/win10/kitsetuppcrpi>

Als de bovengenoemde zaken goed uitgevoerd zijn, kan Visual Studio opgestart worden en een applicatie type gekozen worden. Nadat er een applicatie type gekozen is, kan de afstudeerder beginnen met het ontwikkelen van de applicatie code. Om tussentijds te kunnen debuggen moet eerst de Visual Studio Remote Debugger gestart worden via de Web interface van Windows IoT.

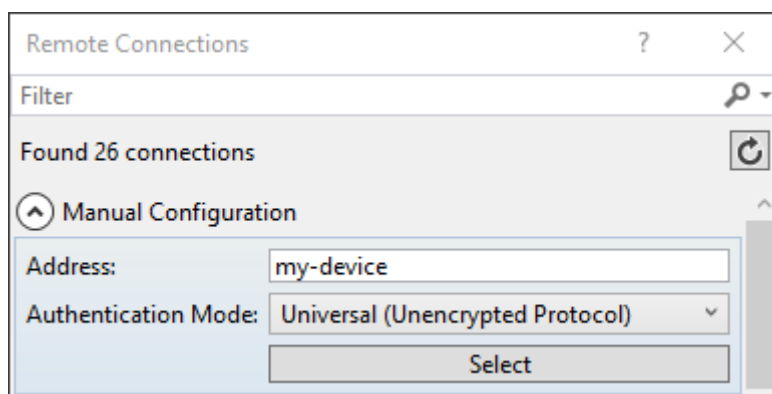


Figuur 13: Visual Studio Remote Debugger

Daarna moeten de juiste instellingen in VS ingesteld worden. De “Debug” mode en “ARM” platform moet er geselecteerd worden en daarna kan er op de “Remote Machine” knop gedrukt worden. Als het goed gaat, vindt VS de Raspberry Pi 2 automatisch. Deze kan dan geselecteerd worden en na het invoeren van authenticatie gegevens wordt de ontworpen applicatie op de Pi 2 gezet en gelijk opgestart. Nu kan er gedebugd worden.



Figuur 14: Visual Studio 2015 Community Editie



Figuur 15: Visual Studio 2015 CE "Remote Connection" opties

Voor het bouwen van de applicatie kunnen dezelfde stappen gevolgd worden. Alleen moet er dan in plaats van “Debug” mode, “Release” mode geselecteerd worden.

Zo kan er door middel van Visual Studio 2015 editie een applicatie gebouwd worden en ook gedebugd worden op Windows IoT.

Bron: <https://www.microsoft.com/en-us/download/details.aspx?id=48155>

3.6.6 Hoe kan er een applicatie gebouwd worden om de Input/Output van de Raspberry Pi 2 aan te kunnen sturen op het Windows IoT besturingssysteem?

Om de “input/output” van de Pi 2 te kunnen aansturen worden door Microsoft de Windows IoT “libraries” bijgeleverd. Door gebruik te kunnen maken van deze standaard meegeleverde libraries kunnen de ingangen/uitgangen aangestuurd worden. Deze libraries moeten aan de applicatie worden toegevoegd waarna de functies geïmplementeerd kunnen worden. Microsoft heeft van sommige communicatie interfaces voorbeeld applicaties beschikbaar.

Aan de hand van deze libraries kunnen de ingangen/uitgangen aangestuurd worden (Windows Dev Center, 2016):

Serial UART = *Windows.Devices.SerialCommunication*

SPI = *Windows.Devices.Spi*

I2C = *Windows.Devices.I2c*

GPIO = *Windows.Devices.Gpio*

Deze libraries moeten in de code geïmplementeerd worden. Daarnaast moeten specifieke functies aangeroepen worden. Al deze zaken zijn te vinden in de test applicatie die de afstudeerder heeft samengesteld uit diverse code voorbeelden die er waren. Om alle interfaces in een keer te testen heeft de afstudeerder de benodigde code in een test applicatie omgezet. Deze applicatie moet ervoor zorgen dat er communicatie opgebouwd kan worden via de desbetreffende interfaces. Daarnaast moeten deze interfaces aangestuurd kunnen worden.

Aan de hand van de onderstaande code kunnen de ingangen en uitgangen aangestuurd worden:

```

using System;
using Windows.Devices.Enumeration;
using Windows.Devices.Spi;
using Windows.Devices.Gpio;
using Windows.Storage.Streams;
using Windows.Devices.SerialCommunication;
using Windows.Devices.I2C;
using Windows.UI.Xaml.Controls;
using DisplayFont;

/* Initialize GPIO, SPI, and the display */
namespace Example_Code_Eray
{
    public sealed partial class MainPage : Page
    {
        public MainPage()
        {
            this.InitializeComponent();

            /* Initialize GPIO, SPI, and the display */
            InitAll();
        }

        private async void InitAll()
        {
            try
            {
                await InitGpio();           /* Initialize the GPIO controller and GPIO pins */
                await InitSpi();            /* Initialize the SPI controller */
                await InitI2C();            /* Initialize the I2C controller */
                await InitSerial();         /* Initialize the I2C controller */
            }
            catch (Exception ex)
            {
                Text_Status.Text = "Exception: " + ex.Message;
                if (ex.InnerException != null)
                {
                    Text_Status.Text += "\nInner Exception: " + ex.InnerException.Message;
                }
                return;
            }
        }
    }
}

```

Figuur 16: Applicatie voorbeeld code

Voor het GPIO communicatie aansturing kunnen deze functies gebruikt worden:

```

private async void InitGpio()
{
    try
    {
        IoController = GpioController.GetDefault(); /* Get the default GPIO controller on the system */

        /* Initialize a pin as output for the Data/Command line on the display */
        DataCommandPin = IoController.OpenPin(DATA_COMMAND_PIN);
        DataCommandPin.Write(GpioPinValue.High);
        DataCommandPin.SetDriveMode(GpioPinDriveMode.Output);

        /* Initialize a pin as output for the hardware Reset line on the display */
        ResetPin = IoController.OpenPin(RESET_PIN);
        ResetPin.Write(GpioPinValue.High);
        ResetPin.SetDriveMode(GpioPinDriveMode.Output);
    }
    catch (Exception ex)
    {
        Text_Status.Text = "Exception: " + ex.Message;
        if (ex.InnerException != null)
        {
            Text_Status.Text += "\nInner Exception: " + ex.InnerException.Message;
        }
        return;
    }
}

```

Figuur 17: Applicatie voorbeeld code GPIO Initialisatie functie

Voor het I2C communicatie en aansturing kunnen deze functies gebruikt worden:

```
public async void InitI2C()
{
    // Get a selector string for bus "I2C1"
    string aqs = I2cDevice.GetDeviceSelector("I2C1");

    // Find the I2C bus controller with our selector string
    var dis = await DeviceInformation.FindAllAsync(aqs);
    if (dis.Count == 0)
        return; // bus not found

    // 0x40 is the I2C device address
    var settings = new I2cConnectionSettings(0x40);

    // Create an I2cDevice with our selected bus controller and I2C settings
    using (I2cDevice device = await I2cDevice.FromIdAsync(dis[0].Id, settings))
    {
        byte[] writeBuf = { 0x01, 0x02, 0x03, 0x04 };
        device.Write(writeBuf);
    }
}
```

Figuur 18: Applicatie voorbeeld code I2C Initialisatie functie

Voor het Serial UART communicatie en aansturing kunnen deze functies gebruikt worden:

```
private async void InitSerial()
{
    string aqs = SerialDevice.GetDeviceSelector("UART0");           /* Find the selector string for the serial device */
    var dis = await DeviceInformation.FindAllAsync(aqs);             /* Find the serial device with our selector string */
    SerialDevice SerialPort = await SerialDevice.FromIdAsync(dis[0].Id); /* Create an serial device with our selected device */

    /* Configure serial settings */
    SerialPort.WriteTimeout = TimeSpan.FromMilliseconds(1000);
    SerialPort.ReadTimeout = TimeSpan.FromMilliseconds(1000);
    SerialPort.BaudRate = 9600;
    SerialPort.Parity = SerialParity.None;
    SerialPort.StopBits = SerialStopBitCount.One;
    SerialPort.DataBits = 8;

    /* Write a string out over serial */
    string txBuffer = "Hello Serial";
    DataWriter dataWriter = new DataWriter();
    dataWriter.WriteString(txBuffer);
    uint bytesWritten = await SerialPort.OutputStream.WriteAsync(dataWriter.DetachBuffer());

    /* Read data in from the serial port */
    const uint maxReadLength = 1024;
    DataReader dataReader = new DataReader(SerialPort.InputStream);
    uint bytesToRead = await dataReader.LoadAsync(maxReadLength);
    string rxBuffer = dataReader.ReadString(bytesToRead);
}
```

Figuur 19: Applicatie voorbeeld code Serial UART Initialisatie functie

Voor het SPI communicatie en aansturing kunnen deze functies gebruikt worden:

```
private async void InitSpi()
{
    try
    {
        var settings = new SpiConnectionSettings(SPI_CHIP_SELECT_LINE); /* Create SPI initialization settings */
        settings.ClockFrequency = 10000000; /* Datasheet specifies maximum SPI clock frequency of 10MHz */
        settings.Mode = SpiMode.Mode3; /* The display expects an idle-high clock polarity, we use Mode3 */
        /* to set the clock polarity and phase to: CPOL = 1, CPHA = 1 */

        string spiAqs = SpiDevice.GetDeviceSelector(SPI_CONTROLLER_NAME); /* Find the selector string for the SPI bus controller */
        var devicesInfo = await DeviceInformation.FindAllAsync(spiAqs); /* Find the SPI bus controller device with our selector string */
        SpiDisplay = await SpiDevice.FromIdAsync(devicesInfo[0].Id, settings); /* Create an SpiDevice with our bus controller and SPI settings */
    }
    catch (Exception ex)
    {
        Text_Status.Text = "Exception: " + ex.Message;
        if (ex.InnerException != null)
        {
            Text_Status.Text += "\nInner Exception: " + ex.InnerException.Message;
        }
        return;
    }
}
```

Figuur 20 Applicatie voorbeeld code SPI Initialisatie functie

De bovenstaande voorbeelden laten zien hoe de ingangen en uitgangen aangestuurd kunnen worden.

Bron: <https://developer.microsoft.com/en-us/windows/iot/win10/samples/pinmappingsrpi2>

3.7 Hoe kan er middels de ontwikkelomgeving MonoDevelop en de programmeertaal C# een applicatie gebouwd worden voor de Raspberry Pi 2 draaiend onder Raspbian die een aangesloten shield aanstuurt?

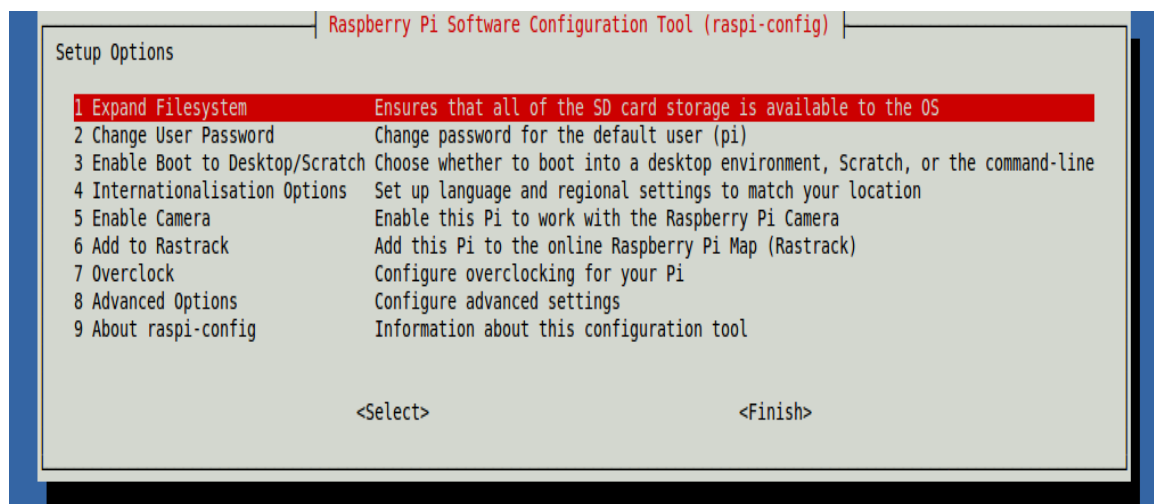
Om deze deelvraag goed te kunnen beantwoorden, is deze deelvraag opgesplitst in meerdere sub-deelvragen. Aan de hand van deze sub-deelvragen moet uiteindelijk een antwoord komen op deze deelvraag

3.7.1 Wat is Raspbian

De afstudeerder heeft voor deze sub-deelvraag de volgende uitgangspunten gehanteerd: Wat houdt Raspbian als een besturingssysteem in, hoe kan dit besturingssysteem geïnstalleerd worden op de Raspberry Pi 2, hoe kan dit besturingssysteem benaderd worden en indien nodig configuratie aangepast worden, hoe het geregeld zit met de 'updates'.

Bron: <https://www.raspbian.org/RaspbianFAQ>

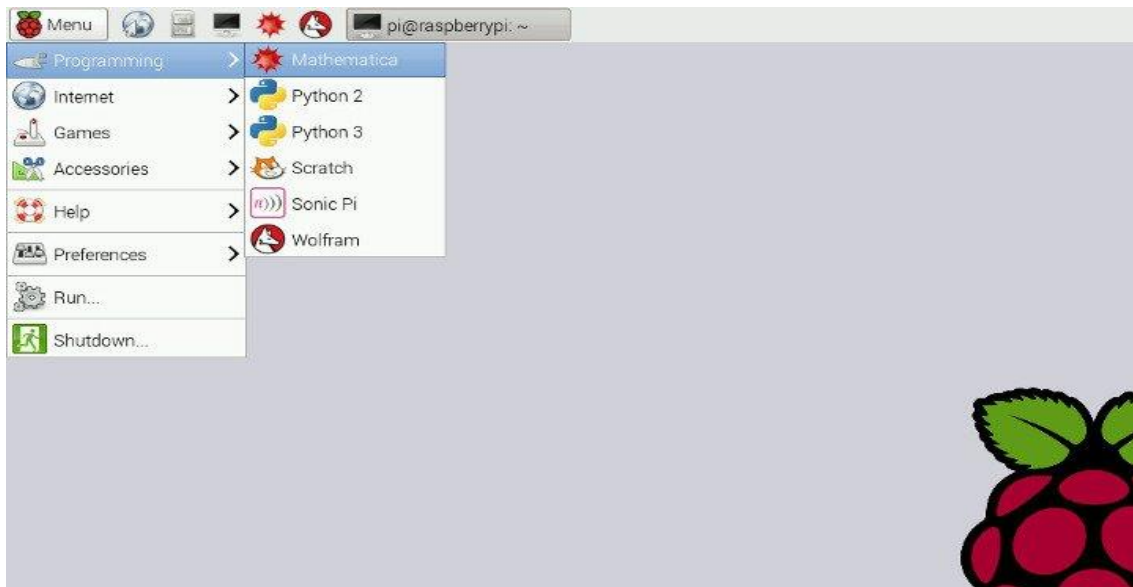
Raspbian is een open-source Linux variant besturingssysteem dat geoptimaliseerd is voor de Raspberry Pi (Harrington, 2015). Dit besturingssysteem wordt door het community actief onderhouden. Hierdoor komen er regelmatig updates uit. De laatste versie van de Raspbian is Jessie (met kernel 4.4). Verder wordt Raspbian geleverd met meer dan 35000 verschillende software pakketten. Daarnaast kan de basisconfiguratie van Raspbian gemakkelijk geconfigureerd worden door de raspi-config tool op te roepen. Hiermee kunnen verschillende opties aangepast worden, zoals het veranderen van wachtwoord, overklokken van de processor, aan- en uitschakelen van de SPI, I2C communicatie, updates enz.



Figuur 21: raspi-config

Bron: <https://www.raspberrypi.org/documentation/configuration/raspi-config.md>

Indien gewenst kan er ook naar een desktop GUI omgeschakeld worden. Dit kan door in de raspi-config tool de 3^e optie te kiezen. De Raspberry kan dan via een grafische interface met een muis, toetsenbord en een monitor bediend worden (Harrington, 2015).



Figuur 22: Raspbian User Interface

<https://www.raspberrypi.org/blog/changes-to-the-raspbian-user-interface/>

Om de Raspbian op een SD-kaart te krijgen, moet de SD-kaart een geheugen hebben van minimaal 4GB. Er wordt een “image” vanaf het internet gedownload en met behulp van SD-kaart tools kan dit image 1:1 overgezet worden op de SD-kaart. Vervolgens kan deze SD-kaart in de Raspberry Pi 2 geplaatst worden.

Bron: <https://www.raspberrypi.org/help/faqs/#sdCards>

Op de Raspbian kan er ontwikkeld worden met allerlei verschillende programmeertalen. Het hangt van de gebruiker af met welke taal hij wil gaan ontwikkelen. Ook hangt het van de beschikbaarheid van de libraries af of er met die specifieke taal benaderd kan worden. Indien de hardware niet benaderd kan worden met de gekozen specifieke taal, dan kan er ook een “wrapper” omheen gebouwd worden om toch voor de communicatie te kunnen zorgen.

Bron: <https://www.raspberrypi.org/help/faqs/#softwareLanguages>

Raspbian wordt actief ondersteund door zowel de Raspberry Pi Foundation als de community. Indien er iets mis gaat, kan er op internet vaak snel een oplossing gevonden worden, omdat er veel mensen mee werken en de kans is groot dat al iemand iets soortgelijks heeft ervaren of heeft ontwikkeld. Daarnaast krijgt Raspbian ook “security” updates. Deze “updates” kunnen handmatig door commando’s binnengehaald worden. Het is ook mogelijk om periodiek “unattended updates” binnen te halen met een script.

Bron: <https://www.raspberrypi.org/help/faqs/#softwareOS>.

3.7.2 Wat is MonoDevelop?

MonoDevelop is een ontwikkel omgeving voor Linux, Mac en Windows die vooral gericht is op software die gebruik maakt van de open-source implementatie van .NET framework (What is MonoDevelop, 2016). Met MonoDevelop kunnen applicaties ontworpen worden met voornamelijk C# en .Net gebaseerde programmeertalen. Voordeel van MonoDevelop is dat het op verschillende soorten platformen kan draaien. Als er met C# voor de Raspberry Pi 2 een applicatie ontworpen en gedraaid moet worden, dan moet er gebruik gemaakt worden van Mono.Net Framework met MonoDevelop als ontwikkel omgeving.

Bron: <http://www.mono-project.com/docs/about-mono/>

3.7.3 Welke typen applicaties zijn er mogelijk op Raspbian?

Op de Raspbian zijn er verschillende typen applicaties mogelijk variërend van een console-tot een normale applicatie met user interface. Deze kunnen gebouwd worden door de desbetreffende library te downloaden en te installeren en een ontwikkel omgeving te downloaden en te installeren. Van achtergrond applicatie tot een volledige applicatie met gebruikersinterface kan er gedraaid worden op de Raspbian. Er moeten scripts samengesteld worden en deze met de "Cron" tooling geconfigureerd worden om de applicatie als een geplande taak "periodiek" in de achtergrond te kunnen draaien. Verder ondersteunt Raspbian voor de C# programmeertaal console applicaties en GUI applicaties door middel van Graphical User Interface Toolkit (GTK#). Om C# applicaties mogelijk te maken op de Raspbian is het open-source Mono.Net Framework vereist. Deze moet handmatig geïnstalleerd worden.

Bron: <http://www.mono-project.com/docs/gui/gtksharp/>

3.7.4 Welke communicatie interfaces ondersteunt de Raspberry Pi 2 met Raspbian?

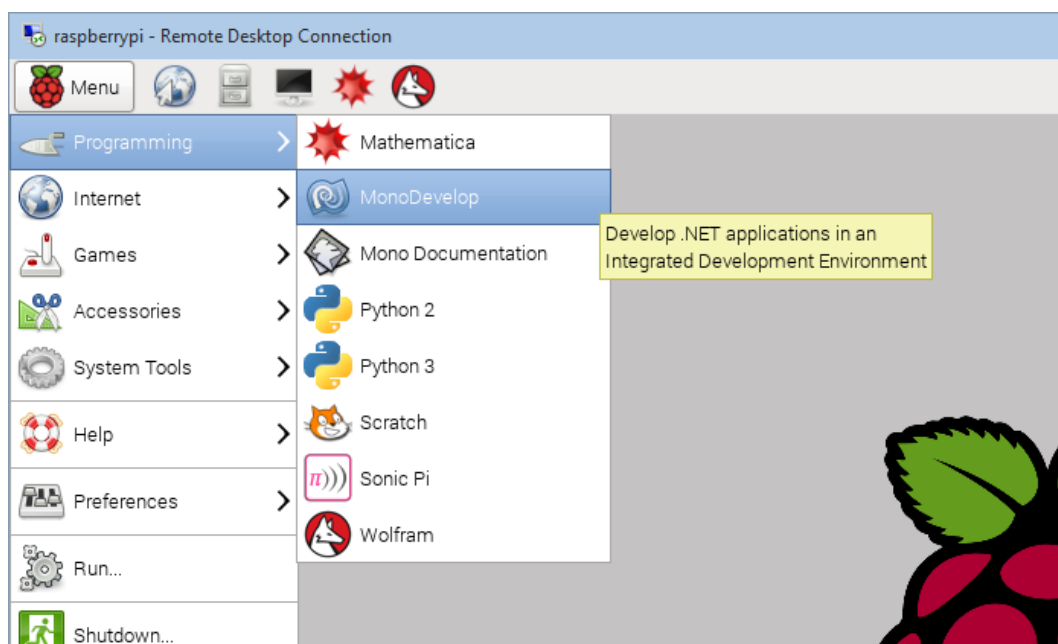
Het Raspbian besturingssysteem wordt officieel ondersteund door de Raspberry Pi Foundation. Hierdoor is dit besturingssysteem geoptimaliseerd voor de Raspberry Pi. Dankzij deze optimalisatie ondersteunt Raspbian alle mogelijke communicatie interfaces (SPI, I²C, Uart, PWM, USB) die er op de Raspberry Pi 2 te vinden zijn. Om sommige van deze communicatie interfaces te kunnen gebruiken, moeten de deze communicatie interfaces eerst ingeschakeld worden door middel van de "raspi-config" tool. Daarnaast moeten ook bepaalde "libraries" voor desbetreffende communicatie interfaces geïnstalleerd worden.

Bron: <https://www.raspberrypi.org/help/faqs/#softwareOS>

3.7.5 Hoe kunnen er door middel van MonoDevelop applicaties gebouwd en gedebugd worden op de Raspbian?

Om applicaties te kunnen bouwen en te kunnen debuggen met MonoDevelop moet eerst het open-source Mono.NET Framework geïnstalleerd worden. Nadat deze geïnstalleerd is, moet MonoDevelop geïnstalleerd worden op Raspbian. Als deze stappen uitgevoerd zijn, kan MonoDevelop opgestart worden om applicaties te bouwen en indien nodig te debuggen op Raspbian. De applicaties die gebouwd worden, kunnen bijvoorbeeld door Visual Studio geopend worden op een PC draaiend onder Windows. Dit maakt het heel handig om bijvoorbeeld met Visual Studio een applicatie te ontwerpen en over te zetten naar Raspbian. Op Raspbian kan dan deze applicatie geopend worden met MonoDevelop om zo verder te gaan met het bouwen/debuggen van de desbetreffende applicatie.

Bron: <http://www.mono-project.com/docs/getting-started/>



Figuur 23: Raspbian MonoDevelop

Bron: <http://j.tlms.be/2015/02/04/getting-started-with-the-raspberry-pi-2-for-net-developers/>

3.7.6 Hoe kan er een applicatie gebouwd worden om de Input/Output van de Raspberry Pi 2 aan te kunnen sturen op het Raspbian besturingssysteem?

Als er een applicatie gebouwd wordt met de C# programmeertaal, dan is kan de Raspberry# IO library (Raspberry# IO repository, 2016) van de Raspberry# community gebruikt worden om de GPIO, SPI en I2C ingangen en uitgangen aan te sturen. Daarnaast is er ook een RaspiSharp library (RaspiSharp, 2016) beschikbaar om GPIO, PWM, I2C en SPI ingangen en uitgangen te kunnen besturen. Tevens is er RaspberryPi.Net library (RaspberryPi.net, 2016) waarbij de SPI ingangen en uitgangen aangestuurd kunnen worden. Al deze libraries moeten dan in de applicatie geïmplementeerd worden om de ingangen en de uitgangen van de Pi2 te kunnen aansturen. Net zoals bij bij Windows IoT moeten door middel van deze libraries de benodigde functies aangeroepen worden en dan kunnen de ingangen en de uitgangen aangestuurd worden.

4. Conclusie Analyse hardware, besturingssystemen en ontwikkeltools.

De hoofdvraag voor de analyse is: *Hoe kunnen regelingen in klimaatcomputers beïnvloed worden zonder de tussenkomst van de betreffende klimaatcomputer leverancier?*

Dit kan gerealiseerd worden door middel van de onderstaande hardware/software componenten te gebruiken.

- Een “Power Supply” Micro-USB adapter voor de Raspberry Pi 2
- Een Micro SD-kaart van minimaal 8GB voor IoT en/of minimaal 4GB voor Raspbian
- Een Raspberry Pi 2 draaiend op het Windows IoT besturingssysteem en/of Raspbian besturingssysteem.
- Raspberry Pi 2 met een actieve internetverbinding door middel van een ‘UTP’ netwerkkabel.
- Een applicatie die geschreven is met programmeertaal C# en die de LetsGrow Web API library als een van de communicatie interface libraries heeft om met de ingangen en de uitgangen van de Pi 2 te kunnen aansturen.
- Een van de gevonden shields gekoppeld aan de Pi 2, die minimaal een spanning produceren tussen de 0-2V en/of 4-20mA met een 8-bit resolutie.
- Het prototype bestaand uit de bovengenoemde punten gekoppeld aan de “remote input/output module” van de klimaatcomputer.

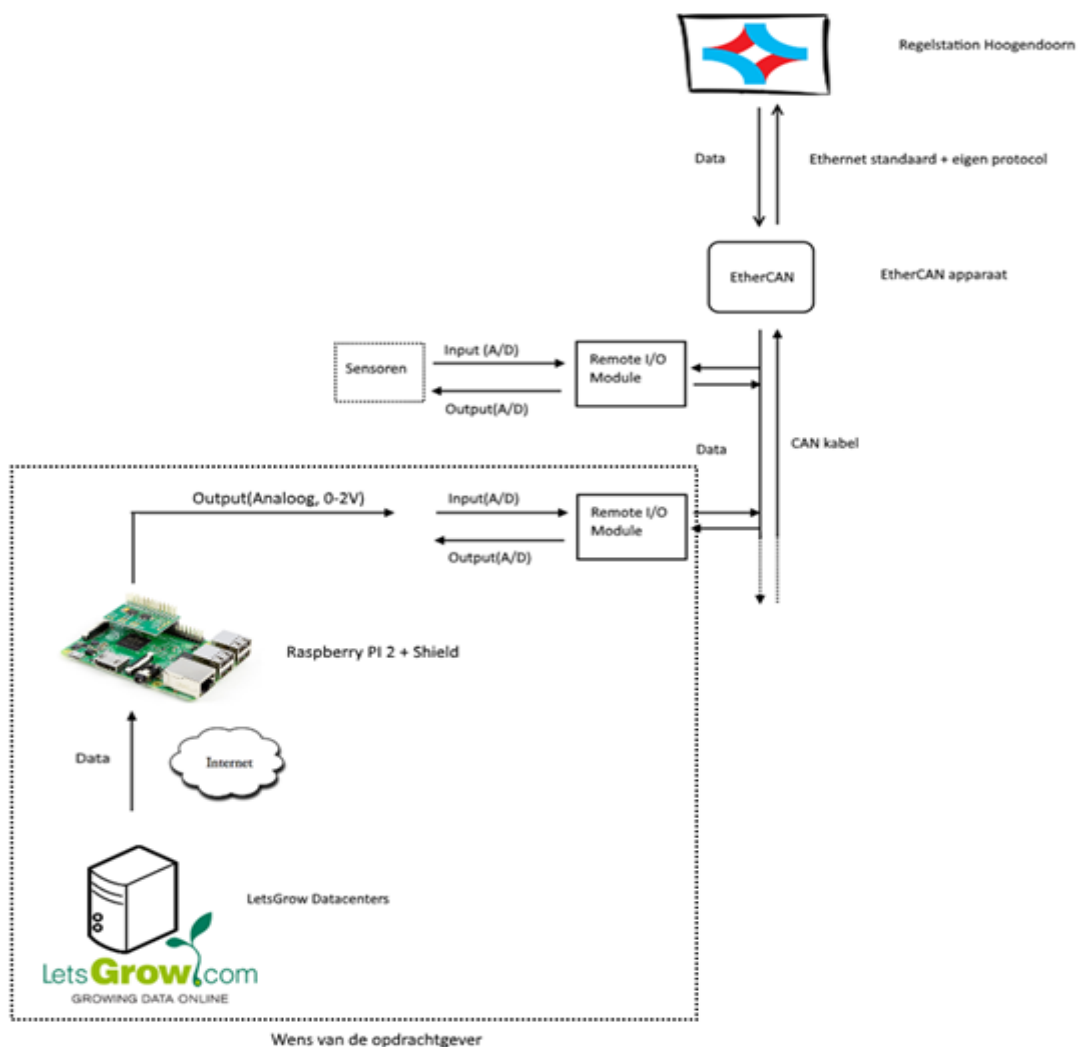
Uit de analyse blijkt dat het beïnvloeden van de regelingen op beide besturingssystemen gerealiseerd kan worden. Zowel Windows IoT als Raspbian ondersteunt allebei volledig de UART, GPIO, I2C, SPI en de USB interface. Op beide systemen is het ook mogelijk om “automatische updates” te kunnen uitvoeren. Bij Windows IoT kunnen de automatische updates “niet” uitgeschakeld worden, terwijl dit bij Raspbian wel kan. Daarnaast kunnen er op beide systemen achtergrond services(applicaties) gedraaid worden en is het ook mogelijk om met de programmeertaal C# te kunnen programmeren en te kunnen debuggen. Visual Studio is voor beide besturingssystemen te gebruiken om software mee te ontwikkelen. Alleen op Raspbian moet er Monodevelop met Mono.NET Framework geïnstalleerd worden, zodat de applicatie die gebouwd is met programmeertaal C# op de Raspberry Pi 2 gedebugd en uitgevoerd kan worden. Daarnaast is de support voor allebei de besturingssystemen goed te noemen. Microsoft brengt in korte periodes updates uit voor Windows IoT en Raspbian krijgt bijna elk jaar een nieuwe versie met allerlei security patches.

Om te kiezen welk besturingssysteem, ontwikkeltool, applicatie type en shield gebruikt gaat worden voor het prototype, moet de afstudeerder eerst de eisen en de wensen van de opdrachtgever en de bedrijfsmentoren te weten komen. Nadat de eisen en de wensen bekend zijn, kan er door de afstudeerder een keuze gemaakt worden welke van de bovengenoemde componenten voor het prototype gebruikt gaan worden.

5. De wensen van de opdrachtgever:

De wens van de opdrachtgever is om naar “alle” gangbare merken klimaatcomputers data te kunnen sturen, zodat LetsGrow.com de regelingen van deze klimaatcomputers kan beïnvloeden.

Om deze wens te kunnen realiseren, is er uit de analyse in hoofdstuk 3 gebleken dat het prototype een minimale spanning tussen de 0-2V met een minimale resolutie van 8-bit moet produceren. Het prototype moet aan de “input/output module” systeem van de klimaatcomputer gekoppeld worden door middel van bedradingen, zodat LetsGrow.com toch de mogelijkheid heeft om de regelingen te kunnen beïnvloeden zonder de tussenkomst van de klimaat leveranciers. Om deze situatie te verduidelijken, is hieronder een schematische tekening gemaakt van de “nieuwe situatie”.



Figuur 24: Wens van de opdrachtgever

6. De Requirements

In dit hoofdstuk zullen de functionele en niet-functionele eisen beschreven worden. Deze eisen zijn aan de hand van verschillende bijeenkomsten met zowel de opdrachtgever als de bedrijfsmentoren vastgesteld.

6.1 Functionele eisen:

1. Het prototype moet automatisch opstarten, nadat het prototype voorzien is van stroom.
2. Het prototype moet zelf het verkeer initiëren met de servers om firewall problemen te kunnen voorkomen.
3. Het prototype moet indien er geen actieve netwerkverbinding aanwezig is een minimale tijd van 60 minuten niets doen.
4. Het prototype moet door middel van een “unieke identificatienummer” bij de servers authenticeren.
5. Het prototype moet zelfstandig/automatisch zijn werking(en) uit gaan voeren, zonder de tussenkomst van een persoon.
6. Het prototype moet om de 5 minuten met de LetsGrow.com servers connectie maken om gegevens te kunnen ophalen.
7. Het prototype moet zijn configuratie informatie sturen naar de servers, indien deze niet aanwezig is.
8. Het prototype moet “minstens een” analoge waarde kunnen sturen naar een klimaatcomputer.
9. De webserver moet een onbekende “unieke identificatienummer” loggen in zijn logging database.
10. Het prototype moet aan de hand van zijn “leds” een indicatie kunnen tonen van zijn interne status.

6.2 Niet-functionele eisen:

1. De hardware van het prototype moet minimaal mogelijk soldeerwerk bevatten.(minimaal in de zin van: het soldeerwerk moet beperkt blijven tot het solderen van headers/pinnetjes en kabeltjes op desbetreffende uitgangen; geen weerstanden/condensatoren).
2. De hardware van het prototype moet binnen 5 werkdagen geleverd kunnen worden.
3. Het prototype moet zijn besturingssysteem periodiek stil “unattended” updaten.
4. De hardware van het prototype moet van een bekende leverancier komen.
5. Een hardware onderdeel van het prototype mag niet meer dan 100 euro kosten.
6. De applicatie van het prototype moet gebouwd worden in Visual Studio met de programmeertaal C#.
7. Het prototype moet draaien op Windows IoT besturingssysteem.
8. Het prototype moet gebouwd worden met Visual Studio.
9. De applicatie code van het prototype moet voldoen aan de ifSQ richtlijnen en de “naming-guidelines” richtlijnen van Microsoft.
10. Het prototype moet 24 uur per dag draaien.
11. Configureren van itemcodes moeten via de LetsGrow.com website gebeuren.

7. De keuze voor het besturingssysteem, ontwikkeltool en applicatie type.

Uit de analyse uit hoofdstuk 3 blijkt dat het prototype met beide besturingssystemen gerealiseerd kan worden. Met beide besturingssystemen kunnen de communicatie interfaces van de Raspberry Pi 2 aangesproken worden. Beide besturingssystemen bevatten ontwikkeltools en de mogelijkheid om in de programmeertaal C# te kunnen bouwen en te kunnen debuggen. Daarnaast kunnen er op beide besturingssystemen verschillende soorten type applicaties gebouwd worden.

Deze zaken kort samengevat in de onderstaande tabel geeft de volgende weer:

	Windows IoT	Raspbian
Ondersteunt het besturingssysteem de GPIO, SPI, I2C, USB interface communicatie van de Raspberry Pi?	Door middel van de Windows.Devices.Spi/I2C/Serial Communication en/of Windows.Devices.Gpio libraries die standaard geleverd wordt met Windows IoT kunnen met de communicatie interfaces communicatie opgebouwd worden. USB ondersteuning is ook mogelijk.	Er moet eerst in de raspi-config de I2C, SPI enz. aangezet worden. Daarna kan er doormiddel van RaspiSharp/RaspberryPi.net/ Raspberry# communicatie opgebouwd worden met de desbetreffende communicatie interface. USB ondersteuning is ook mogelijk.
Is het mogelijk om automatisch updates te kunnen uitvoeren op het desbetreffende besturingssysteem?	Windows IoT voert de automatische updates altijd uit. Deze kan jammergenoeg niet uitgezet worden. Handmatig kan er wel gecheckt worden op updates.	Het is mogelijk om met scripts en cronjobs "unattended updates" periodiek te kunnen uitvoeren.
Welke applicatie types ondersteunt desbetreffend besturingssysteem?	Windows IoT heeft ondersteuning voor "Headless en "Headed" applicatie types die in formaat UWP (Universal Windows App) zijn.	Raspbian ondersteunt van achtergrond applicaties tot applicaties met gebruikersinterface. Het hangt van de ontwikkeltools en Framework af welke gebruikt kunnen worden. Om C# programma's te kunnen draaien, moet open-source Mono. NET Framework geïnstalleerd worden.
Is er ontwikkelsoftware beschikbaar voor het desbetreffende besturingssysteem?	Ja, Visual Studio om te bouwen en te debuggen.	Ja, Monodevelop om C# code te kunnen bouwen en te kunnen debuggen. Daarnaast kan er extern ook andere ontwikkeltools installeert worden voor alle andere programmeertalen.

Is het mogelijk om met de programmeertaal C# te kunnen programmeren/debuggen op het desbetreffende besturingssysteem?	Windows IoT ondersteunt VB.net, C#, C++, Python en Javascript	Op de Raspbian kan er ontwikkeld worden met allerlei verschillende programmeertalen. Het hangt van de gebruiker af met welke taal hij wil gaan ontwikkelen. Het is ook mogelijk om met C# programmeertaal te kunnen programmeren. Hiervoor moeten wel extra Frameworks geïnstalleerd worden.
Heeft het besturingssysteem support van een bedrijf/community?	Support van Microsoft. Ze brengen ook regelmatig updates uit.	Support van open community en in beperkte mate van de Raspberry Pi Foundation.
Voorkeur bedrijfsmentoren?	Bedrijfsmentoren hebben een voorkeur voor Windows IoT	Bedrijfsmentoren hebben een voorkeur voor Windows IoT

Groen gekleurd: voldoet volledig aan de eis.

Geel: voldoet niet aan de eis.

Uit de niet functionele eisen blijkt dat de bedrijfsmentoren een voorkeur hebben voor de Microsoft omgeving en ze willen dan ook graag Windows IoT met Visual Studio als ontwikkelomgeving gebruiken voor het prototype. Vanuit dit oogpunt is bij dit prototype gekozen om gebruik te maken van het Windows IoT besturingssysteem. Daarnaast is het ook handig om alleen Visual Studio te gebruiken als ontwikkeltool, omdat de bedrijfsmentoren hier al mee werken. Ze hoeven zich hierdoor niet te verdiepen in MonoDevelop.

Tevens is er gekozen om gebruik te maken van een “Headless” Universal App applicatie type voor het prototype, omdat het prototype zonder tussenkomst van personen en klimaatcomputer leveranciers zijn werking moet doen en hiervoor heeft de applicatie dus geen gebruikersinterface nodig. Ook brengt Universal App voordelen met zich mee. De ontworpen applicatie kan later op andere platformen gedraaid worden, zonder dat er drastische code wijzigingen doorgevoerd moeten worden

8. Vervolg Analyse shields

Het tweede gedeelte van het prototype bestaat uit een shield dat het mogelijk moet maken om de digitale gegevens te kunnen omzetten naar analoge voltages/mille ampère om deze door te kunnen sturen naar een klimaatcomputer. Uit de analyse uit hoofdstuk 3 blijkt dat de volgende shields beschikbaar zijn om de regelingen in de klimaatcomputers te kunnen beïnvloeden:

- 4-20 mA T click shield van Mikroe Elektronika.
- ADC-DAC Pi van AB electronics UK.
- Expander Pi van AB electronics UK.
- Phidget Analog 4-Output van Phidgets.com.
- 4-20 mA sensor board voor Arduino, Raspberry Pi en Galileo van Cooking Hacks
- UniPi board van UniPi.Technology.
- Custard Pi 2 van sf-innovations.co.uk
- Open Source Analog Input/Output Expansion Board van Osexpansionboard.de

Om een verdere analyse te kunnen maken, moeten uit deze beschikbare shields een selectie gemaakt worden om ingekocht te worden. Om de selectie te kunnen maken is afstudeerder uitgegaan van de niet-functionele eisen waar het prototype aan moet voldoen. Uit de niet-functionele eisen komen de onderstaande zaken naar voren:

1. Shield moet minimaal één analoog uitgang hebben van 0-2V of 4-20mA.
2. Shield moet maximaal binnen 5 werkdagen leverbaar kunnen zijn
3. Shield mag niet meer dan 100 euro kosten.
4. Shield moet zo simpel mogelijk gekoppeld kunnen worden aan de Raspberry Pi 2 (met een USB kabel of door middel van pinnetjes).
5. Shield moet van een bekende (te vertrouwen) leverancier komen.
6. Shield mag de volgende interfaces hebben SPI, I2C, UART, GPIO en USB

De uitgangspunten 1 en 4 spreken voor zich. Voor uitgangspunt 2 is de levertijd genomen van de officiële leverancier. Er is geen gebruik gemaakt van wederverkopers omdat hiervan de levertijden in de regel onbetrouwbaar zijn. Immers zijn deze wederverkopers op hun beurt weer afhankelijk van het hoofd leverancier. Deze levertijd is bepaald door het desbetreffende shield in het winkelmandje te zetten en te kijken wat er als levertijd op het scherm komt te staan. Voor uitgangspunt 3 geldt ook een soortgelijke beredenering. De prijzen kunnen namelijk per wederverkopers verschillen. Daarnaast gelden de genoemde prijzen voor alleen de shields. Dit houdt in dat als er een “tussenshield” nodig is, dat de totale prijs per shield kan verschillen. Ook speelt de wisselkoers een grote rol bij de prijsbepaling. Er is dus uitgegaan van de officiële leverancier prijzen.

Bij uitgangspunt 5 is er uitgegaan van de Een medewerker van Hoogendoorn die onder andere verantwoordelijk is voor de inkoop van hardware. Hij had een lijst met allerlei leveranciers. Er wordt een controle uitgevoerd worden met deze lijst of de leveranciers voorkomen en vertrouwd zijn.

Daarnaast kwam er ook uit de analyse dat de SPI interface sneller is dan de I2C interface. Uit de functionele eisen blijkt dat er om de 5 minuten data gevraagd wordt en weggeschreven gaat worden. Deze data moet minimaal een waarde voor een analoge uitgang bevatten. Omdat de applicatie een meting doorstuurt mag deze data een paar seconden vertraging hebben. Hierdoor voldoen allebei de interfaces qua snelheid aan deze eis. Daarnaast is er bij SPI een limiet aan het aantal apparaten dat de SPI bus kan accepteren. Deze limiet is niet van toepassing voor het prototype, omdat er maar een device met een shield gebruikt gaat worden. Er gaat wel bekeken worden welke interface het desbetreffende shield heeft, maar deze interface zal geen invloed hebben op de uiteindelijke shield keuze.

Aan de hand van de 6 uitgangspunten en verdere selectie is de volgende tabel opgesteld:

	Uitgangspunt 1	Uitgangspunt 2	Uitgangspunt 3	Uitgangspunt 4	Uitgangspunt 5	Uitgangspunt 6
4-20 mA T click	1 Analooq uitgang van 4-20 mA	Leverbaar in 2-3 dagen	\$26.00	Klikbaar, maar vereist wel een tussenshield	MikroElektronica - bekend	SPI
ADC-DAC Pi	2 analoge uitgangen van 0-2V	Leverbaar in 1-2 dagen	£9.98	Klikbaar op de GPIO	Abelectronics - bekend	SPI
Expander PI	2 analoge uitgangen van 0-2V	Levertijd onbekend	£19.99	Klikbaar op de GPIO	Abelectronics - bekend	SPI
Phidget Analog 4-Output	4 analoge uitgangen van 0-10V	Leverbaar in 3-5 dagen	\$90.00	Door middel van USB kabel	Phidgets.com - bekend	USB

4-20 mA sensor board	4 analoge uitgangen van 4-20 mA. NOTE De analoge uitgangen vereisen externe spanningsbron op de Pi 2	Leverbaar in 5-7 dagen	€46.00	Klikbaar, maar vereist wel een tussenschild	Cooking Hacks - bekend	I2C
UniPi board	1 analoge uitgangen van 0-10V	Leverbaar in 1 dag	€149	Door middel van PIN connectoren (flat kabels)	UniPi technology - onbekend	I2C
Open Source Analog Input/Output Expansion Board 1	8 analoge uitgangen van 0-10V of 0-24V	Nog niet leverbaar.	Nog in ontwikkeling. Er is geen prijs aan vastgesteld	Door middel van PIN connectoren. (flat kabels)	Osexpansionboard.de - onbekend	SPI
Custard Pi 2	2 analoge uitgangen van 0-3.3V	Levertijd onbekend	£19.	Klikbaar op de GPIO	SF Innovations - onbekend	SPI

Groen gekleurd: voldoet volledig aan de eis.
Geel: voldoet gedeeltelijk/niet aan de eis.

8.1 Verdere analyse geselecteerde shields.

In bovenstaande tabel is af te lezen dat de volgende 3 shields aan de uitgangspunten voldoen. Daarnaast kunnen deze shields in korte tijd geleverd worden:

- 4-20 mA T click shield van Mikroe Elektronika.
- ADC-DAC Pi van AB electronics UK.
- Phidget Analog 4-Output van Phidgets.com.

De Expander Pi shield komt van dezelfde leverancier als het ADC-DAC Pi shield. Aan de hand van de informatie pagina van Expander Pi blijkt dat ze ook dezelfde componenten voor het aansturen van analoge uitgangen gebruiken, namelijk de Microchip MCP4822 component. Deze component wordt ook in de ADC-DAC Pi gebruikt en dit shield maakt ook van de zelfde library gebruik als de ADC-DAC Pi. Hierdoor maakt het uiteindelijk niet uit of de ADC-DAC Pi wordt of de Expander Pi. Alleen de levertijd van de Expander Pi is op het moment van schrijven onbekend. Hierdoor voldoet hij niet aan het uitgangspunt 2 en kan niet besteld worden om verder geanalyseerd te worden.

De 4-20 mA sensor board shield voldeed qua prijs aan het gestelde uitgangspunt 2, alleen de analoge uitgangen moeten extern door een voeding “gevoed” worden. Anders werken de analoge uitgangen niet met de referentiestroom die Raspberry Pi levert, omdat de Pi niet genoeg stroom levert om deze analoge uitgangen te kunnen laten werken. Dit zorgt ervoor dat de shield niet volledig compatible met de Raspberry Pi. Er wordt van uitgegaan dat er zonder “extra” onderdelen de shield toch een minimale voltage kan geven van 0-2V. Hierdoor voldoet deze shield niet volledig aan het uitgangspunt 1. Tevens is de levertijd aan de lange kant.

De UniPI is niet echt een shield te noemen. In plaats van dat de shields op de Raspberry Pi klikt wordt, moet de Raspberry Pi juist op de UniPI geklikt worden. Dit is in principe geen probleem alleen is de prijs dermate hoog dat hij niet voldoet aan uitgangspunt 3. Daarnaast is de leverancier onbekend bij het bedrijf.

Open Source Analog Input/Output Expansion Board is door een individueel persoon ontworpen. Het bordje bevindt nog in ontwikkel fase, waardoor het bordje niet gekocht kan worden. Het is een hobby-projectje en hiervoor is hij niet geschikt voor het prototype.

Custard Pi 2 is ook een shield waarmee analoge uitgangen tussen de 0-3V (12-bit) gegeneerd kunnen worden. Alleen is er de levertijd van dit shield onbekend. Daarnaast is de leverancier onbekend bij LetsGrow.com en hierdoor valt hij af.

De onderstaande 3 shields zijn ingekocht om verder geanalyseerd te worden:

- 4-20 mA T click shield van Mikroe Elektronika.
- ADC-DAC Pi van AB electronics UK.
- Phidget Analog 4-Output van Phidgets.com.

In de verdere analyse zijn de volgende uitgangspunten door de afstudeerder vastgesteld:

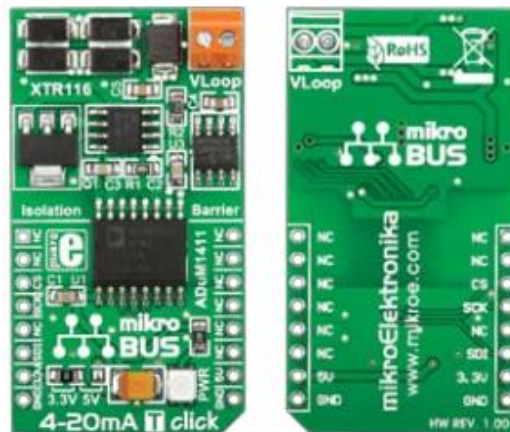
1. Heeft de leverancier zelf een library/voorbeeld applicatie uitgegeven in een C/C# of Python programmeertaal en kunnen met deze library de analoge uitgangen aangestuurd worden?
2. Indien er geen libraries beschikbaar zijn, kan er door standaard library die het desbetreffende besturingssysteem bezit, communicatie opgebouwd worden met het desbetreffende shield om de analoge uitgangen aan te sturen?¹
3. Is er een officiële ondersteuning voor Windows IoT/Raspbian.
4. Haalt het desbetreffende shield ook in praktijk de minimale 8-bit resolutie?²

1. Er wordt voor de standaard SPI library gebruik gemaakt van een eigen opgestelde voorbeeld code/test programma. Deze test programma wordt op de Raspbian (met de een van de C# Sharp IO libraries) en Windows IoT (met de Windows.Devices libraries) getest. Per shield wordt er gekeken of er een waarde weggeschreven kan worden naar de desbetreffende shield. Uiteindelijk wordt er met een multimeter gecontroleerd of het ook daadwerkelijk is weggeschreven.

2. De nauwkeurigheid wordt bepaald aan de hand van een multimeter. Deze multimeter wordt op 4000 mV gezet, zodat er bepaald kan worden of de shield daadwerkelijk een resolutie heeft van minimaal 8-bit.

8.1.1 4-20 mA t click shield

De 4-20 mA T click shield is een 12-bit “addon-bordje” voor verschillende mikroBUS form factor socket toepassingen. Het heeft een ADuM1411 Quad-channel digitale isolater, MCP4921 12-bit Digital/Analog Converter (DAC) en XTR116 4-20mA current loop transmitter aan boord. De communicatie vindt plaats via de 3 mikroBUS SPI lijnen(SDI,SCK,CS).



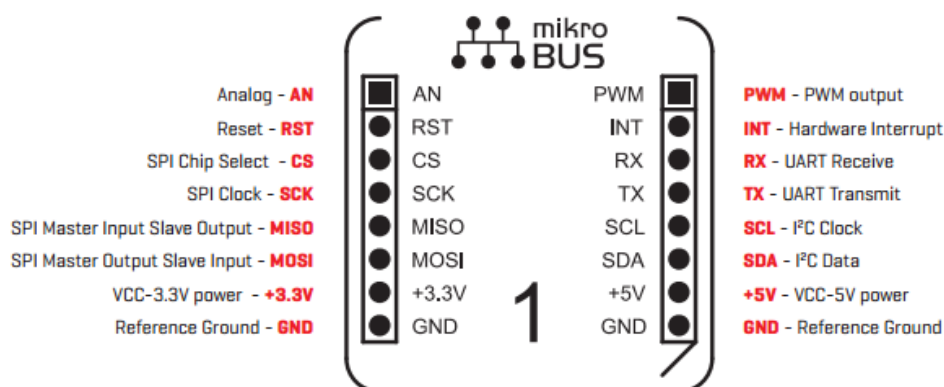
Figuur 25: 4-20 mA t click shield

Het shield is ontworpen om gebruik te kunnen maken van de 3.3V en de 5V stroomvoorziening van de Raspberry Pi. Standaard wordt het shield geleverd met een SMD jumper die gesoldeerd is op 3.3V. Om gebruik te kunnen maken van 5V, dient deze jumper verplaatst en gesoldeerd te worden op de 5V positie.



Figuur 26: SMD Jumper

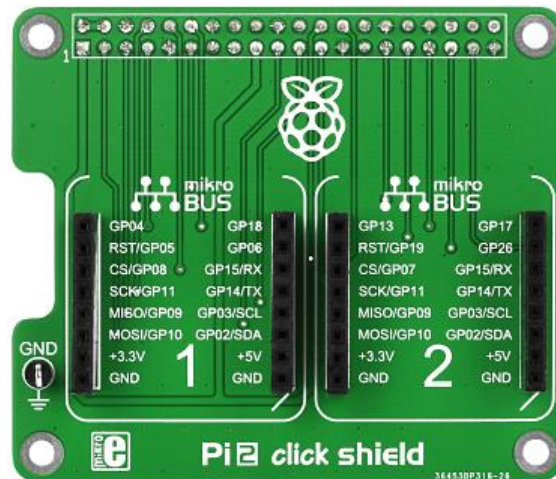
Om deze shield met een Raspberry Pi te kunnen koppelen, moet er gebruikt worden gemaakt van een “tussenbord” Pi 2 click shield die de mikroBUS socket pinconfiguratie bezit. De mikroBUS socket heeft 2 x 8 “headers” pinnen met een eigen pinconfiguratie.



Figuur 27: Schema pinconfiguratie

Zoals aan de bovenstaande pinconfiguratie te zien is, kan er door middel van een mikroBUS met verschillende soorten data standaarden gewerkt worden. Indien de shield die hierop geklikt is een SPI, I2C, UART of PWM standaard ondersteund, dan kan er door middel van de pin configuratie en de GPIO met de Raspberry Pi gecommuniceerd worden.

De Pi 2 click shield bezit twee van de bovenstaande mikroBUS sockets naast elkaar.



Figuur 28: Pi 2 click shield

Voordat de 4-20mA shield hierop geklikt kan worden, moet er eerst gesoldeerd worden. De “header pinnen” moeten nauwkeurig gesoldeerd worden. Daarna kan er eenvoudig een shield op een van de mikroBUS sockets geklikt worden.



Figuur 29: header pinnen

Nadat alles correct op elkaar aan is gesloten, kan door middel van de volgende pinconfiguratie de data-overdracht plaatsvinden:

- Chip Select (CS)/GP08 op de shield maakt connectie met de pin nummer 24 van de Raspberry pi
- Spi Clock (SCL)/GP11 op de shield maakt connectie met de pin nummer 23 van de Raspberry pi
- Master in Slave Out(MISO)/GP09 op de shield maakt connectie met de pin nummer 21 van de Raspberry pi
- Master Out Slave In(MISO)/GP10 op de shield maakt connectie met de pin nummer 19 van de Raspberry pi

Om dit shield aan te spreken moet er gebruik gemaakt worden van de standaard SPI library die het desbetreffende besturingssysteem bezit. MicroElektronica heeft geen library beschikbaar. Het uiteindelijke prototype met deze shields ziet er zo uit:



Figuur 30: Pi 2, Pi 2 Click Shield en 4-20mA click shield samen

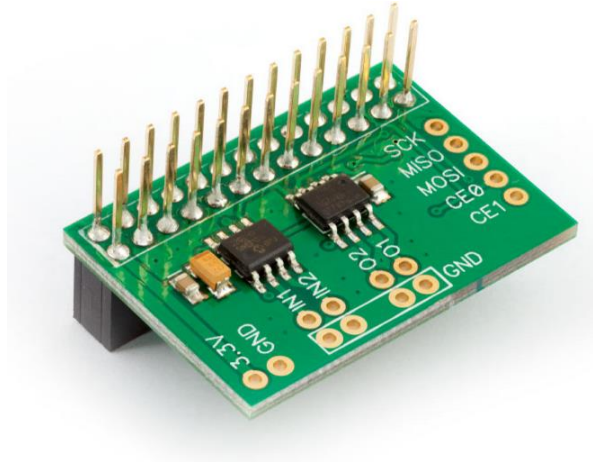
Op de website van Mikro elektronika wordt er geadverteerd dat de Raspberry Pi ook goed ondersteund wordt met behulp van het tussenshield Pi 2 Click, alleen blijkt er dat er helemaal geen libraries en SDK's aanwezig zijn om de 4-20 mA T click te kunnen aanspreken. Wat er wel te vinden is, is gericht op hun eigen MikroBus Pro software en daarvoor moet er dan weer extra betaald worden. Er moet met het standaard SPI protocol gewerkt worden om deze shield te kunnen aanspreken, alleen in de praktijk is het gebleken dat het niet werkt.

Tijdens het testen is het op zowel Raspbian als op Windows IOT “niet” gelukt om met voorbeeld code en de standaard SPI libraries communicatie op te kunnen bouwen met dit shield en een spanning te kunnen tonen. Er is op internet gezocht naar een “soortgelijke” voorbeeld applicaties, maar deze werkten niet met dit shield. Tevens is geprobeerd om met de support afdeling in contact te komen. Alleen verwees de live chat support medewerker naar de voorbeeld applicaties op de website die voor hun software bedoeld waren. Ook bleek tijdens de live chat support dat ze de Pi helemaal niet ondersteunen met hun shields. Het enige shield dat officieel een ondersteuning had was de Pi Click shield 2, maar dit is dan ook een leeg shield met 2 sockets erop. De shields die hierop geklikt kunnen worden, worden niet officieel voor de Pi ondersteund. Dit was wel een zeer opvallend nieuwtje en lijkt op misleiding door Mikro elektronika.

Dit shield is opnieuw besteld om hardwarematige beschadigingen uit te sluiten. De tweede test met deze shield gebeurde op dezelfde manier als eerste. Jammer genoeg kon er met het tweede shield ook geen connectie opgebouwd worden. Er kan geconcludeerd worden dat er geen hardware matige defects zijn, maar dat het shield softwarematig “op dit moment” niet werkend te krijgen is.

8.1.2 ADC-DAC Pi shield

De ADC-DAC Pi is een shield die speciaal ontworpen is voor de Raspberry Pi. Dit shield heeft 12-bit 2 analoge ingangen met een maximum 3.3V en 12bit 2 analoge uitgangen met een maximale output van 0-2V. De ingangen worden gerealiseerd door de MCP3202 A/D converter chip en de uitgangen door MCP4822 D/A chip.



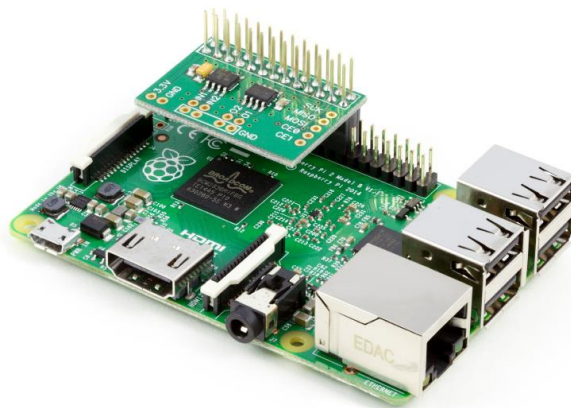
Figuur 31: ADC-DAC Pi shield

Voor dit shield is er geen “extensie” shield nodig. Deze kan gelijk op de Raspberry Pi pinnetjes geklikt worden. De ADC-DAC Pi shield wordt op de eerste 26 GPIO pinnen van de Raspberry Pi 2 gezet. De data communicatie vindt plaats door middel van de SPI standaard. De stroomvoorziening krijgt hij van de Raspberry Pi.

Ook wordt met dit shield dezelfde pinnummering aangesproken zoals bij de 4-20mA het geval was.

- Chip Select (CS) maakt connectie met de pin nummer 24 van de Raspberry Pi 2
- Spi Clock (SCL) maakt connectie met de pin nummer 23 van de Raspberry Pi 2
- Master in Slave Out(MISO) maakt connectie met de pin nummer 21 van de Raspberry Pi 2
- Master Out Slave In(MISO) maakt connectie met de pin nummer 19 van de Raspberry Pi 2

Het onderstaande plaatje laat zien hoe het prototype er uiteindelijk met dit shield uit ziet.



Figuur 32: Pi 2 en ADC-DAC Pi samen

Op de website van ABelektronics UK kunnen de libraries gedownload worden voor de Raspberry Pi. Ze hebben voor de Raspberry Pi libraries die geschreven zijn in C, C#, Python 2,3 en beschikbaar zijn voor Windows IoT en Raspbian. Daarnaast hebben ze ook voorbeeld applicaties die voor de Raspberry Pi beschikbaar zijn. Ook hun support is goed te noemen. Er wordt gericht gereageerd op de gestelde vragen in hun forums

Tijdens het testen is het gelukt om op zowel Raspbian als Windows IoT met de voorbeeld applicatie die door ABelektronics geleverd wordt een spanning te kunnen genereren. Op Raspbian moeten echter wel op extra libraries(Raspberry # IO) geïnstalleerd worden. Daarnaast moet de ADC-DAC library met een DLL import functie in de C# project toegevoegd worden om de code in C# te kunnen schrijven. Voor Python programmeertaal is er wel de mogelijkheid om alles native te doen. Op Windows IoT hebben ze “native” library beschikbaar voor C# programmeertaal.

Er kan een spanning tussen de 0-2V geproduceerd worden met dit shield, zonder dat er extern gevoed hoeft te worden. De multimeter geeft de juiste spanning aan tot aan 2 decimalen achter de komma nauwkeurig. De support van ABelektronics is ook als goed te bestempelen. Er wordt op de mails binnen 2 dagen gericht gereageerd.

Onder dit shield wordt er wel vermeld dat deze bedoeld is voor gebruik voor demonstratie en/of evaluatie doeleinden. Voor eindproduct kan deze shield nagebouwd worden met gewenste configuratie, omdat de DAC als ADC gebruik maakt van de Microchip Technologie. Deze micro onderdelen kunnen namelijk ingekocht worden. Dit zal niet een probleem opleveren.

8.1.3 Phidget Analog 4-Output.

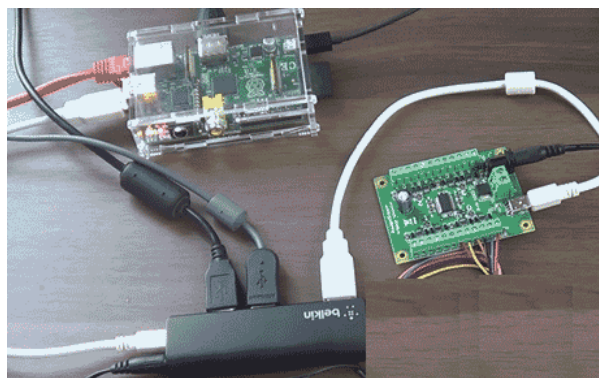
De Phidget Analog 4-Output is een bordje dat eigenlijk bedoeld is om spanning te produceren met 12-bit resolutie tussen de -10V en de 10V. Dit shield is uitgerust met de CY7C64215 full speed USB controller chip, AD5724 Quad-Channel Digital-to-Analog converter chip en een MinMax MWD1016 DC converter chip.

Dit bordje kan met allerlei andere hardware componenten gebruikt worden, zoals een PC, Raspberry Pi en Arduino. Het is niet specifiek gebouwd voor de Raspberry Pi. Via USB communicatie is het echter wel mogelijk om met de Raspberry Pi te kunnen werken. Zoals de naam al verradt, heeft deze shield 4 12-bit analoge uitgangen, wat betekent dat er 4 regelingen tegelijk gestuurd kunnen worden.



Figuur 33: Phidget Analog 4-Output bordje

Om de volledige output spanning te kunnen benutten, moet er wel een extra USB hub aangesloten worden, omdat anders de spanning van de Raspberry Pi 2 niet voldoende is om de Phidget aan te kunnen sturen. Het onderstaande plaatje laat zien hoe het prototype er uiteindelijk uit ziet met dit shield.



Figuur 34: Pi 2, USB HUB, Pi en Phidget Analog 4-Output samen

Op de website van Phidgets.com kunnen libraries en voorbeeld applicaties gedownload worden voor de Windows X86, X64 en Linux systemen. De libraries zijn als DLL bestanden te downloaden en voorbeeld applicaties kunnen in “bijna” alle talen van C tot C# gedownload worden. Alleen blijkt er dat ze geen “ARM-systeem” dll ondersteuning hebben voor Windows IoT systeem. Ze hebben wel voorbeelden beschikbaar in hun forums om hun Phidgets met de Raspberry Pi te laten communiceren.

Tijdens het testen is het gelukt op de Raspbian met het voorbeeld code die door Phidgets.com geleverd wordt, een spanning te kunnen genereren. De nauwkeurigheid was tot 2 decimalen achter de komma. Er is ook geprobeerd om de communicatie met het shield op het Windows IoT besturingssysteem op te zetten, alleen dit is niet gelukt. Phidsget.com heeft geen “ARM” libraries voor Windows IoT. Door de USB connectie is de datacommunicatie echter universeel en dus moet Windows IoT eerst dit shield herkennen, voordat er dataoverdracht kan plaatsvinden. Daarnaast vermeldt Phidgets.com support medewerker ook aan om Linux(Raspbian) te gebruiken, omdat het op Windows IoT niet gaat werken.

De support van Phidgets.com is als goed te bestempelen. Dit houdt in dat ze in hun forums/e-mails binnen ongeveer 1-2 dagen gericht op gestelde vragen reageren.

8.2 Conclusie Shields

Uit de analyse in hoofdstuk 3 en verdere analyse komt de onderstaande tabel 3 uit.

Tabel 3: Analyse resultaten 3 shields.

	4-20 mA T click	ADC-DAC Pi	Phidget Analog 4-Output
Heeft de leverancier zelf een bibliotheek/voorbeeld applicatie uitgegeven in een C/C# of Python programmeertalen kan met deze library de analoge uitgangen aangestuurd worden?	Voor zijn eigen software 'Mikroc pic' zijn bibliotheken uitgegeven. Deze bibliotheken waren onbruikbaar op de PI, waardoor er geen spanning gegenereerd konden worden	Abelektronics heeft bibliotheken voor Windows IoT in C# en Raspbian in Python programmeertaal. Daarnaast heeft de leverancier ook voorbeeld applicaties beschikbaar voor allebei de besturingssystemen. Er kon met de library een voorbeeld spanning verstuurd worden naar de uitgangen.	Phidgets heeft verschillende bibliotheken voor allerlei besturingssystemen in verschillende programmeertalen behalve voor IoT. Er kon op de Raspbian met behulp van C# library een voorbeeld spanning verstuurd worden naar de uitgangen.
Indien er geen bibliotheken beschikbaar zijn, kan er door standaard library die het desbetreffende besturingssysteem bezit, communicatie opgebouwd worden met het desbetreffende shield en de analoge uitgangen aangestuurd worden?	Tijdens het testen is het gebleken dat er op geen enkel manier gecommuniceerd kan worden met dit shield. Nadat het shield opnieuw is besteld, is er weer geprobeerd maar ook met de 2 ^e shield kon er geen communicatie opgebouwd worden.	Er kon met de standaard SPI library van Windows IoT als Raspbian een communicatie opgebouwd worden. Op de Raspbian moest er wel de Python 2/3 bibliotheken gebruikt worden. Deze kon in C# gebruikt worden door "dllimport" functie.	Er kon met de standaard SPI library in Raspbian een communicatie opgebouwd worden. Windows IoT lukte het niet om communicatie op te kunnen bouwen.
Is er een officiële ondersteuning voor de Windows IoT besturingssysteem?	Geen officiële ondersteuning	Er is een officiële ondersteuning aanwezig voor IoT	Er is geen officiële support aanwezig voor de IoT.
Is er een officiële ondersteuning voor de Raspbian besturingssysteem?	Geen officiële ondersteuning! Dit is wel opvallend, omdat ze zelf de tussenshield Pi Click Shield verkopen om het te laten werken met de Pi2.	Er is een officiële ondersteuning aanwezig voor Raspbian	Er is geen officiële support aanwezig voor de Raspbian. (Wel voor Linux).

Haalt het desbetreffende shield ook in praktijk de minimale 8-bit resolutie	Kon niet getest worden.	Er is gebleken uit het testen dat er een nauwkeurigheid van 2 cijfers achter de komma waargenomen kon worden.	Er is gebleken uit het testen dat er een nauwkeurigheid van 2 cijfers achter de komma waargenomen kon worden.
---	-------------------------	---	---

Groen gekleurd: voldoet volledig aan de eis.

Geel: voldoet gedeeltelijk

Rood: Voldoet niet aan de eisen.

Uit de bovenstaande tabel en de analyses kan er geconstateerd worden dat ADC-DAC Pi het enige shield is dat op beide besturingssystemen werkt en waarvoor ook voorbeeld applicaties beschikbaar zijn. Daarnaast kan er ook door middel van SPI standaard libraries met het shield gecommuniceerd worden.

Phidget Analog 4-Output shield werkt niet op de Windows IoT besturingssysteem wegens het ontbreken van “ARM” libraries voor de Windows IoT. De communicatie met deze shield gaat doormiddel van USB en is dus universeel te noemen. Hierdoor is het niet mogelijk om de standaard USB communicatie library te gebruiken. Daarnaast wordt het door de support afdeling aangeraden om bij Phidgets shields gebruik te maken van een Linux besturingssysteem. Windows IoT ontwikkeling is onbekend. Onder Raspbian werkt dit shield goed. Er kan een spanning geproduceerd worden tussen de -10 en 10 volt. Ook hun support is goed te noemen. Ze geven in 1-2 dagen antwoord en reageren gericht op de vragen.

4-20 mA T click van Micro Elektronika werkt op geen van beide besturingssystemen. De leverancier heeft ook geen libraries beschikbaar voor deze besturingssystemen. Tevens kan er door middel van de standaard SPI library geen communicatie opgebouwd worden met de 4-20 mA T click. Dit shield is wel opnieuw besteld om de hardware matig defect buiten te sluiten. Alleen is het weer gebleken dat het op geen enkele manier communicatie opgebouwd kan worden. Hierdoor is dit shield voor “nu” voor dit prototype onbruikbaar. Het kan zijn dat de leverancier na verloop van tijd een “library” beschikbaar stelt voor deze shield. Alleen tijdens de live support bleek dat ze toch kortaf zijn en de klanten misleiden. Dit komt het vertrouwen in deze leverancier niet ten goede. Tevens zijn deze voorbeeld applicaties ook allemaal gericht op hun eigen software Mikroc compiler software. Deze software is natuurlijk niet gratis te gebruiken en er moet er licentie kosten betaald worden.

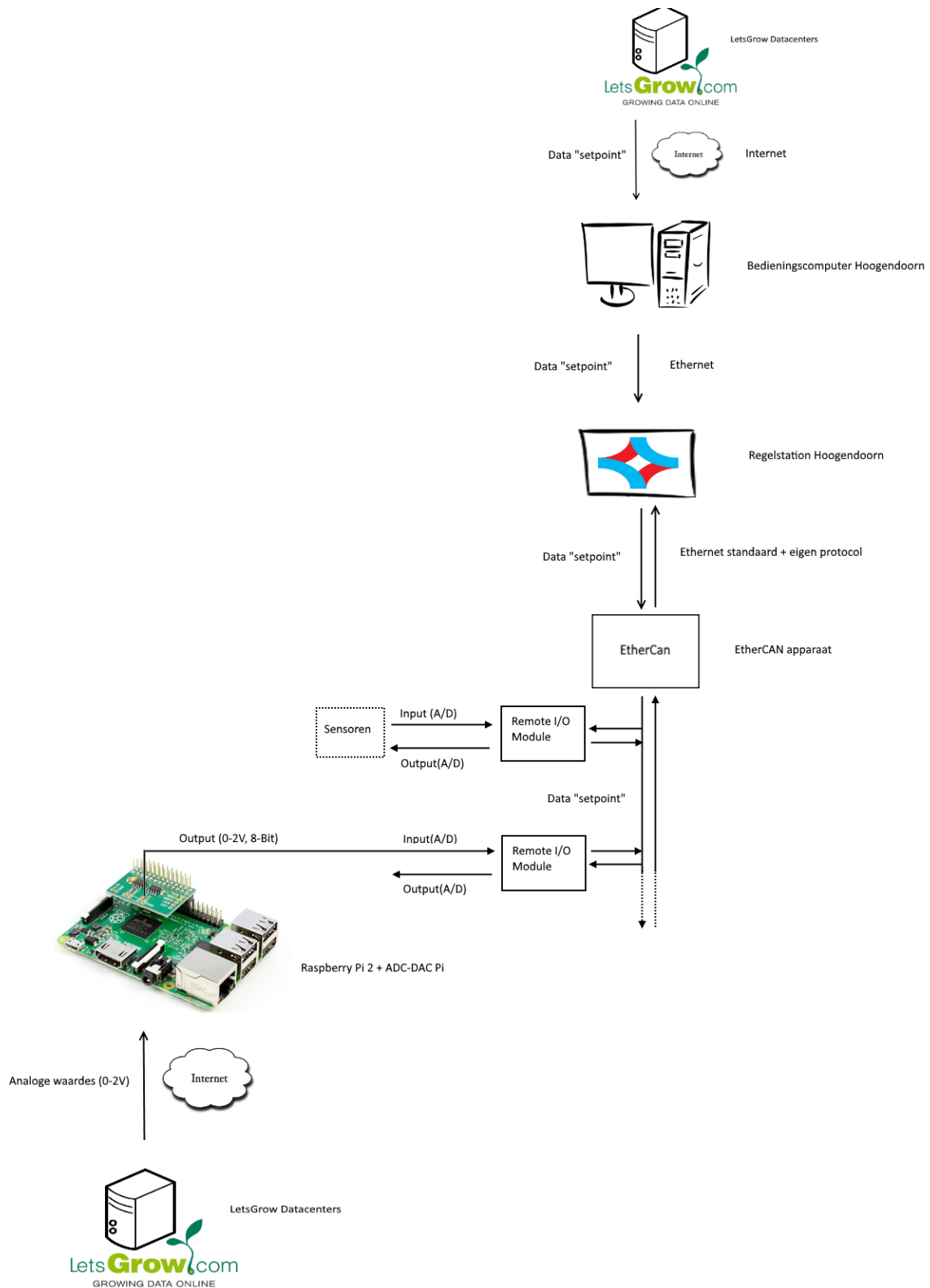
9. De keuze voor het shield.

Vanwege de voorkeur voor het Windows IoT besturingssysteem is de keuze gevallen op de ADC-DAC Pi van Abelektronics. Dit shield heeft 2 analoge in- en 2 analoge uitgangen. Tevens heeft de leverancier voor Windows IoT officieel libraries beschikbaar en worden deze om 5-6 maanden geüpdatet. Ook zijn er voorbeeld applicaties te vinden die op de Raspberry Pi 2 goed draaien. Ze hebben ook een officiële ondersteuning voor de Windows IoT en Raspbian besturingssystemen. Daarnaast is de support van Abelektronics ook goed te noemen. Ze geven gerichte antwoorden op de gestelde vragen en misleiden de klant niet. De resolutie is 12-bit en dit houdt in dat het nauwkeurig genoeg is. Daarnaast kan er door middel met dit shield een spanning geproduceerd te worden, dankzij de referentie spanning van de Pi 2. Hierdoor heeft deze shield geen externe voeding nodig.

Belangrijkste is dat het door middel van dit shield de behoefte van de opdrachtgever gerealiseerd kan worden. Dankzij dit shield kan er een analoge waarde tussen de 0-2V geproduceerd worden die minimaal 8-bit resolutie heeft om de regelingen te kunnen beïnvloeden.

10. Architectuur nieuwe situatie

De nieuwe situatie gaat er als volgt uitzien:



Figuur 35: Architectuur nieuwe opstelling

Uitleg nieuwe situatie:

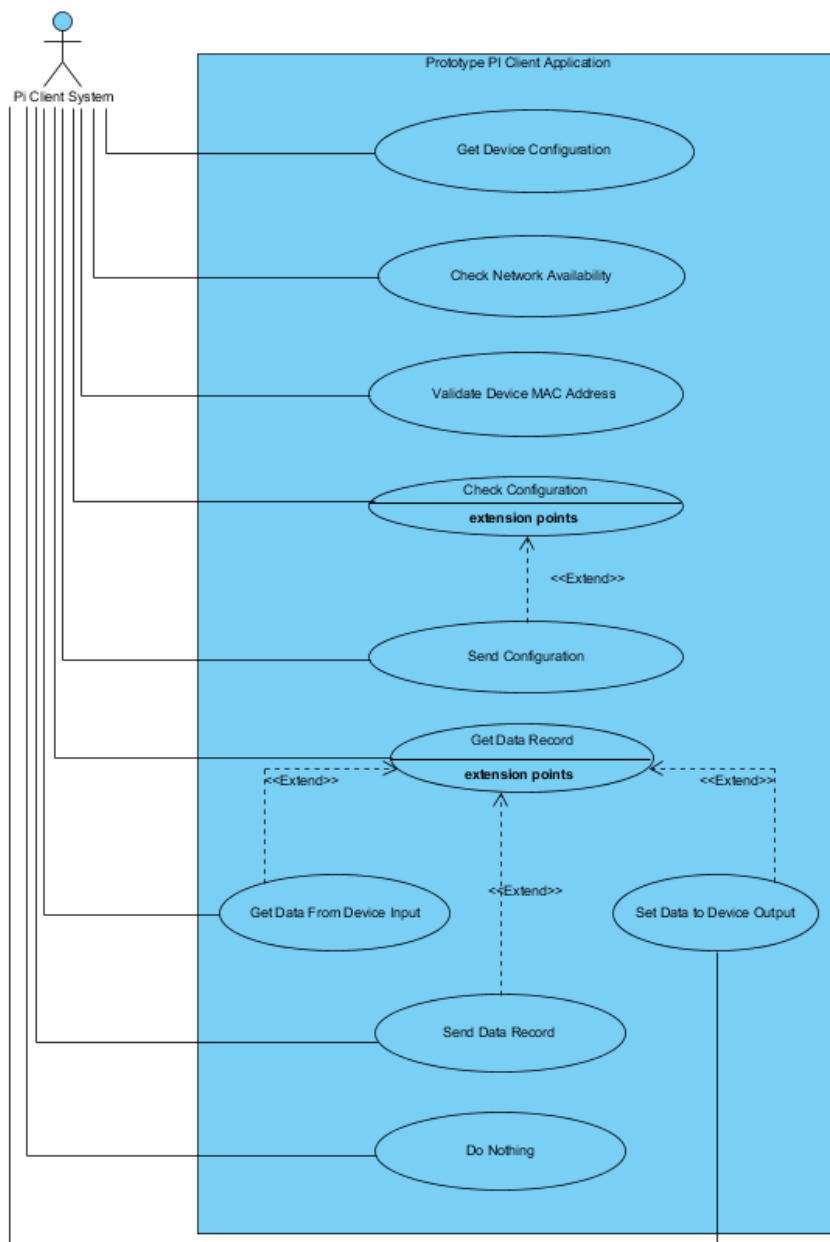
De Raspberry Pi 2 met de ADC-DAC Pi gaat door middel van bedrading hangen aan de 'remote input/output modules'. Om de kabels stevig vast te kunnen zetten in de 'remote input/output modules' wordt er gebruik gemaakt van 75mm² draaddoorsnede. Deze bedrading wordt aan de "plus-kant" gekoppeld en de andere aan de "grond-kant". Nadat de bedrading gekoppeld zijn, kan er een "ethernet-kabel" aangehangen worden aan de Raspberry Pi 2. Uiteindelijk wordt er een voeding gekoppeld en dan start het prototype uit zich zelf op.

11. UML Diagrammen

In dit hoofdstuk worden verschillende Unified Modeling Language (UML) diagrammen behandeld die betrekking hebben op de te ontwikkelen software.

11.1 Use-case diagrammen en beschrijvingen.

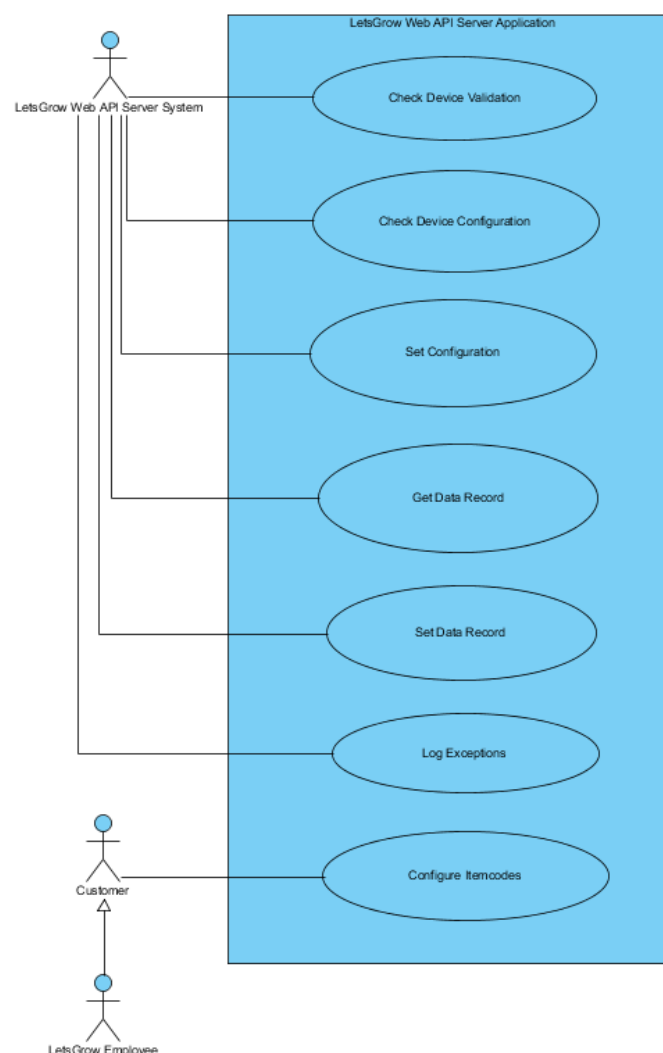
Voor het prototype moet er een applicatie opgebouwd worden en de LetsGrow Web API server applicatie uitgebreid moeten worden, omdat er sprake is van een client-server werking. Dit houdt in dat er een cliënt applicatie opgebouwd moet worden die connectie moet maken met de LetsGrow.com Web API server. Daarnaast moet de server uitgebreid worden om het binnenkomende verkeer van het prototype cliënt te kunnen ontvangen en te kunnen bewerken. De scenario's zijn afgeleid uit de functionele eisen van de opdrachtgever en de bedrijfsmentoren.



Figuur 36: Use-Case diagram Client Applicatie

Het Pi cliënt systeem moet eerst de device configuratie samenstellen. Nadat hij de device configuratie samengesteld heeft, kijkt hij of er internet aanwezig is. Indien er internet aanwezig is, stuurt hij het MAC adres van het Raspberry Pi op naar de LetsGrow webserver om te valideren.

Als het validatie proces goed is gegaan, gaat het Pi cliënt systeem een aanvraag doen bij de webserver om te weten komen of er een configuratie aanwezig is van het desbetreffende device. Indien dit het niet geval is, verstuurt het cliënt systeem zijn configuratie naar de webserver. Daarna vraagt het Pi cliënt systeem een lijst van data record op. In deze lijst kan het Pi cliënt systeem zien wat hij als actie moet ondernemen. Indien hij gegevens moet opsturen naar de webserver, dan stuurt hij een nieuwe lijst van datarecord op naar de webserver. Deze handeling is niet noodzakelijk, maar is mogelijk van toepassing in toekomstige uitbreidingen van de applicatie. Indien hij niets moet doen, dan doet hij ook niets het komende paar minuten.



Figuur 37: Use Case diagram Server applicatie

Het Webserver systeem kijkt na of het MAC adres dat verkregen is van het Pi cliënt systeem bekend is bij LetsGrow.com. Indien dit niet het geval is, logt de webserver het onbekende MAC adres. Daarna verzendt hij het resultaat terug naar het Pi cliënt systeem. Daarna kijkt

hij of er device configuratie informatie aanwezig is bij LetsGrow.com. Indien dit niet het geval is, dan verstuurt hij een bericht terug naar het Pi cliënt systeem. Het Pi cliënt systeem verstuurt dan zijn configuratie informatie en als het webserver systeem dat ontvangen heeft, zet hij deze informatie in de LetsGrow database. Nadat hij dit proces verlopen is, verzamelt hij alle data die te vinden is voor het desbetreffend device MAC adres en bouwt een lijst van data record aan. Deze lijst van data record verstuurt hij weer terug naar Pi cliënt systeem. Als een nieuwe lijst van het type “data record” wordt ontvangen van het Pi cliënt systeem, dan zet hij dit nieuwe data record in de database.

Om het data record door de Webserver goed te laten samenstellen, moet een klant en/of een LetsGrow medewerker de moduleconfiguratie voor het prototype via de LetsGrow.com website uitvoeren. Deze configuratie gebeurt op dezelfde manier als alle andere configuraties. Alleen moet hier opgelet worden op de itemcode benamingen. Hiervoor moet het klant/LetsGrow personeel houden aan bepaalde benaming prefix.

De itemcodes mogen maximaal 8 karakters hebben. Zo moet een analoge uitgang itemcode eruitzien: “IOOUTA01”. Het prefix “IO” is voor INPUT/OUTPUT, “OUT” voor OUTPUT, “A” voor Analog(Analoog) en “01” is de channel (uitgang kanaal). De analoge ingang itemcode ziet er zo uit: “IOINA01”, waarbij “IN” voor INPUT staat.

De eisen waarop het use-case diagram is gebaseerd betreffen de volgende functionele eisen:

Prototype Pi Client applicatie:

Get Device Configuration: 1, 7
Check Network Availability: 3
Validate Device MAC address: 4
Check Configuration: 7
Send Configuration: 7
Get Data Record: 5, 6
Set Data To Device Output: 8
Do Nothing: 3

“Get Data From Device Input” en “Send Data Record” use-cases zijn gebaseerd op de uitbreidbaarheid in de toekomst. Deze komen niet voort uit de eisen, maar zijn door de afstudeerder geïmplementeerd voor toekomstig gebruik.

LetsGrow Web API Server applicatie:

Check Device Validation: 4
Check Device Configuration: 7
Put Configuration: 7
Collect Data Record: 6
Send Data Record: 6
Log Exceptions: 10
Configure Itemcodes: 12

“Put Data Record” use-case is gebaseerd op uitbreidbaarheid. Deze komt niet voort uit de eisen, maar zijn door de afstudeerder geïmplementeerd voor toekomstig gebruik.

11.2 Use-case beschrijvingen

Prototype Pi cliënt applicatie.

Get Device Configuration

Naam	Get Device Configuration
Samenvatting	Het Pi cliënt systeem haalt zijn device configuratie informatie op en stelt deze samen.
Actoren	Het Pi cliënt systeem.
Aannamen	De service monteur moet het prototype goed hebben aangesloten. Het prototype is opgestart.
Beschrijving	(1) Pi cliënt systeem haalt zijn device configuratie informatie op. Indien dit niet lukt, dan treedt er <u>[Configuratie ophalen mislukt]</u> uitzondering op.
Uitzondering	<u>[Configuratie ophalen mislukt]</u> (1) Pi cliënt systeem gaat na 5 minuten weer proberen om de configuratie op te halen.
Resultaat	Het Pi cliënt systeem heeft zich succesvol configuratie informatie opgehaald van de device.

Check Network Availability

Naam	Check Network Availability
Samenvatting	Het Pi cliënt systeem kijkt of er een actieve netwerkverbinding is.
Actoren	Het Pi cliënt systeem.
Aannamen	De service monteur moet het prototype goed hebben aangesloten. Het prototype is opgestart.
Beschrijving	(1) Pi cliënt systeem kijkt of er een actieve internetverbinding aanwezig is. Indien dit niet het geval is, dan treedt er <u>[Geen netwerkverbinding]</u> uitzondering op.
Uitzondering	<u>[Geen netwerkverbinding]</u> (1) Pi cliënt systeem laat groene en rode leds tonen. (2) Pi cliënt systeem gaat na 30 seconden weer proberen om te kijken of er wel een internetverbinding is.
Resultaat	Het Pi cliënt systeem heeft een actieve internetverbinding.

Validate Device MAC address

Naam	Validate Device MAC address
Samenvatting	Het Pi cliënt systeem verifieert aan de hand zijn MAC adres of hij zijn verdere werking kan starten.
Actoren	Het Pi cliënt systeem.
Aannamen	De service monteur moet het prototype goed hebben aangesloten. Het prototype moet een actieve internetverbinding hebben. De service monteur moet de MAC adres al weten van het prototype. De MAC adres moet bekend zijn bij de LetsGrow.com servers ¹ . De Web API server moet online zijn.
Beschrijving	(1) Pi cliënt systeem stuurt zijn MAC adres voor verificatie naar de Web API server. (2) Web API server ontvangt deze MAC adres. (3) Web API server stuurt een antwoord bericht terug. (4) Pi cliënt systeem ontvangt het antwoord bericht. Indien het bericht een negatief antwoord bevat, dan treedt de <u>[Verificatie mislukt]</u>

	uitzondering op. (5) Pi cliënt systeem is klaar om de check configuratie uit te kunnen voeren.
Uitzondering	<u>[Verificatie mislukt]</u> (1) Pi cliënt systeem voert 59 minuten niets uit. (2) Pi cliënt systeem stuurt zijn MAC adres na 60 minuten weer op om te verifiëren. (3) Web API stuurt een antwoord bericht terug.
Resultaat	Het Pi cliënt systeem heeft zich succesvol geverifieerd bij de Web API server en kan verder gaan met zijn verdere werking.

1. Het MAC adres wordt eerst door LetsGrow medewerkers opgevraagd uit de Pi 2 en in de database geregistreerd. Daarna wordt dit prototype overhandigd aan de service monteur. Opvragen van de MAC adres gebeurt door het prototype aan te sluiten op een monitor. Daarna kan het MAC adres uitgelezen worden.

Check Configuration

Naam	Check Configuration
Samenvatting	Het Pi cliënt systeem gaat zijn configuratie nakijken of het bekend is bij de Web API server om verder te gaan met zijn verdere werking.
Actoren	Het Pi cliënt systeem.
Aannamen	Het prototype moet een actieve internetverbinding hebben. Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server. De Web API server moet online zijn.
Beschrijving	(1) Pi Cliënt systeem stuurt een bericht om te kijken of zijn configuratie bekend is bij de Web API server (2) Web API server ontvangt dit bericht. (3) Web API server stuurt een antwoord bericht terug. (4) Pi cliënt systeem ontvangt het antwoord bericht. Indien de configuratie niet aanwezig is, dan treedt de <u>[Geen configuratie]</u> uitzondering op. (5) Pi Cliënt systeem is klaar om verder te gaan met het ophalen van een lijst van data record.
Uitzondering	<u>[Geen configuratie]</u> (1) Pi Cliënt systeem voert de "Send Configuration" use-case uit.
Resultaat	Het Pi Cliënt systeem heeft succesvol zijn configuratie check uitgevoerd en kan verder gaan met zijn verdere werking.

Send Configuration

Naam	Send Configuration
Samenvatting	Het Pi cliënt systeem gaat zijn configuratie opsturen naar de Web API server.
Actoren	Het Pi cliënt systeem.
Aannamen	Het prototype moet een actieve internetverbinding hebben. Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server. Het Pi cliënt systeem heeft zijn configuratie check uitgevoerd. De Web API server moet online zijn.
Beschrijving	(1) Pi cliënt systeem haalt de device configuratie informatie op van het prototype. Indien er geen configuratie opgehaald kan worden, dan treedt de <u>[Ophalen mislukt]</u> uitzondering op. (2) Pi cliënt systeem stuurt de opgehaalde configuratie op naar de Web API server.

	<p>(3) Web API server ontvangt deze configuratie informatie.</p> <p>(4) Web API server stuurt een antwoord bericht terug.</p> <p>(5) Pi cliënt systeem ontvangt het antwoord bericht van de Webserver. Indien het bericht een negatief antwoord bevat, dan treedt de <u>[Configuratie Mislukt]</u> uitzondering op.</p> <p>(6) Pi cliënt systeem is klaar om verder te gaan met het ophalen van de datarecord.</p>
Uitzondering	<p><u>[Ophalen mislukt]</u></p> <p>(1) Pi cliënt systeem gaat na 5 minuten weer de configuratie informatie ophalen van de device.</p> <p><u>[Configuratie mislukt]</u></p> <p>(1) Pi cliënt systeem gaat na 5 minuten weer de configuratie informatie versturen.</p>
Resultaat	Het Pi cliënt systeem heeft zijn configuratie succesvol opgestuurd naar de Web API server en kan verder gaan met zijn verdere werking.

Get Data Record

Naam	Get Data Record
Samenvatting	Het Pi cliënt systeem gaat een datarecord ophalen van de WebServer.
Actoren	Het Pi cliënt systeem.
Aannamen	<p>Het prototype moet een actieve internetverbinding hebben.</p> <p>Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server.</p> <p>Het Pi cliënt systeem heeft zijn configuratie check uitgevoerd.</p> <p>De Web API server moet online zijn.</p>
Beschrijving	<p>(1) Pi cliënt systeem vraagt aan de Webserver of hij een lijst van datarecord kan krijgen.</p> <p>(2) Web API server ontvangt dit aanvraag bericht.</p> <p>(3) Web API server stelt een lijst van datarecord op.</p> <p>(4) Web API server stuurt deze lijst van datarecord naar het Pi cliënt systeem.</p> <p>(5) Pi cliënt systeem ontvangt het antwoord van de Webserver</p> <p>(6) Pi cliënt systeem gaat de ontvangen lijst van data record processen en kijkt wat hij als actie moet ondernemen. Indien de actie wegschrijven is, dan treedt de [Data wegschrijven naar device uitgangen] uitzondering op. Indien de actie ophalen is, dan treedt de [Data ophalen van de device ingangen] uitzondering op. Indien de actie niets doen is, dan treedt de uitzondering [Doe niets] op.</p> <p>(7) Pi cliënt systeem is klaar om verder te gaan met de volgende werking.</p>
Uitzondering	<p><u>[Data wegschrijven naar device uitgangen]</u></p> <p>(1) Pi cliënt systeem voert de “Set Data To Device Output” use-case beschrijving uit.</p> <p><u>[Data ophalen van de device ingangen]</u></p> <p>(1) Pi cliënt systeem voert de “Get Data From Device Input” use-case beschrijving uit.</p> <p><u>[Doe niets]</u></p> <p>(1) Pi cliënt systeem voert de “Do Nothing” use-case beschrijving uit.</p>
Resultaat	Het Pi cliënt systeem heeft succesvol een lijst van datarecord ontvangen van de Web API server en gaat deze procedure na 5 minuten weer herhalen.

Get Data From Device Input

Naam	Get Data From Device Input
Samenvatting	Het Pi cliënt systeem gaat de waardes van de device ingangen uitlezen.
Actoren	Het Pi cliënt systeem.
Aannamen	Het prototype moet een actieve internetverbinding hebben. Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server. Het Pi cliënt systeem heeft zijn configuratie check uitgevoerd. De Web API server moet online zijn. Het Pi cliënt systeem heeft succesvol een lijst van data record ontvangen.
Beschrijving	<ol style="list-style-type: none"> (1) Pi cliënt systeem haalt van de lijst van de datarecord de lege waardes die ingevuld moeten worden en bijbehorende uitgangkanalen. (2) Pi cliënt systeem leest de waardes op van de device uitgangen. (3) Pi cliënt systeem laat een groen led zien voor het succesvol ophalen van de waardes. (4) Pi Client systeem voegt deze nieuwe waardes in een nieuwe lijst van datarecord. Indien niet succesvol opgehaald kan worden, dan treedt de [Uitlezen mislukt] uitzondering op.
Uitzondering	[Uitlezen mislukt.] <ol style="list-style-type: none"> (1) Pi cliënt systeem laat een rode ledje tonen. (2) Pi cliënt systeem vult in de lijst van datarecord de waarde "0" in. (3) Pi cliënt systeem gaat na 5 minuten weer proberen om waardes uit te lezen van de shield uitgangen.
Resultaat	Het Pi cliënt systeem heeft succesvol de ingang waardes van de shield uitgelezen.

Set Data To Device Output

Naam	Set Data To Device Output
Samenvatting	Het Pi cliënt systeem gaat de datarecord waardes wegschrijven naar de uitgangen
Actoren	Het Pi cliënt systeem
Aannamen	Het prototype moet een actieve internetverbinding hebben. Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server. Het Pi cliënt systeem heeft zijn configuratie check uitgevoerd. De Web API server moet online zijn. Het Pi cliënt systeem heeft succesvol een lijst van data record ontvangen.
Beschrijving	<ol style="list-style-type: none"> (1) Pi cliënt systeem haalt van de lijst van de datarecord de waardes en bijbehorende uitgangkanalen. (2) Pi cliënt systeem stuurt de uitgehaalde waardes naar de desbetreffende kanalen. (3) Pi cliënt systeem laat een groen ledje tonen voor het succesvol sturen van de waardes naar de uitgangen. Indien niet succesvol weggeschreven kan worden, dan treedt de [Wegschrijven mislukt] uitzondering op.
Uitzondering	[Wegschrijven mislukt.] <ol style="list-style-type: none"> (1) Pi cliënt systeem laat een rode ledje tonen. (2) Client systeem gaat na 5 minuten weer proberen om weg te schrijven naar de uitgangen.
Resultaat	De cliënt systeem heeft de ontvangen lijst van data record waardes succesvol

	weggeschreven naar de device uitgangen.
--	---

Send Data Record

Naam	Set Data Record
Samenvatting	Het Pi cliënt systeem gaat de vernieuwde datarecord lijst versturen naar de Web API server.
Actoren	Het Pi cliënt systeem.
Aannamen	Het prototype moet een actieve internetverbinding hebben. Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server. Het Pi cliënt systeem heeft zijn configuratie check uitgevoerd. De Web API server moet online zijn. Het Pi cliënt systeem heeft succesvol een lijst van data record ontvangen. Het Pi cliënt systeem heeft succesvol de ingang waardes van de shield uitgelezen.
Beschrijving	<ol style="list-style-type: none"> 1. Pi cliënt systeem stelt aan de hand van de uitgelezen waardes een nieuwe lijst van datarecord op. 2. Pi cliënt systeem stuurt deze nieuwe lijst van datarecord op naar de Web API server. Indien niet succesvol verstuurd kan worden, dan treedt de [Versturen mislukt] uitzondering op. 3. Web API server ontvangt deze lijst van datarecord. 4. Web API server stuurt een antwoord terug. 5. Pi cliënt systeem ontvangt het antwoord. 6. Pi cliënt systeem is voor nu klaar met zijn werking. Na 5 minuten gaat de Client systeem de "Get Data Record" use-case beschrijving opnieuw uitvoeren.
Uitzondering	[Verzenden mislukt] (1) Pi cliënt systeem gaat na 5 minuten weer proberen om de data te verzenden.
Resultaat	Het Pi cliënt systeem heeft succesvol een nieuwe lijst van datarecord verzonden naar de Web API server

Do Nothing

Naam	Do Nothing
Samenvatting	Het Pi cliënt systeem gaat de komende bepaalde tijd in "Doe Niets" stand
Actoren	Het Pi cliënt systeem.
Aannamen	Het prototype is opgestart.
Beschrijving	<ol style="list-style-type: none"> 1. Pi cliënt systeem gaat de "Doe niets" in. 2. Pi cliënt systeem laat de groene en rode ledje branden.
Resultaat	Het Pi cliënt systeem is succesvol de "Doe Niets" stand ingegaan.

LetsGrow Web API server applicatie.

Check Device Validation

Naam	Check device validation
Samenvatting	De Web API server verifieert de verkregen MAC adres of de device gemachtigd is om gegevens te kunnen uitwisselen.
Actoren	De Web API server.
Aannamen	De service monteur moet het prototype goed hebben aangesloten. Het prototype moet een actieve internetverbinding hebben. De service monteur moet de MAC adres al weten van het prototype. De Web API server moet een MAC adres ontvangen hebben van het Pi cliënt systeem. De Web API server moet online zijn.
Beschrijving	<ol style="list-style-type: none">(1) Pi cliënt systeem stuurt zijn MAC adres voor verificatie naar de Web API server.(2) Web API ontvangt een MAC adres van het Pi cliënt systeem.(3) Web API server kijkt na of de MAC adres in de SQL database voorkomt. Indien dit niet het geval is, dan treedt de <u>[Verificatie mislukt]</u> uitzondering op.(4) Web API server stuurt een positief antwoord bericht terug naar het Pi cliënt systeem.(5) Pi cliënt systeem ontvang een antwoord bericht
Uitzondering	<u>[Verificatie mislukt]</u> <ol style="list-style-type: none">(1) Web API Server logt deze “onbekende” MAC adres naar de “logging” database.(2) Web API Server stuurt een negatief antwoord bericht terug naar het Pi cliënt systeem.
Resultaat	De Web API server heeft succesvol geverifieerd dat de device MAC adres bemachtigd is om gegevens te kunnen uitwisselen.

Check Device Configuration

Naam	Check device configuration
Samenvatting	De Web API server kijkt of de desbetreffende device configuratie al aanwezig is in de SQL database.
Actoren	De Web API server.
Aannamen	De service monteur moet het prototype goed hebben aangesloten. Het prototype moet een actieve internetverbinding hebben. Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server. De Web API server moet online zijn.
Beschrijving	<ol style="list-style-type: none">(1) Pi cliënt systeem stuurt een apparaat configuratie check aanvraag bericht.(2) Web API ontvangt een configuratie aanvraag bericht van het Pi cliënt systeem.(3) Web API server kijkt na of de device configuratie in de SQL database voorkomt. Indien dit niet het geval is, dan treedt de <u>[Geen configuratie aanwezig]</u> uitzondering op.(4) Web API server stuurt een positief antwoord bericht terug naar het Pi cliënt systeem.(5) Pi cliënt systeem ontvang een antwoord bericht

Uitzondering	[Geen configuratie aanwezig] (1) Web API server stuurt een negatief antwoord bericht terug naar het Pi cliënt systeem.
Resultaat	De Web API server heeft succesvol gekeken of er een device configuratie aanwezig is in de SQL database.

Put Configuration

Naam	Put configuration
Samenvatting	De Web API server gaat de desbetreffende device configuratie in de SQL database zetten.
Actoren	De Web API server.
Aannamen	De service monteur moet het prototype goed hebben aangesloten. Het prototype moet een actieve internetverbinding hebben. Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server. De Web API server heeft geen configuratie informatie van de desbetreffende device. De Web API server moet online zijn.
Beschrijving	(1) Pi cliënt systeem stuurt zijn configuratie informatie op naar de Web API server. (2) Web API ontvangt een configuratie informatie pakket van het Pi cliënt systeem. (3) Web API server kijkt eerst na of de configuratie informatie niet leeg is en zet de ontvangen configuratie informatie in de SQL database. Indien dit niet het geval is, dan treedt de [Opslaan mislukt] uitzondering op. (4) Web API server stuurt een positief antwoord bericht terug. (5) Pi cliënt systeem ontvang een antwoord bericht.
Uitzondering	[Opslaan mislukt] (1) Web API server logt deze foutmelding. (2) Web API server stuurt een negatief antwoord bericht terug.
Resultaat	De Web API server heeft succesvol de device configuratie in de SQL database gezet.

Collect Data Record

Naam	Collect Data Record
Samenvatting	De Web API server gaat een lijst van data record verzamelen aan de hand van de gegevens die in de SQL database voorkomen.
Actoren	De Web API server.
Aannamen	De service monteur moet het prototype goed hebben aangesloten. Het prototype moet een actieve internetverbinding hebben. Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server. De Web API server heeft configuratie informatie van de desbetreffende device. De Klant/LetsGrow.com medewerker moet de itemcodes goed ingesteld hebben. ² De Web API server moet online zijn.
Beschrijving	(1) Pi cliënt systeem stuurt data verzoek naar de Web API server. (2) Web API server ontvangt een verzoek bericht van het Pi cliënt

	<p>systeem.</p> <p>(3) Web API server verzamelt allerlei data gegevens uit de SQL database. Indien de server geen data kan verkrijgen uit de SQL database, dan treedt de <u>[Verzamelen mislukt]</u> uitzondering op.</p>
Uitzondering	<p><u>[Verzamelen mislukt]</u></p> <p>(1) Web API server logt deze foutmelding.</p> <p>(2) Web API server maakt een lege lijst van data record aan.</p>
Resultaat	De Web API server heeft succesvol een lijst van data record verzameld aan de hand van de gegevens die in de SQL database voorkomen.

2. Deze aanname wordt via de LetsGrow.com website verricht. Hier heeft de projectleider niet de bevoegdheid om de gehele bovenstaande aanname procedure te beïnvloeden. Deze procedure wordt voor elke willekeurige itemcode configuratie doorlopen. Op bladzijde 50 is beschreven hoe de itemcodes ingevoerd moeten worden. Aan deze richtlijnen moet strikt gehouden worden voor een goede werking van deze use-case.

Send Data Record

Naam	Send Data Record
Samenvatting	De Web API server gaat een lijst van data record verzenden naar het Pi cliënt systeem
Actoren	De Web API server.
Aannamen	<p>De service monteur moet het prototype goed hebben aangesloten.</p> <p>Het prototype moet een actieve internetverbinding hebben.</p> <p>Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server.</p> <p>De Web API server heeft configuratie informatie van de desbetreffende device.</p> <p>De Klant/LetsGrow.com medewerker moet de itemcodes goed ingesteld hebben.</p> <p>De Web API server heeft een lijst van data record verzameld.</p> <p>De Web API server moet online zijn.</p>
Beschrijving	<p>(1) Web API server verstuurt de verzamelde lijst van data record naar het Pi cliënt systeem. Indien dit niet lukt, dan treed er <u>[Verzenden mislukt]</u> uitzondering op.</p> <p>(2) Pi cliënt systeem ontvangt deze verstuurde lijst van data records.</p>
Uitzondering	<p><u>[Verzenden mislukt]</u></p> <p>(1) Web API server logt deze foutmelding.</p> <p>(2) Web API server stuurt een negatief antwoord terug naar het Pi cliënt systeem.</p>
Resultaat	De Web API server heeft succesvol een lijst van data record verzonden het Pi cliënt systeem.

Put Data Record

Naam	Put Data Record
Samenvatting	De Web API server gaat de nieuwe ontvangen lijst van data record in de SQL database toevoegen.
Actoren	De Web API server.
Aannamen	<p>De service monteur moet het prototype goed hebben aangesloten.</p> <p>Het prototype moet een actieve internetverbinding hebben.</p> <p>Het Pi cliënt systeem moet succesvol geverifieerd zijn door de Web API server.</p> <p>De Web API server heeft configuratie informatie van de desbetreffende device.</p> <p>De Klant/LetsGrow.com medewerker moet de itemcodes goed ingesteld</p>

	<p>hebben.</p> <p>De Web API server heeft een lijst van data record ontvangen.</p> <p>De Web API server moet online zijn.</p>
Beschrijving	<p>(1) Pi cliënt systeem stuurt een lijst van datarecord op.</p> <p>(2) Web API server ontvangt een lijst van datarecord van het Pi cliënt systeem.</p> <p>(3) Web API server zet het ontvangen lijst van datarecord in de SQL database. Indien de server geen data kan opslaan in de SQL database, dan treedt de <u>[Opslaan mislukt]</u> uitzondering op.</p> <p>(4) Web API Server stuurt een positief antwoord bericht terug naar het Pi cliënt systeem.</p> <p>(5) Pi cliënt systeem ontvangt een antwoord bericht.</p>
Uitzondering	<p><u>[Opslaan mislukt]</u></p> <p>(1) Web API server logt deze foutmelding.</p> <p>(2) Web API Server stuurt een negatief antwoord bericht terug naar het Pi cliënt systeem.</p>
Resultaat	De Web API server heeft een lijst van data record succesvol toegevoegd in de SQL database.

Log Exceptions

Naam	Log Exceptions
Samenvatting	De Web API server gaat foutmeldingen loggen in de logging database
Actoren	De Web API server.
Aannamen	De Web API server moet online zijn.
Beschrijving	(1) Web API server zet de opgetreden foutmeldingen in de database.
Uitzondering	Geen
Resultaat	De Web API server heeft de foutmeldingen succesvol gelogd in de logging database.

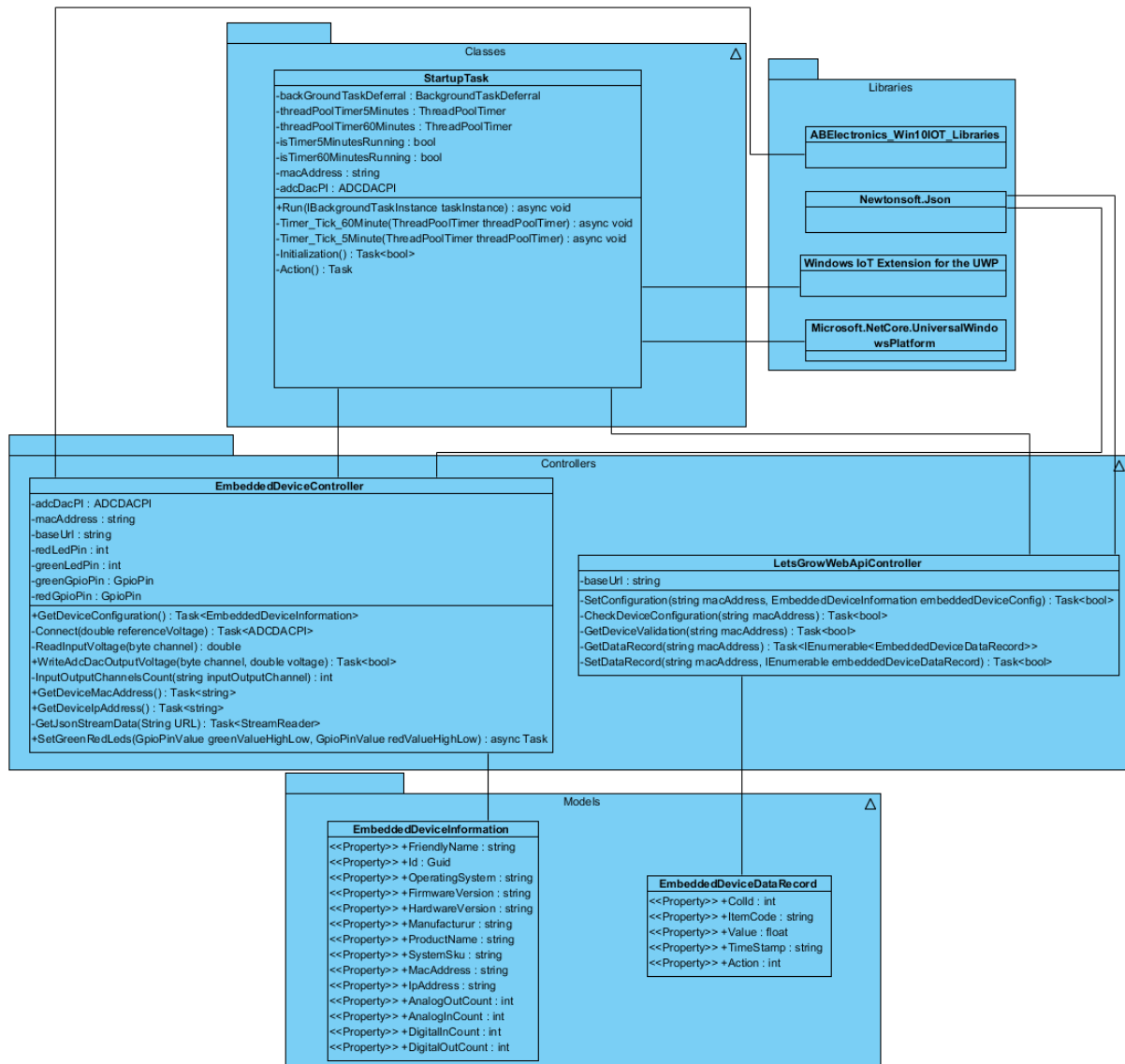
Configure Itemcodes

Naam	Configure itemcodes
Samenvatting	De klant en/of Letsgrow medewerker gaat de itemcodes invoeren.
Actoren	De klant, LetsGrow medewerker
Aannamen	<p>LetsGrow.com moet online zijn.</p> <p>De klant/letsgrow medewerker moet ingelogd zijn.</p> <p>Moduleconfiguratie pagina moet open staan.</p>
Beschrijving	<p>(1) Klant/medewerker voert de itemcode volgens de regels in.</p> <p>(2) Klant/medewerker drukt op de knop "save"</p> <p>(3) Systeem heeft de itemcode in de SQL database ingevoerd.</p>
Uitzondering	Geen
Resultaat	De klant en/of Letsgrow medewerker heeft de itemcodes succesvol ingevoerd.

11.3 Klassendiagrammen

De klassendiagrammen zijn hieronder afgebeeld. Er is gekozen voor de architectuur van de cliënt applicatie te realiseren in meerdere lagen, zodat de functionaliteiten duidelijk gescheiden worden. Hierdoor heeft elke laag zijn “eigen” verantwoordelijkheid. Daarnaast kunnen de lagen in de toekomst vervangen of meerdere lagen geïmplementeerd worden zonder dat er een groot stuk van de software gewijzigd hoeft te worden.

Client klassendiagram:



Figuur 38: Client klassendiagram

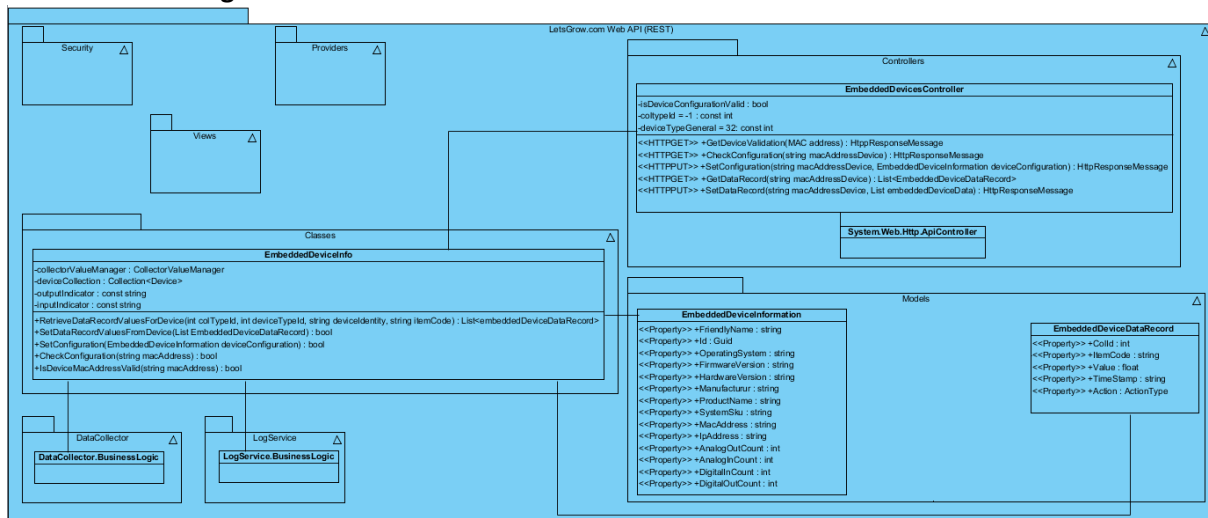
De cliënt applicatie bestaat uit de volgende lagen: Een Classes laag voor de “business logic” van de cliënt applicatie en voor de “main” werking van de cliënt applicatie.

De Controllers laag zorgt ervoor dat de http “verzoeken” (REST communicatie) behandeld worden door de LetsGrowWebApiController en de “device” handelingen(connecteren, waardes opvragen/wegschrijven) die benodigd zijn voor het prototype door de

EmbeddedDeviceController. De Models laag bevat allerlei modellen die benodigd zijn om de configuratiefile samen te stellen en de datarecords samen te stellen.

Het server gedeelte is een uitbreiding van de bestaande LetsGrow Web API. Deze Web API is gebaseerd op ASP.NET Web API. Dit houdt in dat er verschillende objecten zijn. Models map bevat verschillende modellen. Deze houden de data in de applicatie in. Controllers mapje zorgt ervoor dat de HTTP “verzoeken” behandeld worden door de verschillende controllers. De Classes mapje bezit de “Business logic”.

Server Klassendiagram:



Figuur 39: Server Klassendiagram

11.4 Klassen beschrijvingen.

Client klassen:

StartupTask:

De “StartupTask” zorgt voor de “business logic” van het gehele prototype applicatie systeem. Hierin wordt eerst een “BackgroundTask” aangemaakt en ook gestart. Daarna wordt deze “task” altijd in een actieve stand gehouden, zodat het prototype voor altijd in de achtergrond kan blijven werken.

De methodes die in deze klasse voorkomen, zorgen ervoor dat er “Timer” aangemaakt kunnen worden van 5 en 60 minuten, het gehele prototype proces opgestart kan worden en netwerkverbinding check uitgevoerd kan worden. Tevens wordt de verkregen lijst van data record set gesorteerd om te kijken welke actie er genomen kan worden, zodat het prototype correct zijn werking kan doen.

EmbeddedDeviceController:

Deze klasse zorgt ervoor dat er een communicatie opgebouwd kan worden met de desbetreffende shield. De methodes in deze klasse zorgen ervoor dat er device configuratie opgehaald kan worden, geconnecteerd kan worden met de shield, Ingang data weggeschreven kunnen worden, Uitgang data opgehaald kunnen worden, MAC-adres en IP-adres opgehaald kunnen worden, shield aantal ingangen/uitgangen kanalen uitgelezen kunnen worden, een “Logging” mapje op het prototype zelf aangemaakt kan worden en de logging naar een bestand weggeschreven kan worden.

LetsGrowWebApiController:

Deze klasse is verantwoordelijk voor het opzetten van de verbinding tussen het prototype applicatie en de LetsGrow.com Web Server door middel van de REST architectuur principe. De methodes in deze klasse zorgen ervoor dat er met het LetsGrow server device validatie verkregen kan worden, lijst van data record opgehaald kunnen worden, nieuwe lijst van data record verzonden kunnen worden, configuratie check uitgevoerd kan worden en configuratie van de device verzonden kan worden.

Server klassen:

EmbeddedDevicesController:

Deze klasse is verantwoordelijk voor het opzetten van de verbinding tussen de LetsGrow.com Web Server en het prototype applicatie door middel van de REST architectuur principe. De methodes in deze klasse zorgen ervoor dat er device validatie ontvangen kan worden, configuratiegegevens ontvangen kunnen worden, Data ontvangen kan worden van de device en data verstuurd kan worden naar de device.

EmbeddedDeviceInfo:

Deze klasse is verantwoordelijk voor het beheren van de configuratie, validatie en data in de SQL database van LetsGrow.com. De methodes in deze klasse zorgen ervoor dat er configuratie nagekeken kan worden in de SQL database, configuratie gestuurd kan worden naar de SQL database, Validatie van de device uitgevoerd kan worden, data opgevraagd kan worden van de SQL database voor de desbetreffende device, data van device die verkregen is, doorgestuurd kan worden naar de SQL database.

11.5 Klassendiagram beschrijvingen.

Client klassen:

StartupTask klasse is de 'main' van de applicatie en bezit de 'business logic'. Vanuit hier worden allerlei acties ondernomen om het prototype juist te laten werken.

StartupTask	Methodes
Run(IBackgroundTaskInstance taskInstance)	Deze methode is verantwoordelijk voor het opstarten van de achtergrond applicatie en voortdurend draaien in de achtergrond als een achtergrond proces.
Timer_Tick_60Minute(ThreadPoolTimer threadPoolTimer)	Deze methode is verantwoordelijk voor het aanmaken van de 60 minuten timer, zodat het prototype om de 60 minuten gaat initialiseren, indien er de eerste keer tijdens het opstarten geen validatie kan plaatsvinden.
Timer_Tick_5Minute(ThreadPoolTimer threadPoolTimer)	Deze methode is verantwoordelijk voor het aanmaken van de 5 minuten timer, zodat het prototype om de 5 minuten zijn werking kan uitvoeren.
Initialization()	Deze methode zorgt voornamelijk voor dat het prototype zichzelf eenmalig initialiseert bij het opstarten. (valideren door middel van zijn MAC-adres bij de LetsGrow.com servers).
Action()	Deze methode zorgt ervoor dat alle acties die het prototype moet uitvoeren, ook daadwerkelijk worden uitgevoerd.

EmbeddedDeviceController klasse zorgt voor het beheren van het ADC-DAC PI shield component.

EmbeddedDeviceController	Methodes
GetDeviceConfiguration()	Deze methode is verantwoordelijk voor het opvragen van de 'device configuratie informatie' van de Raspberry Pi
Connect(double referenceVoltage)	Deze methode is verantwoordelijk voor het opzetten van de verbinding met de ADC-DAC PI shield.
ReadInputVoltage(byte channel)	Deze methode is verantwoordelijk voor het verkrijgen van de ingang-waardes van de ADC-DAC Pi shield
WriteAdcDacOutputVoltage(byte channel, double voltage)	Deze methode zorgt voornamelijk voor het wegschrijven van voltages naar de desbetreffende uitgangen
InputOutputChannelsCount(string inputOutputChannel)	Deze methode zorgt ervoor dat het totale aantal in- en uitgangen van de ADC-DAC Pi opgevraagd kan worden
GetDeviceMacAddress()	Deze methode zorgt er voornamelijk voor dat het MAC-adres van het prototype verkregen wordt.
GetDeviceIpAddress()	Deze methode zorgt er voornamelijk voor dat het IP-adres van het prototype verkregen wordt.
GetJsonStreamData(String URL)	Deze methode zorgt er voornamelijk voor dat de ipconfig (wat is dat?) van de Pi als een JsonStreamData verkregen wordt. Deze methode is benodigd om het MAC-adres van het prototype op te kunnen halen.
SetGreenRedLeds(GpioPinValue greenValueHighLow, GpioPinValue redValueHighLow)	Deze methode zorgt ervoor dat de groene en rode "leds" van de PI op een hoog waarde of laag waarde gezet kunnen worden.

De LetsGrowWebApiController klasse zorgt ervoor dat er communicatie opgezet kan worden met de LetsGrow.com server. Deze klasse handelt de HTTP 'requests' af.

LetsGrowWebApiController	Methodes
SetConfiguration(string macAddress, EmbeddedDeviceInformation embeddedDeviceConfig)	Deze methode is verantwoordelijk voor het sturen van de configuratie informatie van de 'device' naar de LetsGrow.com server.
CheckDeviceConfiguration(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van de configuratie informatie van de 'device' van de LetsGrow.com server.
GetDataRecord(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van een lijst van het type "data record" van de LetsGrow.com server.
SetDataRecord(string macAddress, IEnumerable embeddedDeviceDataRecord)	Deze methode is verantwoordelijk voor het sturen van een lijst van het type "data record" naar de LetsGrow.com server.
GetDeviceValidation(string macAddress)	Deze methode is verantwoordelijk voor het valideren van een device door middel van een MAC-adres bij de LetsGrow.com server.

Server klassen:

De EmbeddedDeviceInfo klasse is de 'base' klasse voor Embedded apparaten van de LetsGrow.com Server applicatie en bezit de 'business logic'.

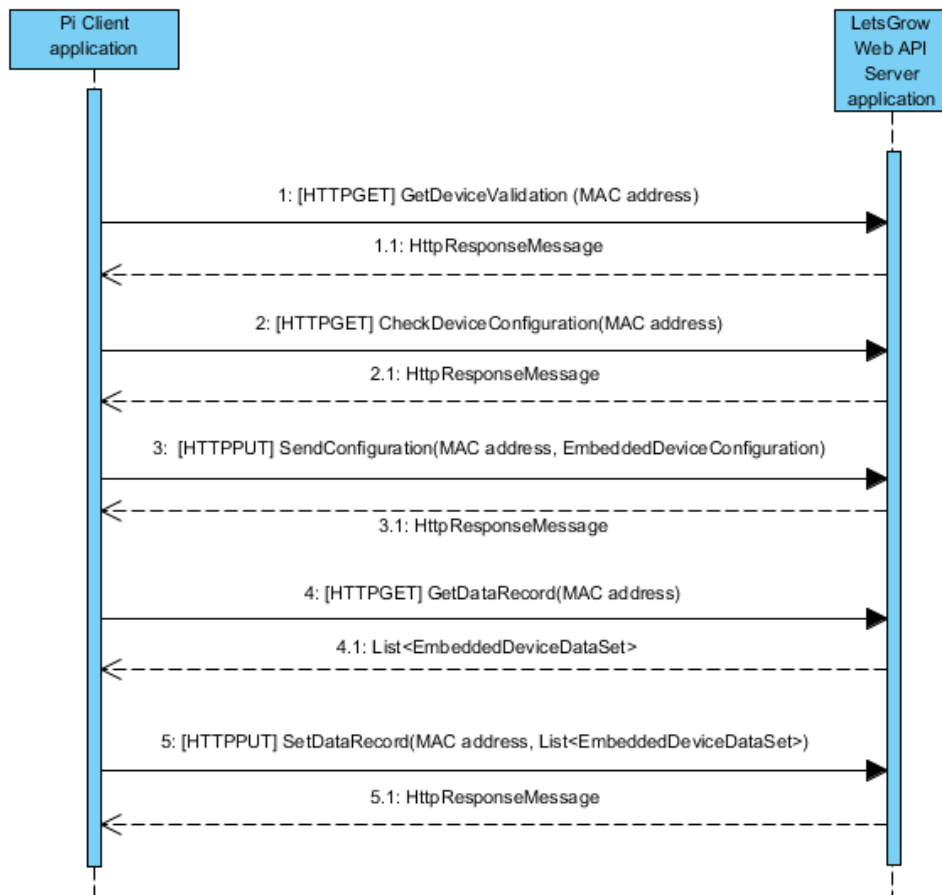
EmbeddedDeviceInfo	Methodes
RetrieveDataRecordValuesForDevice(int colTypeld, int deviceTypeld, string deviceIdentity, string itemCode)	Deze methode is verantwoordelijk voor het ophalen van de data record waardes vanuit de SQL ¹ database tabel door middel van de parameters colTypeld, deviceTypeld, deviceId en itemCode.
SetDataRecordValuesFromDevice(List EmbeddedDeviceDataRecord)	Deze methode is verantwoordelijk voor het versturen van de data record waardes naar de SQL database tabel.
SetConfiguration(EmbeddedDeviceInformation deviceConfiguration)	Deze methode is verantwoordelijk voor het versturen van de configuratie informatie van de desbetreffende 'device' naar de SQL database tabel.
CheckConfiguration(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van de configuratie informatie van de desbetreffende 'device' uit de SQL database tabel.
IsDeviceMacAddressValid(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van de MAC-adres van de desbetreffende 'device' uit de SQL ² database tabel.

EmbeddedDevicesController klasse zorgt ervoor dat er communicatie opgezet kan worden met de Client applicatie. Deze klasse handelt de HTTP 'requests' af.

LetsGrowWebApiController	Methodes
SetConfiguration(string macAddress, EmbeddedDeviceInformation embeddedDeviceConfig)	Deze methode is verantwoordelijk voor het doorsturen van de configuratie informatie van het 'device' naar de LetsGrow.com server.
CheckDeviceConfiguration(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van de configuratie informatie van het 'device' bij de LetsGrow.com server.
GetDataRecord(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van een lijst van data record bij de LetsGrow.com server.
SetDataRecord(string macAddress, IEnumerable embeddedDeviceDataRecord)	Deze methode is verantwoordelijk voor het sturen van een lijst van data record naar de LetsGrow.com server.
GetDeviceValidation(string macAddress)	Deze methode is verantwoordelijk voor het valideren van een device door middel van een MAC-adres bij de LetsGrow.com server.

11.6 Sequentiediagram.

De communicatie tussen het prototype en de LetsGrow.com servers wordt opgebouwd door middel van de REST API protocol. Er is voor REST API methode gekozen, omdat de LetsGrow.com Web API dit protocol ook gebruikt. Hierdoor past het beter in de huidige infrastructuur. Aan de hand van het globale sequentiediagram wordt verduidelijkt hoe de connectie en communicatie tussen de cliënt kant en server kant plaats gaat vinden.



Figuur 40: Sequentiediagram client-server communicatie

De prototype software(hierboven afgebeeld als de Pi cliënt application) gaat om het uur een HTTPGET doen om zijn MAC adres te laten valideren bij de LetsGrow server(hierboven afgebeeld als de LetsGrow Web API server application). De server stuurt dan een HttpResponseMessage terug met de uitkomst. Indien er geen validatie plaatst kan vinden, dan gaat het prototype de komende 1 uur in “Doe niets” status. Deze status houdt in dat er de komende 1 uur geen een actie verricht wordt.

Als de validatie proces goed doorlopen is, dan gaat het prototype om de 5 minuten configuratie checken doormiddel van een HTTPGET, waarbij hij een HttpResponseMessage terug krijgt of er daadwerkelijk configuratie aanwezig is aan de kant van de Server.

Als de bovenstaande procedure met succes is verlopen, zal de cliënt applicatie om de 5 minuten een “DataRecord” aanvraag doen door middel van een HTTPGET functie. Server beantwoordt deze met een Lijst van datarecord. Aan de hand van deze datarecord kijkt Client of er data nog teruggestuurd moet worden naar de Server. Indien dit geval is, zal de Client met een HTTPPUT functie de Lijst met vernieuwde datarecords terugsturen naar de Server. Server zal dan een HttpResponseMessage terugsturen. Deze procedure wordt ook om de 5 minuten herhaald.

Bibliografie

- ABelectronics ADC-DAC Pi*. (2016). Opgeroepen op Maart 7, 2016, van abelectronics:
<https://www.abelectronics.co.uk/p/39/ADC-DAC-Pi-Raspberry-Pi-ADC-and-DAC-expansion-board>
- ABelectronics Expander Pi*. (2016). Opgeroepen op Maart 6, 2016, van abelectronics:
<https://www.abelectronics.co.uk/p/50/Expander-Pi?CatName=0&c=3>
- Cooking Hacks 4-20mA Sensor Board*. (2016). Opgeroepen op Maart 7, 2016, van Cooking Hacks:
<https://www.cooking-hacks.com/documentation/tutorials/4-20ma-sensor-board-shield-tutorial-for-arduino-raspberry-pi-intel-galileo/>
- Cooking Hacks Shields For Raspberry Pi*. (2016). Opgeroepen op Maart 7, 2016, van Cooking-Hacks:
<https://www.cooking-hacks.com/lorawan-shield-for-raspberry-pi-433-mhz-xbee-socket>
- Ekuan, M. (2016). *Guid to Universal Windows Platform (UWP)*. Opgeroepen op Maart 10, 2016, van Windows Dev Center: <https://msdn.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide>
- Hanabusa, R. (2007, April 13). *comparing_serial_interfaces_an. Comparing Jtag, SPI, I2C*. Spanspion.
- Harrington, W. (2015). *Learning Raspbian*. 35 Livery Street, Birmingham: Packt Publishing.
- Hendriks, P. (2016, Maart 14). *Huidige Situatie*. (E. Soy, Interviewer)
- Herbison. (2014). *Introducing the Raspberry Pi B+*. Opgeroepen op Maart 2, 2016, van Designspark:
<http://www.rs-online.com/designspark/electronics/eng/blog/introducing-the-raspberry-pi-b-plus>
- Learn SparkFun Tutorials/Arduino Shields*. (2016). Opgeroepen op Maart 2, 2016, van SparkFun:
<https://learn.sparkfun.com/tutorials/arduino-shields>
- Linden, A. v. (2014, Augustus 15). *Construction EtherCan. Intern PDF documention*. Vlaardingen: Hoogendoorn.
- Linden, A. v. (2014, Augustus 15). *Construction iSii Hardware. Intern PDF documention*. Vlaardingen, Zuid-Holland: Hoogendoorn.
- Long, S. (2014, December 28). *Changes to the raspbian user interface*. Opgeroepen op Maart 10, 2016, van Raspberrypi.org: <https://www.raspberrypi.org/blog/changes-to-the-raspbian-user-interface/>
- MikroElektronika. (2016). *Mikro 4-20mA T click shield*. Opgeroepen op Maart 7, 2016, van MikroElektronika: <http://www.mikroe.com/click/4-20ma-t/>
- Miltenburg, O. v. (2015, Februari 2). *Microsoft kondigt Windows 10 voor Raspberry Pi 2 aan*. Opgeroepen op Maart 10, 2016, van Tweakers.net:
<http://tweakers.net/nieuws/101142/microsoft-kondigt-windows-10-voor-raspberry-pi-2-aan.html>

- Monk, S. (2015, April 5). *pwm*. Opgeroepen op Maart 12, 2016, van adafruit:
<https://learn.adafruit.com/adafruit-raspberry-pi-lesson-9-controlling-a-dc-motor/pwm>
- Osexpansionboard.de. (2016). Opgeroepen op Maart 8, 2016, van Osexpansionboard.de:
<http://www.osexpansionboard.de/>
- Phidgets.com. (2012). *Phidget 1002_0 - PhidgetAnalog 4-Output*. Opgeroepen op Maart 7, 2016, van Phidgets.com: http://www.phidgets.com/products.php?product_id=1002
- Phidgetsinc. (sd). *Getting started with Phidgets on the Raspberry Pi*. Opgeroepen op Maart 12, 2016, van instructables.com: <http://www.instructables.com/id/Getting-Started-with-Phidgets-on-the-Raspberry-Pi/>
- Posthumus, C. M. (2016, Maart 4). Wat zijn klimaatcomputers/regelingen. (E. Soy, Interviewer)
- Raspberry Pi Forums. (2016). Opgeroepen op Maart 8, 2016, van Disable updates in Windows IoT Core: <https://www.raspberrypi.org/forums/viewtopic.php?f=105&t=146087>
- Raspberry Pi Foundation. (2015, Maart). The MagPi Magazine Issue 31. *Raspberry Pi* 2. 341 Hawthorne Avenue, Palo Alto, CA 94301, US: Weaved.
- Raspberry Pi Foundation. (2015, Juni). The MagPi Magazine Issue 34. *Windows 10 On Raspberry Pi*. 41 Hawthorne Avenue, Pali Alto, CA 94301, US: Weaved.
- Raspberry Pi Foundation. (2016). *Documentation configuration Raspi-config*. Opgeroepen op Maart 10, 2016, van Raspberrypi.org:
<https://www.raspberrypi.org/documentation/configuration/raspi-config.md>
- Raspberry Pi Foundation. (2016). *Usage GPIO pins*. Opgeroepen op Maart 8, 2016, van Raspberrypi.org: <https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/>
- Raspberry Pi Software Languages. (2016). Opgeroepen op Maart 10, 2016, van Raspberrypi.org:
<https://www.raspberrypi.org/help/faqs/#softwareLanguages>
- Raspberry Pi Software OS. (2016). Opgeroepen op Maart 10, 2016, van Raspberrypi.org:
<https://www.raspberrypi.org/help/faqs/#softwareOS>.
- Raspberry# IO repository. (2016). Opgeroepen op Maart 12, 2016, van Github.com:
<https://github.com/raspberry-sharp/raspberry-sharp-io>
- RaspberryPi.net. (2016). Opgeroepen op Maart 11, 2016, van Github.com:
<https://github.com/cypherkey/RaspberryPi.Net>
- RaspiSharp. (2016). Opgeroepen op Maart 11, 2016, van Github.com:
<https://github.com/gusmanb/RaspiSharp>
- Richard, W. (2014). *Teach yourself VISUALLY Raspberry Pi*. Visual: 10475 Crosspoint Boulevard, Indianapolis, IN 46256 1 edition.
- Robillard, S. (2015, December 31). *Raspberry Pi StackExchange GPIO pins*. Opgeroepen op maart 3, 2016, van Raspberry Pi StackExchange:

- <http://raspberrypi.stackexchange.com/questions/40318/raspberry-pi-2-can-gpio-pins-29-40-be-used-gpio-gen-input-output-configurable-in>
- Scheduling Tasks with Cron.* (2016). Opgeroepen op Maart 11, 2016, van Raspberrypi.org: <https://www.raspberrypi.org/documentation/linux/usage/cron.md>
- Schipper, J. (2015, April 2). Commissioning values various senso. *Intern PDF documentation*. Hoogendoorn: Vlaardingen.
- Schipper, J. (2016, Maart 1). Werking Klimaatcomputers. (E. Soy, Interviewer)
- Sean McManus and Mike Cook. (2013). *Raspberry Pi for Dummies 1 edition*. 111 River Street, Hoboken: John Wiley & Sons, Inc.
- SF innovations Custard Pi 2.* (2016). Opgeroepen op Maart 7, 2016, van SF innovations: <http://www.sf-innovations.co.uk/custard-pi-2.html>
- Sparkfun Tutorials I2C.* (2016). Opgeroepen op Maart 10, 2016, van Sparkfun Tutorials I2C: <https://learn.sparkfun.com/tutorials/i2c>
- Sparkfun Tutorials Serial Communication.* (2016). Opgeroepen op Maart 10, 2016, van Sparkfun Tutorials UARTS: <https://learn.sparkfun.com/tutorials/serial-communication/uarts>
- Sparkfun Tutorials SPI.* (2016). Opgeroepen op Maart 10, 2016, van Sparkfun Tutorials Serial peripheral interface SPI: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>
- Tielens, J. (2015). *Getting started with the raspberry pi 2 for net developers*. Opgeroepen op Maart 11, 2016, van Jan Tielens Spot: <http://j.tlms.be/2015/02/04/getting-started-with-the-raspberry-pi-2-for-net-developers/>
- UniPi.technology Products.* (2016). Opgeroepen op Maart 7, 2016, van UniPi.technology: <http://unipi.technology/product/unipi/>
- Upton, L. (2016). *What hardware interfaces does it have?* Opgeroepen op Maart 11, 2016, van Raspbian.org: <https://www.raspbian.org/RaspbianFAQ>
- Upton, L. (2016). *What operating system (os) does it use?* Opgeroepen op Maart 10, 2016, van Raspberrypi.org: <https://www.raspberrypi.org/help/faqs/>
- Vis, E. (2016, Februari). LetsGrow Web API. *Interface description LetsGrow Web API Version 1.5*. Vlaardingen: LetsGrow.com.
- Visual Studio 2015 product editions.* (2016). Opgeroepen op Maart 9, 2016, van Visual Studio: <https://www.visualstudio.com/en-us/products/vs-2015-product-editions.aspx>
- What is MonoDevelop.* (2016). Opgeroepen op Maart 11, 2016, van Monodevelop.com: <http://www.monodevelop.com/help/faq/>
- What is Windows IoT Core.* (2016). Opgeroepen op Maart 8, 2016, van Microsoft Dev Center: <https://developer.microsoft.com/en-us/windows/iot/iotcore>

Windows Dev Center. (2016). *Raspberry Pi 2 & 3 Pin Mappings*. Opgeroepen op Maart 10, 2016, van Windows Dev Center: <https://developer.microsoft.com/en-us/windows/iot/win10/samples/PinMappingsRPi2.htm>

Windows IoT Headed and Headless mode. (2016). Opgeroepen op Maart 10, 2016, van Windows Dev Center: <https://developer.microsoft.com/en-us/windows/iot/win10/headlessmode>

Windows IoT Starter Pack Contents. (2016). Opgeroepen op Maart 8, 2016, van Microsoft Dev Center: <https://developer.microsoft.com/en-us/windows/iot/win10/adafruitkitcontents>

Windows IoT Starter Pack Overview. (2016). Opgeroepen op Maart 8, 2016, van Microsoft Dev Center: <https://developer.microsoft.com/en-us/windows/iot/win10/kitsetupcrpi>

K Ontwerprapport

Ontwerprapport

Data wegschrijven naar klimaatcomputers

Bedrijf: LetsGrow.com
Versie: 1.0
Datum: 13-05-2015
Plaats: Vlaardingen

Opdrachtgever:
Peter Hendriks

Bedrijfsmentoren:
Leon Batta/Edwin Vis

Documenthistorie:

Datum	Versie	Beschrijving
4-5-2016	0.5	Concept
13-5-2016	1.0	Definitief

Distributie:

Naam	Versie Concept	Versie Definitief
Peter Hendriks	X	
Leon Batta		
Edwin Vis	X	X

Akkoord opdrachtgever en projectleider.

Peter Hendriks	Eray Soy
Handtekening:	Handtekening:

Inhoudsopgave

1.	Inleiding	1
2.	Wijzingen ten opzichte van het definitierapport.	2
3.	Klassendiagrammen	4
3.1	Client klassendiagram.....	4
3.2	Client klassendiagram beschrijvingen:	5
3.3	Server klassendiagram.....	7
3.4	Server klassendiagram beschrijvingen:	8
4.	Sequentiediagrammen	10
4.1	Client-server communicatie sequentiediagram:.....	10
4.2	Client applicatie sequentiediagram:.....	11
4.3	Server applicatie sequentiediagram:.....	14

1. Inleiding

Dit ontwerprapport bevat het ontwerp voor de software van het prototype. Om het prototype goed te kunnen bouwen was er van te voren een architectuur opgesteld. Deze stond in het definitierapport. Aan de hand van dit rapport is er doorgegaan met het ontwerprapport. De wijzigingen ten opzichte van het definitierapport zullen eerst vermeld worden voordat er verder gegaan wordt met beschrijvingen en de sequentiediagrammen.

2. Wijzingen ten opzichte van het definitierapport.

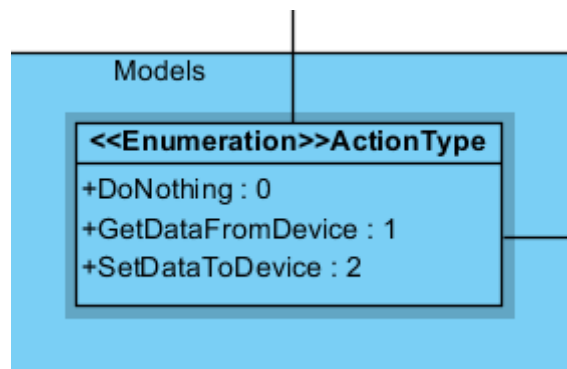
Client klassendiagram:

In EmbeddedDeviceController zijn er 5 methodes en 4 variabelen bijgekomen.

De 5 methodes zijn benodigd om de eerst gedefinieerde functies asynchroon op te kunnen roepen. Hiervoor moest dus een soort “wrapper” gebouwd worden. Daarnaast zijn er 4 variabelen bijgekomen, omdat de librerie van het desbetreffende shield niet de analoge ingangen en uitgangen aantallen weergeeft. Om de ‘device’ configuratie informatie compleet te maken zijn deze gegevens hard-coded ingezet.

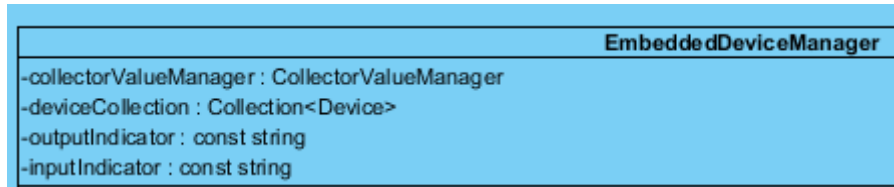
-analogIn : string
-analogOut : string
-digitalIn : string
-digitalOut : string
+GetDeviceConfigurationOperation() : IAsyncOperation<EmbeddedDeviceInformation>
+WriteAdcDacOutputVoltageOperation(byte channel, double voltage) : IAsyncOperation<bool>
+GetDeviceMacAddressOperation() : IAsyncOperation<string>
+GetDeviceIpAddressOperation() : IAsyncOperation<string>
+SetGreenRedLedsOperation(GpioPinValue greenValueHighLow, GpioPinValue redValueHighLow) : IAsyncAction

In het Models mapje is: ‘Enumeratie van ActionType’ toegevoegd. Eerst was de bedoeling om deze als een integer te noteren, alleen dan werd het in de code wat onoverzichtelijker. Om de code overzichtelijker te houden is deze enumeratie toegevoegd.



Server klassendiagram:

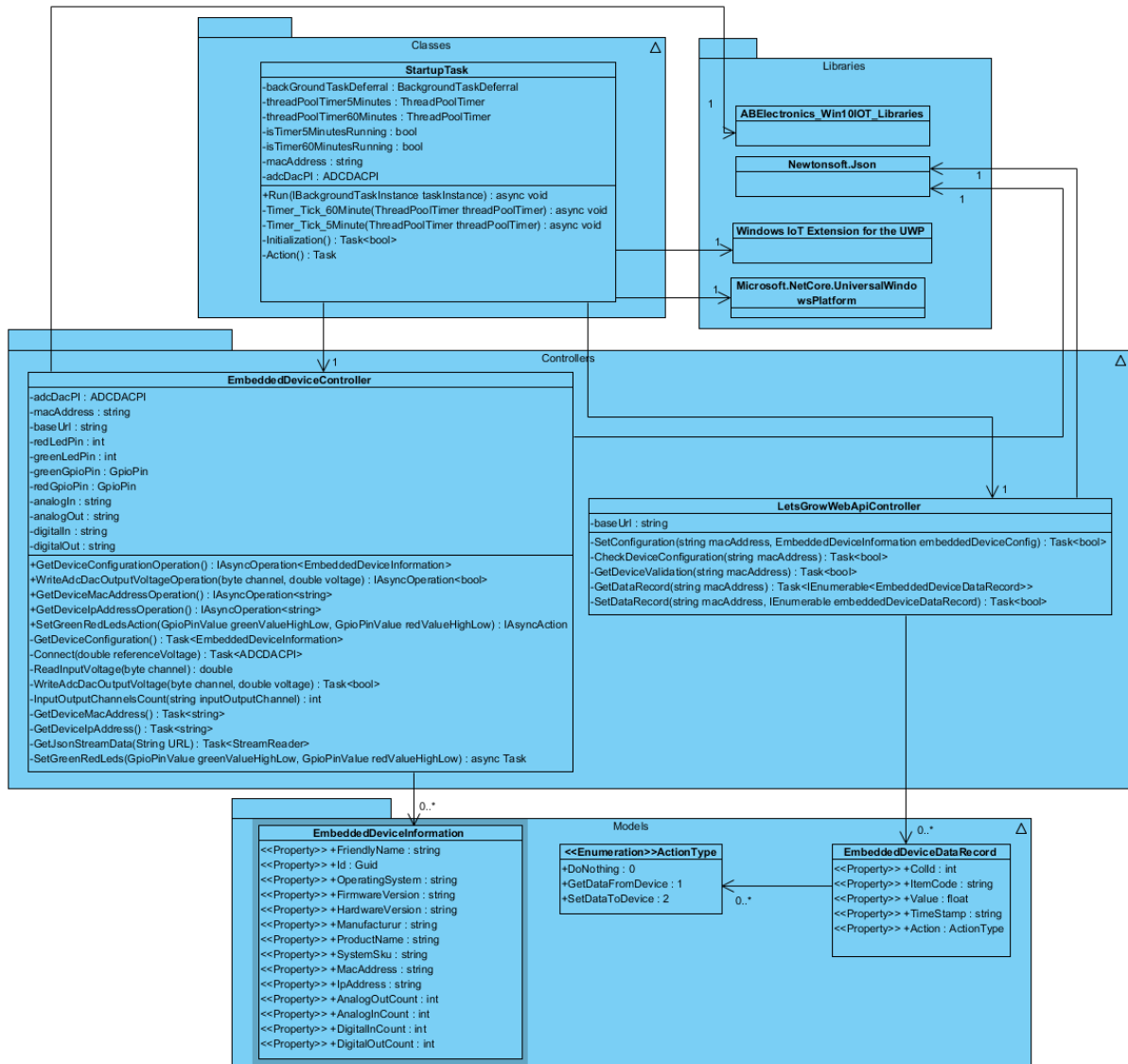
De naam van de klasse EmbeddedDeviceInfo is veranderd naar EmbeddedDeviceManager, omdat er ook al een model klasse bestond (in de LetsGrow.com Web API server applicatie) met de naam EmbeddedDeviceInformation. Deze zorgde voor verwarring. Daarnaast zijn de variabelen in deze functie gewijzigd van 'public' naar 'private', omdat deze variabelen buiten de klasse niet gebruikt worden.



Ook bij de server is in het Models mapje: 'Enumeratie van ActionType' toegevoegd. Deze is toegevoegd om de applicaties aan de server kant gelijk te trekken met de cliënt kant om verwarringen te voorkomen. Deze verwarringen konden ontstaan doordat op de server anders alleen maar een 0,1 of 2 te zien was. Hierdoor kan een andere ontwikkelaar(persoon) niet weten wat een 0,1 of 2 inhoudt. Tevens zorgt deze klasse ervoor dat het code overzichtelijker wordt.

3. Klassendiagrammen

3.1 Client klassendiagram



3.2 Client klassendiagram beschrijvingen:

StartupTask klasse is de 'main' van de applicatie en bezit de 'business logic'. Vanuit hier worden allerlei acties ondernomen om het prototype juist te laten werken.

StartupTask	Methodes
Run(IBackgroundTaskInstance taskInstance)	Deze methode is verantwoordelijk voor het opstarten van de achtergrond applicatie en voortdurend draaien in de achtergrond als een achtergrond proces.
Timer_Tick_60Minute(ThreadPoolTimer threadPoolTimer)	Deze methode is verantwoordelijk voor het aanmaken van de 60 minuten timer, zodat het prototype om de 60 minuten gaat initialiseren, indien er de eerste keer tijdens het opstarten geen validatie kan plaatsvinden.
Timer_Tick_5Minute(ThreadPoolTimer threadPoolTimer)	Deze methode is verantwoordelijk voor het aanmaken van de 5 minuten timer, zodat het prototype om de 5 minuten zijn werking kan uitvoeren.
Initialization()	Deze methode zorgt voornamelijk voor dat het prototype zichzelf eenmalig initialiseert bij het opstarten. (valideren door middel van zijn MAC-adres bij de LetsGrow.com servers).
Action()	Deze methode zorgt ervoor dat alle acties die het prototype moet uitvoeren, ook daadwerkelijk worden uitgevoerd.

EmbeddedDeviceController klasse zorgt voor het beheren van het ADC-DAC PI shield component.

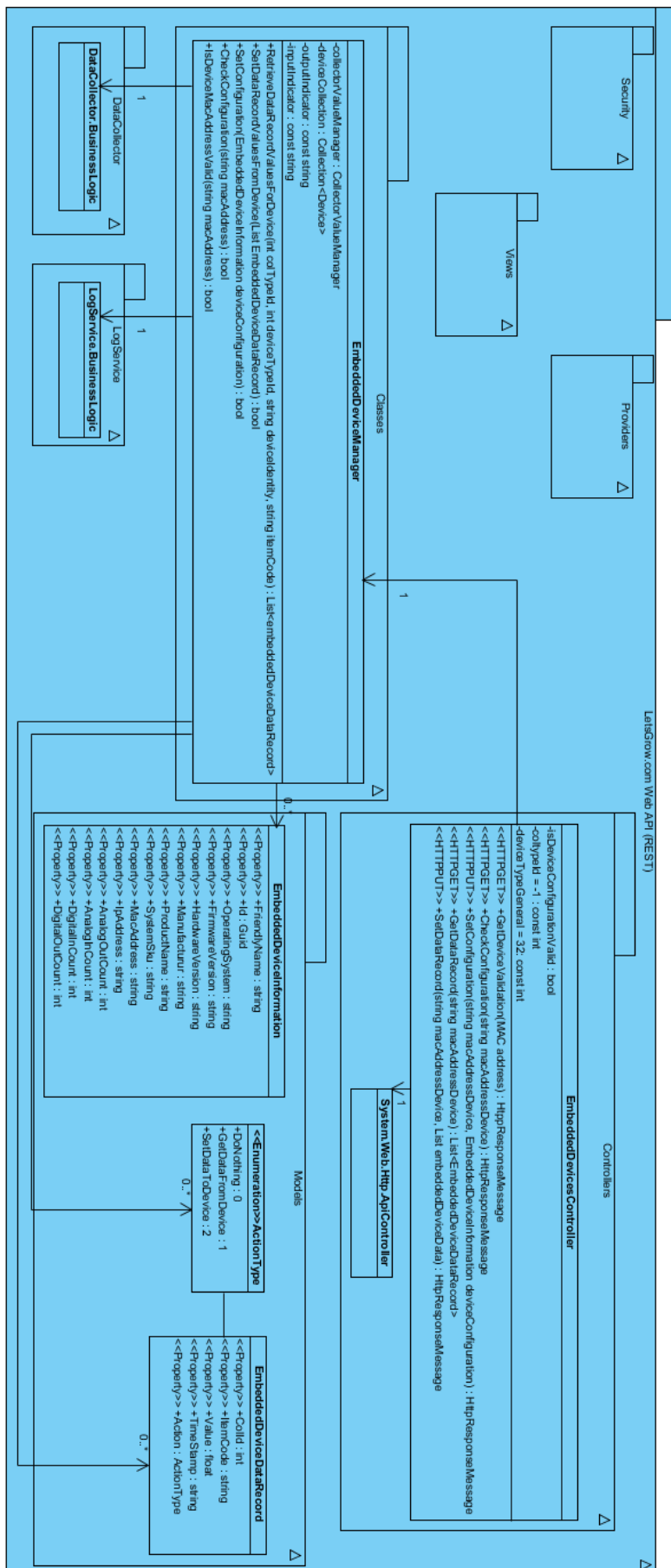
EmbeddedDeviceController	Methodes
GetDeviceConfiguration()	Deze methode is verantwoordelijk voor het opvragen van de 'device configuratie informatie' van de Raspberry PI
Connect(double referenceVoltage)	Deze methode is verantwoordelijk voor het opzetten van de verbinding met de ADC-DAC PI shield.
ReadInputVoltage(byte channel)	Deze methode is verantwoordelijk voor het verkrijgen van de ingang-waardes van de ADC-DAC Pi shield
WriteAdcDacOutputVoltage(byte channel, double voltage)	Deze methode zorgt voornamelijk voor het wegschrijven van voltages naar de desbetreffende uitgangen
InputOutputChannelsCount(string inputOutputChannel)	Deze methode zorgt ervoor dat het totale aantal in- en uitgangen van de ADC-DAC Pi opgevraagd kan worden
GetDeviceMacAddress()	Deze methode zorgt er voornamelijk voor dat het MAC-adres van het prototype verkregen wordt.
GetDeviceIpAddress()	Deze methode zorgt er voornamelijk voor dat het IP-adres van het prototype verkregen wordt.
GetJsonStreamData(String URL)	Deze methode zorgt er voornamelijk voor dat de ipconfig (wat is dat?) van de Pi als een JsonStreamData verkregen wordt. Deze methode is benodigd om het MAC-adres van het prototype op te kunnen halen.
SetGreenRedLeds(GpioPinValue greenValueHighLow, GpioPinValue redValueHighLow)	Deze methode zorgt ervoor dat de groene en rode "leds" van de PI op een hoog waarde of laag waarde gezet kunnen worden.

Note: Methodes met Operation(); zijn wrappers. Windows IoT applicaties ondersteunen niet 'native' de async Task<> functie om te 'awaiten'. Hierdoor moest er een soort wrapper eromheen gebouwd worden door middel van IAsyncOperation, waarbij op deze operaties gewacht kan worden.

De LetsGrowWebApiController klasse zorgt ervoor dat er communicatie opgezet kan worden met de LetsGrow.com server. Deze klasse handelt de HTTP 'requests' AF.

LetsGrowWebApiController	Methodes
SetConfiguration(string macAddress, EmbeddedDeviceInformation embeddedDeviceConfig)	Deze methode is verantwoordelijk voor het sturen van de configuratie informatie van de 'device' naar de LetsGrow.com server.
CheckDeviceConfiguration(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van de configuratie informatie van de 'device' van de LetsGrow.com server.
GetDataRecord(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van een lijst van het type "data record" van de LetsGrow.com server.
SetDataRecord(string macAddress, IEnumerable embeddedDeviceDataRecord)	Deze methode is verantwoordelijk voor het sturen van een lijst van het type "data record" naar de LetsGrow.com server.
GetDeviceValidation(string macAddress)	Deze methode is verantwoordelijk voor het valideren van een device door middel van een MAC-adres bij de LetsGrow.com server.

3.3 Server klassendiagram







3.4 Server klassendiagram beschrijvingen:

De EmbeddedDeviceManager klasse is de 'base' klasse voor Embedded apparaten van de LetsGrow.com Server applicatie en bezit de 'business logic'.

Vanuit hier worden allerlei acties ondernomen om data te kunnen verzamelen en indien nodig te kunnen versturen naar de SQL databases.








EmbeddedDeviceManager	Methodes
RetrieveDataRecordValuesForDevice(int colTypeId, int deviceTypeId, string deviceIdentity, string itemCode)	Deze methode is verantwoordelijk voor het ophalen van de data record waardes vanuit de SQL ¹ database tabel door middel van de parameters colTypeId, deviceTypeId, deviceId en itemCode.
SetDataRecordValuesFromDevice(List EmbeddedDeviceDataRecord)	Deze methode is verantwoordelijk voor het versturen van de data record waardes naar de SQL database tabel.
SetConfiguration(EmbeddedDeviceInformation deviceConfiguration)	Deze methode is verantwoordelijk voor het versturen van de configuratie informatie van de desbetreffende 'device' naar de SQL database tabel.
CheckConfiguration(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van de configuratie informatie van de desbetreffende 'device' uit de SQL database tabel.
IsDeviceMacAddressValid(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van de MAC-adres van de desbetreffende 'device' uit de SQL ² database tabel.

1. De SQL database tabel die voor deze toepassing gebruikt wordt, is de tValue tabel in de DataCollector database. De structuur ziet er als volgt uit:

TValue			
	ColID	integer(10)	
	Utc	time(7)	
	UtcOffset	smallint(5)	N
	Val	real(10)	N

Door middel van de itemCode (ColID) wordt de waarde opgehaald. Voordat deze actie kan gebeuren, wordt eerst op colTypeId, deviceTypeId, en deviceId gekeken. Deze gebeurt dan weer door middel van het tabel TCollectorType, TDeviceType, TDevice.

2. De SQL database tabel die wordt gebruikt is de tDevice tabel in de DataCollector database. De structuur ziet er als volgt uit:

TDevice			
	DeviceID	integer(10)	
	DeviceTypeID	smallint(5)	N
	DeviceCode	varchar(128)	N
	DeviceDescription	varchar(128)	N
	DeviceZoneId	integer(10)	N
	DeviceLongitude	float(10)	N
	DeviceLatitude	float(10)	N

De MAC adres van het desbetreffende apparaat staat geregistreerd in de 'DeviceCode'. Vanuit hier wordt er een validatie proces uitgevoerd.

EmbeddedDevicesController klasse zorgt ervoor dat er communicatie opgezet kan worden met de Client applicatie. Deze klasse handelt de HTTP 'requests' af.

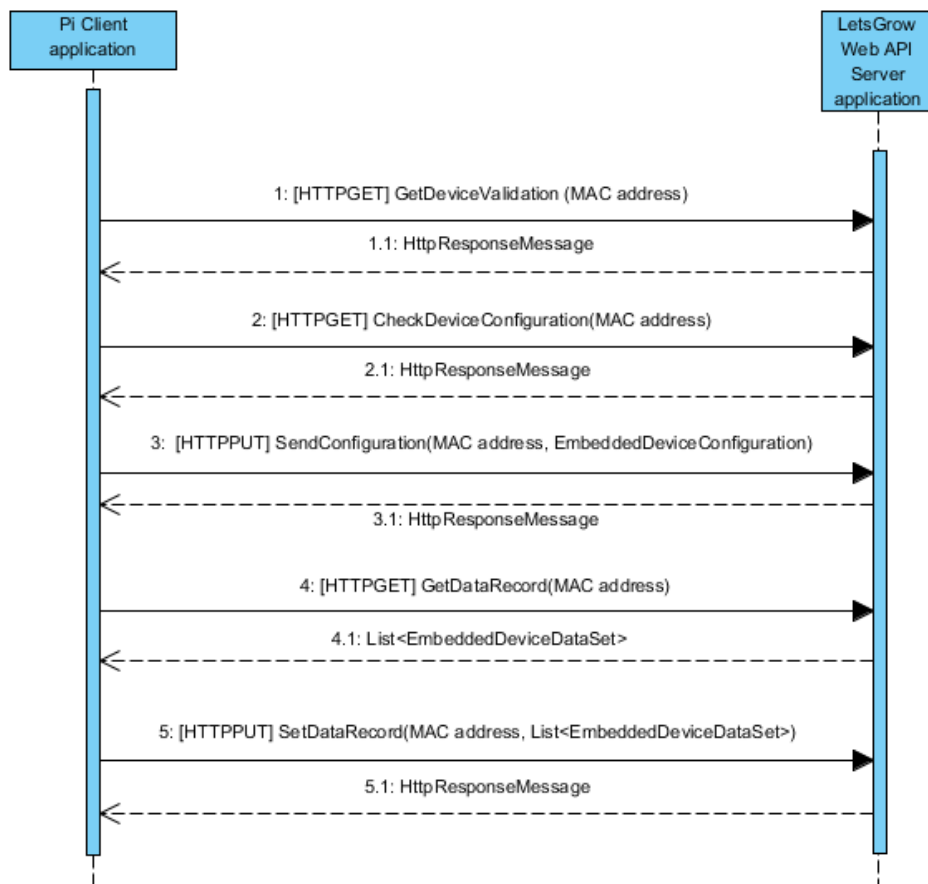
LetsGrowWebApiController	Methodes
SetConfiguration(string macAddress, EmbeddedDeviceInformation embeddedDeviceConfig)	Deze methode is verantwoordelijk voor het doorsturen van de configuratie informatie van het 'device' naar de LetsGrow.com server.
CheckDeviceConfiguration(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van de configuratie informatie van het 'device' bij de LetsGrow.com server.
GetDataRecord(string macAddress)	Deze methode is verantwoordelijk voor het opvragen van een lijst van data record bij de LetsGrow.com server.
SetDataRecord(string macAddress, IEnumerable embeddedDeviceDataRecord)	Deze methode is verantwoordelijk voor het sturen van een lijst van data record naar de LetsGrow.com server.
GetDeviceValidation(string macAddress)	Deze methode is verantwoordelijk voor het valideren van een device door middel van een MAC-adres bij de LetsGrow.com server.

4. Sequentiediagrammen

Hieronder volgen de beschrijvingen van de sequentiediagrammen. Om een duidelijk beeld te krijgen zijn alle functionaliteiten(use-cases) in een sequentiediagram ondergebracht om de flow van de applicaties beter te kunnen uitleggen. Deze sequentiediagrammen zullen ook afzonderlijk in een A3 formaat uitgeprint worden en als bijlage meegeleverd worden voor de leesbaarheid.

4.1 Client-server communicatie sequentiediagram:

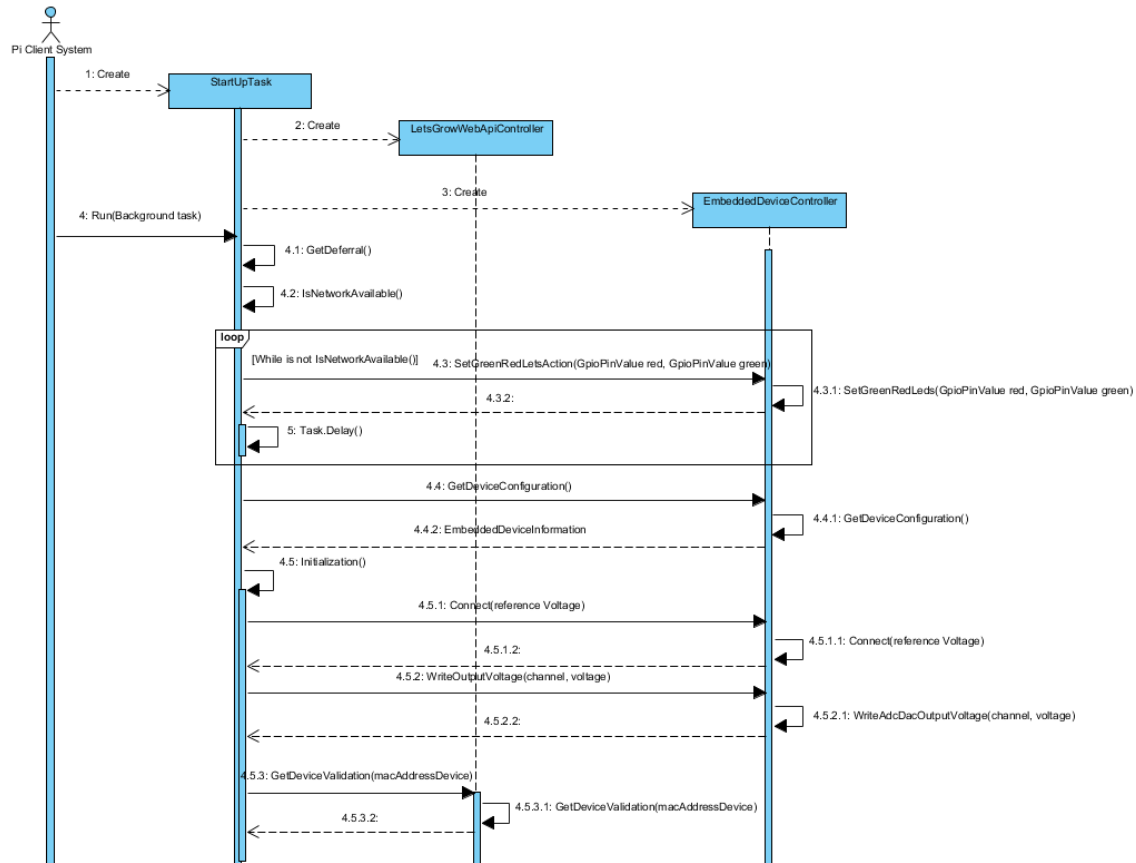
Hieronder zal het sequentiediagram van de communicatie tussen de Client en server beschreven worden.



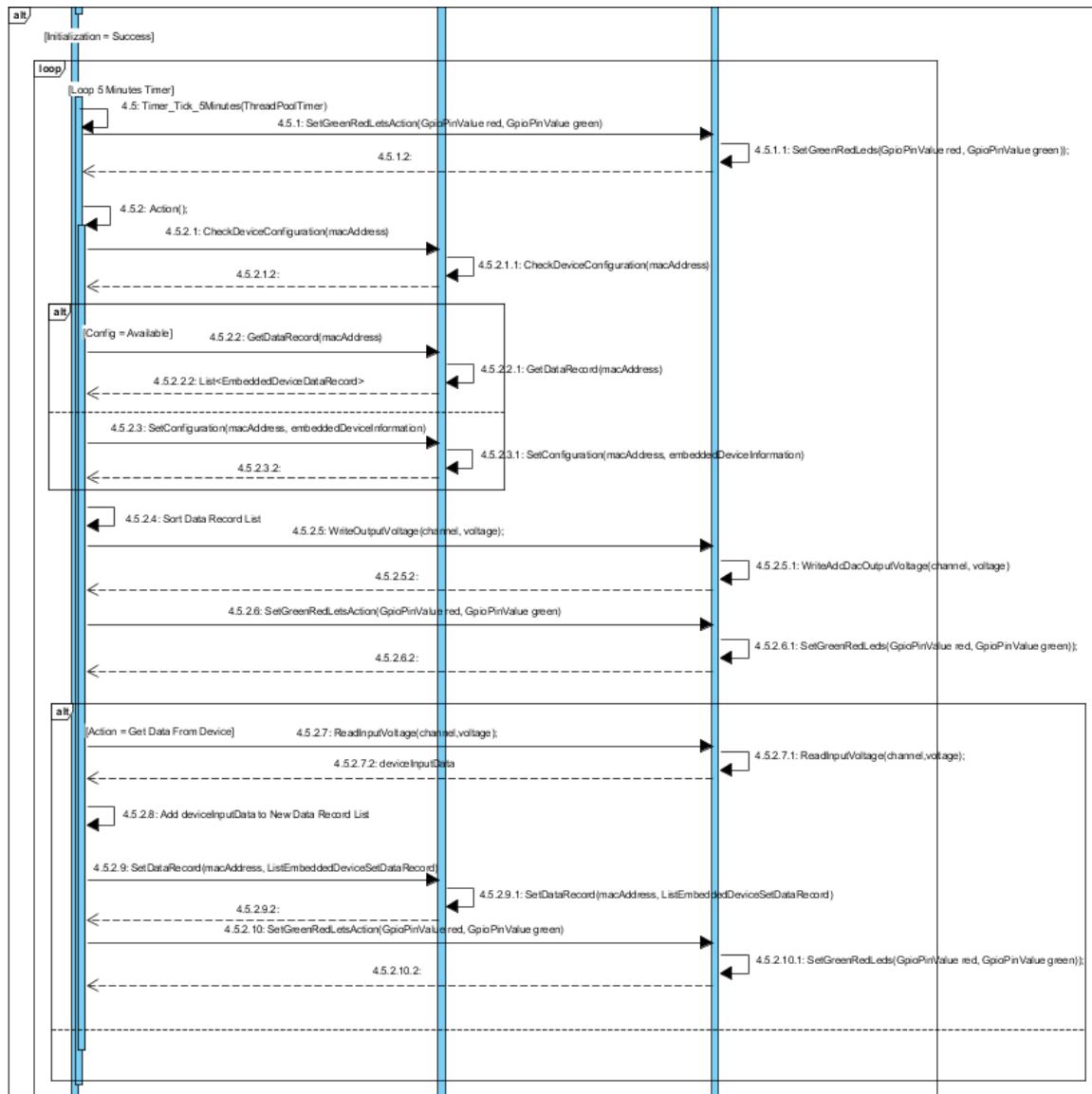
De communicatie tussen het prototype en de LetsGrow.com servers wordt opgebouwd door middel van het REST API protocol. Dit houdt in dat er HTTP 'request' berichten van de cliënt (embedded devices) naar de server(LetsGrow.com) gestuurd wordt. De server beantwoordt deze berichten met een "HttpResponseMessage" terug. Alleen bij het opvragen van de datarecord "GetDataRecord" stuurt de server een lijst van datarecord terug. De communicatie vindt op basis van het MAC adres plaats. Client moet elke keer zijn MAC adres opsturen, zodat de server aan de hand van deze MAC adres allerlei acties kan verrichten. Wat dan? Voor de communicatie is er gebruik gemaakt van 2 typen http functies, de "PUT" en de "GET" functies. "Put" functies zetten data in de databases en de "GET" functies halen data vanuit de databases op.

4.2 Client applicatie sequentiediagram:

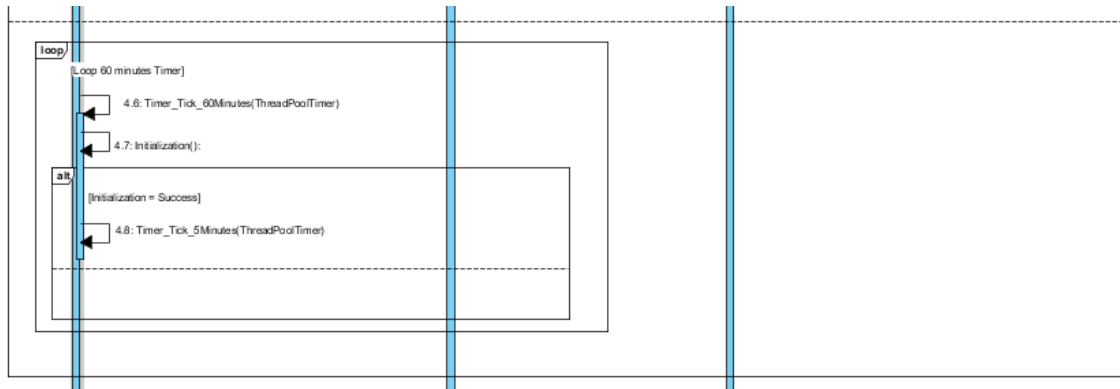
Hieronder zal het sequentiediagram van de cliënt applicatie beschreven worden. Deze sequentie diagram is opgedeeld in 3 stukken om duidelijker een inzicht te kunnen geven van bepaalde acties. Herformuleren!



Nadat het cliënt systeem opgestart is, wordt de applicatie automatisch in de achtergrond opgestart. De eerste actie die wordt uitgevoerd is het controleren of er internet toegang aanwezig is. Indien er geen netwerk aanwezig is, dan wordt de applicatie tijdelijk gepauzeerd en wordt op een later tijdstip nogmaals geprobeerd of er netwerktoegang mogelijk is. Als er geen netwerktoegang is wordt het groene en rode ledje van de Pi 2 aangezet (SetGreenRedLedsAction) na 30 seconden (Task.Delay) wordt weer gekeken of er een internetverbinding aanwezig is. Als er een actieve internetverbinding is, gaat hij eerst zijn apparaat configuratie informatie (GetDeviceConfiguration) ophalen. Nadat hij de apparaat configuratie informatie opgehaald heeft, gaat hij initialisatie (Initialization) uitvoeren. Dit houdt in dat hij eerst met het ADC-DAC PI shield connecteert (Connect) en een referentie voltage zet. Daarna schrijft hij naar de kanalen spanningen van "0" volt (WriteOutputVoltage) om het shield gereed te maken voor gebruik. Uiteindelijk gaat hij door middel van zijn MAC-adres (GetDeviceValidation) het apparaat valideren.



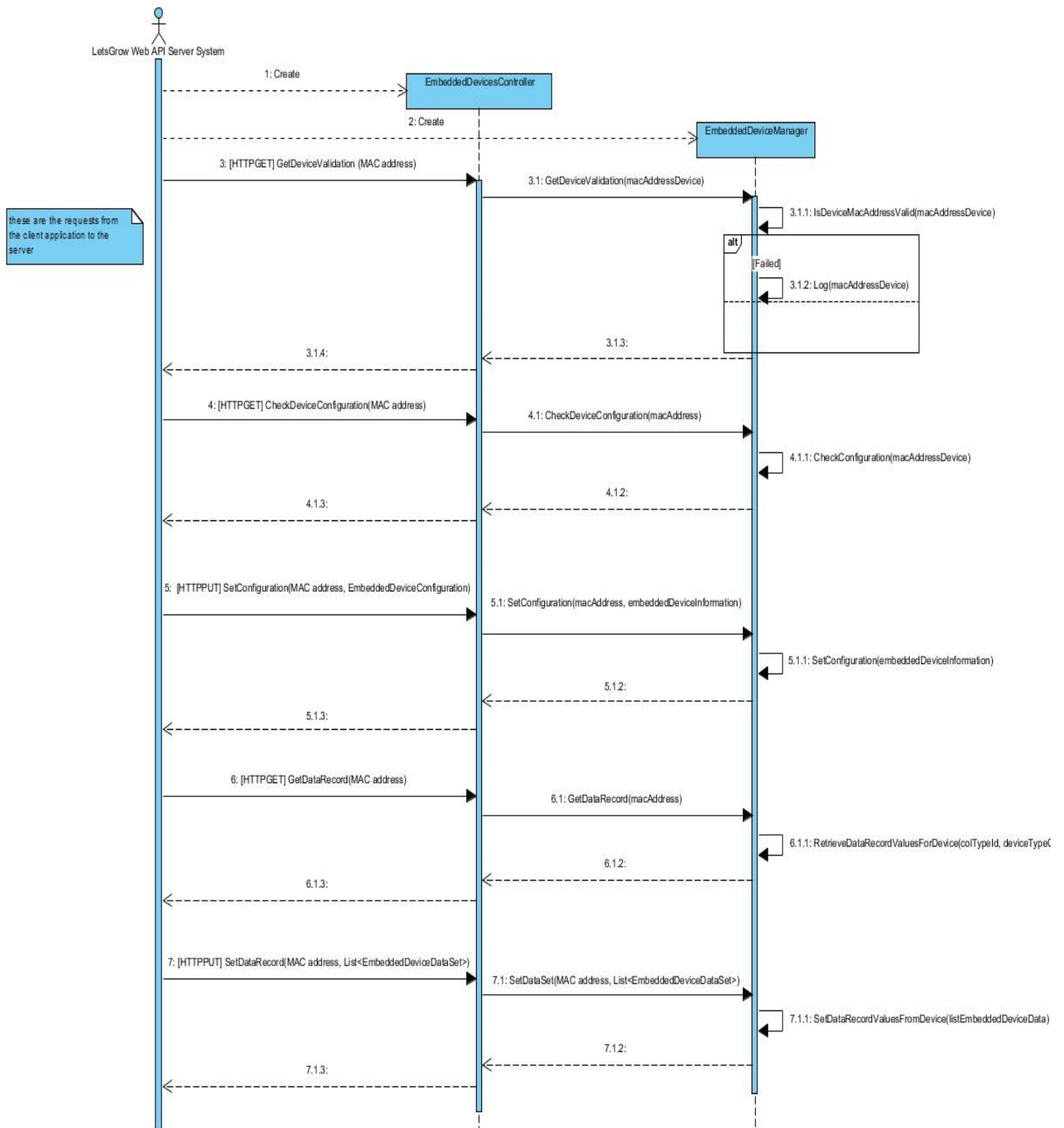
Als de initialisatie (Initialization) met succes uitgevoerd is, start het cliënt systeem een timer op van 5 minuten (Timer_Tick_5Minutes). Het cliënt systeem gaat dan zijn status tonen aan de hand van de groene en rode led (SetGreenRedLedsAction). Nadat hij dit heeft gedaan, gaat hij de actie (Action) uitvoeren. Dit houdt in dat hij eerst gaat kijken of er apparaat configuratie informatie (CheckDeviceConfiguration) op de servers aanwezig is. Indien dit niet het geval is, dan stuurt hij apparaat configuratie informatie (SetConfiguration) naar de server. Als de configuratie informatie aanwezig is, dan vraagt hij een "datarecord" lijst op (GetDataRecord). Van de server krijgt hij een lijst van datarecord (List<EmbeddedDeviceDataRecord>). Deze lijst van "datarecord" gaat hij sorteren en aan de hand van de actie die hij moet uitvoeren schrijft hij de voltages weg naar de desbetreffende kanalen (WriteOutputVoltage). Indien hij de spanning waarden van de ingangen moet ophalen, dan haalt hij deze ingang waarden op (ReadInputVoltage) en voegt hij deze waarden toe aan een nieuwe lijst van "datarecord". Deze nieuw aangemaakte lijst van "datarecord" stuurt hij vervolgens op naar de server (SetDataRecord). Van al deze zaken worden ook led statussen weergegeven (SetGreenRedLedsAction).



Indien het apparaat validatie niet lukt, dan wordt er een timer opgestart van 60 minuten (Timer_Tick_60Minutes). Deze timer gaat na 60 minuten draaien en gaat weer de initialisatie (Initialization) uitvoeren. Indien dit succesvol wordt uitgevoerd, wordt er een timer van 5 minuten (Timer_Tick_5Minutes) opgestart om de normale werking in gang te zetten.

4.3 Server applicatie sequentiediagram:

Hieronder zal het sequentiediagram van de server applicatie beschreven worden.



Als de server van het cliënt systeem een apparaat validatie bericht krijgt (GetDeviceValidation), dan kijkt hij in de database na of het MAC adres van het desbetreffende apparaat voorkomt. Als het voorkomt, stuurt hij een positief antwoord terug, anders een negatief antwoord en logt de server dit MAC adres naar de database.

Als de server van het cliënt systeem een “apparaat configuratie informatie check bericht” krijgt (CheckDeviceConfiguration), dan kijkt hij na in de database na of de configuratie informatie van het desbetreffende apparaat voorkomt. Als het wel voorkomt, stuurt hij een positief antwoord terug, anders een negatief antwoord.

Als de server van het cliënt systeem een “apparaat configuratie informatie bericht” krijgt (SetConfiguration), dan kijkt hij na in de database na of de configuratie informatie van het desbetreffende apparaat voorkomt. Als het wel voorkomt, stuurt hij een positief antwoord terug, anders een negatief antwoord.

Als de server van het cliënt systeem een aanvraag krijgt voor een lijst van “datarecord”(GetDataRecord), dan stelt de server een lijst van datarecord op en stuurt hij deze lijst terug naar de cliënt. Indien hij geen waardes heeft, dan stuurt hij een lege lijst terug.

Als de server van het cliënt systeem een lijst van datarecord krijgt (SetDataRecord), dan gaat de server deze lijst van datarecord wegschrijven naar de database. Indien hij goed weggeschreven heeft, stuurt hij een positief antwoord terug, anders wordt het een negatief antwoord bericht gestuurd.

L Testrapport

Testrapport

Data wegschrijven naar klimaatcomputers

Bedrijf: LetsGrow.com

Versie: 1.0

Datum: 27-05-2015

Plaats: Vlaardingen

Opdrachtgever:

Peter Hendriks

Bedrijfsmentoren:

Leon Batta/Edwin Vis

Documenthistorie:

Datum	Versie	Beschrijving
23-5-2016	0.5	Concept
28-5-2016	1.0	Definitief

Distributie:

Naam	Versie Concept	Versie Definitief
Joost Simons	X	X
Edwin Vis	X	X

Akkoord opdrachtgever en projectleider.

Joost Simons	Eray Soy
Handtekening:	Handtekening:

Inleiding

Dit testplan is samengesteld om het prototype te laten testen door de officiële tester van LetsGrow.com. Wegens de beperkte kennis van de tester op het gebied van code, is er gekozen om voor hem een black box methode te hanteren. Deze test wordt uitgevoerd aan de hand van de eisen die opgesteld zijn.

Doel van het testplan.

Het doel van dit testrapport is om het prototype te kunnen testen op de functionaliteiten die te vinden is in het definitierapport in hoofdstuk 6. De opdrachtgever wil uiteindelijk zien of het voldoet aan de vooraf opgestelde eisen en wensen.

De testmethode.

In dit rapport gaat er gebruik gemaakt worden van de Black Box testmethode.

Testomgeving.

De testen die in dit rapport beschreven zijn worden uitgevoerd gebruikmakend van verschillende zaken. Een computer met Windows 10 erop, een Hoogendoorn Klimaatcomputer, een multimeter met een 4000mV meetbereik en het gehele prototype met de samengestelde applicatie. Het prototype is aangesloten aan de Hoogendoorn klimaatcomputer. Deze is te vinden in de testruimte in de werkplaats op de begane grond van het Hoogendoorn Gebouw.

De handelingen die vooraf het testen verricht moeten worden:

- Itemcodes zijn geconfigureerd door middel van LetsGrow.com website door de afstudeerder.
- Prototype is goed aangesloten aan de klimaatcomputer en er is ook een actieve internetverbinding.
- Daarnaast is er ook een 'ethernet' kabel gekoppeld aan het prototype.
- LetsGrow.com Web API server applicatie is in de testomgeving gezet door de bedrijfsmentor.
- Service monteur heeft het prototype volgens de handleiding gereedgemaakt, behalve de stroomtoevoer. Voeding kabel 'mag' niet gekoppeld worden aan de Pi, dit gaat de tester zelf doen.
- SQL Server 2014 opgestart en de tValue, tDeviceInfo, tDevice en tabellen zijn geopend om het dataverkeer te kunnen zien.
- Voor het testen zijn er in de bovenstaande tabellen een 'timestamp' ingebouwd om binnenkomend verkeer te kunnen detecteren op een bepaald tijdstip.

Aannamen:

- Prototype werkt naar behoren.
- De bedradingen zijn correct aangesloten.
- Hoogendoorn klimaatcomputer werkt naar behoren.
- LetsGrow.com Web API server is online en werkt naar behoren.

Testgevallen:

Automatische opstarten van prototype, authenticiseren van het prototype en zelf het verkeer initiëren

De volgende eisen worden met dit testgeval getest:

Het prototype moet automatisch opstarten, nadat het prototype voorzien is van stroom.

Het prototype moet door middel van een 'uniek identificatienummer' bij de servers authenticiseren

Het prototype moet zelf het verkeer initiëren met de servers om firewall problemen te kunnen voorkomen.

Pre condities	Test-scenario	Beschrijving	Uitvoer-voorspelling	Werkelijke uitvoer	Opmerking
	Het prototype moet automatisch opstarten, nadat het prototype voorzien is van stroom.	De tester gaat een voeding(stroomtoevoer) koppelen aan het prototype.	Het prototype start automatisch op en toont aan de hand van zijn (groen) led aan dat het opgestart is. Groen led moet voor een korte periode aanstaan.	Het prototype start automatisch op en toont aan de hand van zijn (groen) led aan dat het opgestart is. Groen led moet voor een korte periode aanstaan.	Het kan 2 a 3 minuten in beslag nemen voordat het prototype is opgestart.
<ul style="list-style-type: none">Het prototype is opgestart.LetsGrow.com Web API server applicatie is opgestart.SQL Server Management Studio 2014 opgestart en de tDevice tabel is geopend.	Het prototype moet het verkeer zelf initiëren en gaat eerst zijn MAC-adres opsturen naar de servers om te authenticiseren.	<p>Het prototype stuurt zijn MAC-adres naar de servers.</p> <p>De tester gaat aan de hand van tDevice tabel (timestamp kolom) kijken of er afgelopen 5 minuten een MAC adres verificatie bericht is binnengekomen.</p>	Het prototype heeft zijn MAC-adres succesvol gestuurd naar de servers en succesvol geauthentiseerd. In de LetsGrow.com database server tabel (tDevice) is de datum te zien wanneer er validatie binnengekomen is.	Het prototype heeft zijn MAC-adres succesvol gestuurd naar de servers en succesvol geauthentiseerd. In de LetsGrow.com database server tabel (tDevice) is de datum te zien wanneer er validatie binnengekomen is.	Tester kan aan de hand van de tDevice tabel zien of er dergelijke MAC-adres aanvraag binnen is gekomen. Indien er binnen 5 minuten helemaal geen verkeer is binnengekomen, heeft het prototype het verkeer niet kunnen initiëren.

60 minuten in 'Doe Niets' mode bij een niet actieve internetverbinding

De volgende eisen worden met dit testgeval getest:

Het prototype moet indien er geen actieve netwerkverbinding aanwezig is een minimale tijd van 60 minuten niets doen.

Pre condities	Test-scenario	Beschrijving	Uitvoer-voorspelling	Werkelijke uitvoer	Opmerking
<ul style="list-style-type: none">Het prototype is opgestart.Internet kabel is niet aangesloten aan het prototypeLetsGrow.com Web API server applicatie is opgestart.SQL Server Management Studio 2014 opgestart en de tValue tabel is geopend.	Het prototype moet indien er geen actieve netwerkverbinding aanwezig is een minimale tijd van 60 minuten niets doen.	<p>Het prototype gaat proberen om te connectie op te bouwen met de LetsGrow.com server. Er is geen actieve netwerkverbinding mogelijk, dus laat het prototype de groene en rode leds branden en doet de komende 60 minuten niets.</p> <p>De tester ziet aan de hand van leds (groene en rode) en aan de tValue tabel dat het prototype in 'DoeNiets' stand is beland voor de komende 60 minuten.</p>	Het prototype laat de groene en rode leds branden en doet de komende 60 minuten lang niets.	<i>Het prototype laat de groene en rode leds branden en doet de komende 60 minuten lang niets.</i>	<p>Om weten te komen dat hij 60 minuten lang niets doet, kan er na 5 minuten de internetkabel weer gekoppeld worden aan de Pi 2 en aan de tValue tabel kan er gezien worden dat er geen waarden opgevraagd worden de komende 60 minuten. Daarnaast branden de rode en groene ledje de komende 60 minuten lang.</p> <p><i>Notitie: Soms gaan de leds binnen deze 60 minuten voor 1-2 seconden uit en dan weer aan!</i></p>

Zelfstandig/automatisch zijn werkingen uit gaan voeren en om de 5 minuten gegevens ophalen.

De volgende eisen worden met dit testgeval getest:

*Het prototype moet zelfstandig/automatisch zijn werking(en) uit gaan voeren, zonder de tussenkomst van een persoon.
Het prototype moet om de 5 minuten met de LetsGrow.com servers connectie maken om gegevens te kunnen ophalen.
Het prototype moet aan de hand van zijn "leds" een indicatie kunnen tonen van zijn interne status*

Pre condities	Test-scenario	Beschrijving	Uitvoer-voorspelling	Werkelijke uitvoer	Opmerking
<ul style="list-style-type: none"> Het prototype is opgestart. LetsGrow.com Web API server applicatie is opgestart. SQL Server Management Studio 2014 opgestart en de tValue en tDeviceConfigInfo tabellen zijn geopend. 	Het prototype moet zelfstandig/automatisch zijn werking(en) uit gaan voeren, zonder de tussenkomst van een persoon.	De tester gaat de eerste 30 minuten lang kijken of het prototype zijn configuratie verzonden heeft (door middel van de SQL tDeviceConfigInfo tabel) en gegevens opgehaald heeft aan de hand van de tValue tabel.	Het prototype heeft zijn configuratie succesvol opgestuurd naar de servers. Het prototype heeft succesvol om de 5 minuten gegevens van de servers opgehaald	Het prototype heeft zijn configuratie succesvol opgestuurd naar de servers. Het prototype heeft succesvol om de 5 minuten gegevens van de servers opgehaald	Aan de hand van tValue tabel (timestamp kolom) kan er gezien worden wanneer het prototype waardes heeft opgehaald.
<ul style="list-style-type: none"> Het prototype is opgestart. LetsGrow.com Web API server applicatie is opgestart. SQL Server Management Studio 2014 opgestart en de 	Het prototype moet om de 5 minuten met de LetsGrow.com servers connectie maken om gegevens te kunnen ophalen en Het prototype moet aan de hand van zijn "leds" een indicatie kunnen tonen van zijn interne status	De tester gaat 1 uur lang de gehele klimaatcomputer pc in de gaten houden om te kijken of er om de 5 minuten waardes zijn binnengekomen van het prototype. Het prototype gaat om de 5 minuten met de	Het prototype heeft met succes in 5 minuten tijd de analoge waardes opgehaald van uit de servers. Deze procedure is om de 5 minuten met succes uitgevoerd. Het prototype laat om de 5	Het prototype heeft met succes in 5 minuten tijd de analoge waardes opgehaald van uit de servers. Deze procedure is om de 5 minuten met succes uitgevoerd. Het prototype laat om de	

tValue tabel is geopend.		servers connectie maken om gegevens op te halen.	minuten groene ledje branden.	5 minuten groene ledje branden.	
--------------------------	--	--	-------------------------------	---------------------------------	--

Configuratie informatie opsturen naar de servers, indien deze niet aanwezig is.

De volgende eisen worden met dit testgeval getest:

Het prototype moet zijn configuratie informatie sturen naar de servers, indien deze niet aanwezig is

Pre condities	Test-scenario	Beschrijving	Uitvoer-voorspelling	Werkelijke uitvoer	Opmerking
<ul style="list-style-type: none"> Het prototype is opgestart. LetsGrow.com Web API server applicatie is opgestart. SQL Server Management Studio 2014 opgestart en de tDeviceConfigInfo tabel is geopend. Tester heeft voor het desbetreffende apparaat de configuratie informatie verwijderd. 	Het prototype moet zijn configuratie informatie sturen naar de servers, indien deze niet aanwezig is	<p>Het prototype voert om de 5 minuten een configuratie check uit met de servers. Indien er geen configuratie aanwezig is in de tDeviceConfigInfo tabel, dan stuurt het prototype zijn configuratie informatie naar de servers.</p> <p>De tester gaat aan de hand van tDeviceConfigInfo tabel zien of er daadwerkelijk de verwijderde configuratie informatie waardes weer na 6 minuten in de tabel zijn teruggekomen.</p>	Het prototype heeft succesvol gekeken of er geen configuratie informatie aanwezig was. Daarna heeft het prototype zijn configuratie informatie gegevens gestuurd naar de servers.	Het prototype heeft succesvol gekeken of er geen configuratie informatie aanwezig was. Daarna heeft het prototype zijn configuratie informatie gegevens gestuurd naar de servers.	Dit testgeval kan op de volgende manier uitgevoerd worden. Eerst DeviceConfigInfo het desbetreffende apparaat configuratie informatie verwijderen. Na ongeveer 5-6 minuten kijken of er een configuratie informatie van het desbetreffende MAC adres weer aanwezig is. Indien dit niet het geval is, kan er door middel van (timestamp) gekeken worden of er dergelijke configuratie check uitgevoerd is binnen de afgelopen 5 minuten.

Minstens een analoge waarde kunnen sturen naar de klimaatcomputer.

De volgende eisen worden met dit testgeval getest:

Het prototype moet minstens een analoge waarde kunnen sturen naar een klimaatcomputer

Pre condities	Test-scenario	Beschrijving	Uitvoer-voorspelling	Werkelijke uitvoer	Opmerking
<ul style="list-style-type: none"> Het prototype is opgestart. LetsGrow.com Web API server applicatie is opgestart. SQL Server 2014 opgestart en de benodigde tabellen zijn geopend. Tester heeft de Hoogendoorn klimaatcomputer voor zich met de Hoogendoorn software. De sensor-metingen tabblad is geopend en zichtbaar voor de tester. 	Het prototype moet minstens een analoge waarde kunnen sturen naar een klimaatcomputer	<p>Het prototype gaat een analoge waarde naar de klimaatcomputer sturen. Het prototype gaat het groene ledje laten branden.</p> <p>De tester ziet aan de hand van Hoogendoorn Klimaatcomputer (tabblad sensor-metingen) dat er een analoge waarde het systeem binnen is gekomen.</p>	Het prototype heeft succesvol een analoge waarde gestuurd naar de klimaatcomputer. Het prototype heeft het groene ledje laten branden.	<p>Het prototype heeft succesvol een analoge waarde gestuurd naar de klimaatcomputer. Het prototype heeft het groene ledje laten branden.</p>	<p>Dit testgeval kan ook op de volgende manier getest worden. Door middel van een multimeter kan er aan de 1^e en 2^e analoge uitgangen van het shield gemeten worden of er ook daadwerkelijk de desbetreffende spanning waar te nemen is.</p> <p>Bovenstaande manier ook uitgeprobeerd, lukt ook goed.</p>

Loggen van onbekende uniek identificatienummer in de logging database.

De volgende eisen worden met dit testgeval getest:

De LetsGrow Web API server moet een onbekende unieke identificatienummer loggen in zijn logging database.

Pre condities	Test-scenario	Beschrijving	Uitvoer-voorspelling	Werkelijke uitvoer	Opmerking
<ul style="list-style-type: none">Het prototype is opgestart.LetsGrow.com Web API server applicatie is opgestart.SQL Server 2014 opgestart en de tDevice tabel is geopend.Tester heeft de Logging database omgeving geopend en kan zien welke logs het afgelopen uur heeft plaatsgevonden.	De LetsGrow Web API server moet een onbekende MAC-adres loggen in zijn logging database.	<p>De tester gaat eerst de desbetreffende MAC-adres verwijderen van de tDevice tabel.</p> <p>Daarna ziet de tester aan de hand van de logging database of er een MAC-adres gelogd is binnen de 5 minuten.</p>	De LetsGrow Web API server heeft succesvol het onbekende MAC-adres gelogd in zijn logging database.	De LetsGrow Web API server heeft succesvol het onbekende MAC-adres gelogd in zijn logging database.	Logging werkt goed. Goed geïmplementeerd!

24 uur duurtest.

De volgende eisen worden met dit testgeval getest:

Het prototype moet 24 uur per dag draaien.

Pre condities	Test-scenario	Beschrijving	Uitvoer-voorspelling	Werkelijke uitvoer	Opmerking
<ul style="list-style-type: none"> Het prototype is opgestart. LetsGrow.com Web API server applicatie is opgestart. SQL Server 2014 opgestart en de tDevice tabel is geopend. Tester is ingelogd bij LetsGrow.com en heeft de klimaatcomputer grafiek voor zich. Tester heeft de Hoogendoorn klimaatcomputer voor zich met de Hoogendoorn software. 	Het prototype draait 24 uur per dag zonder fouten.	<p>De tester gaat na 24 uur de historie van de klimaatcomputer bekijken. Hier kan hij zien dat er het afgelopen 24 uur waardes zijn weggeschreven.</p> <p>De tester gaat na 24 uur op de LetsGrow.com website de klimaatcomputer grafiek bekijken.</p>	<p>Het prototype heeft het afgelopen 24 uur succesvol en zonder fouten gedraaid.</p> <p>De tester ziet op de LetsGrow.com website aan de klimaatcomputer grafiek dat er de afgelopen 24 uur desbetreffende waardes zijn binnengekomen in de klimaatcomputer.</p>	<p>Het prototype heeft het afgelopen 24 uur succesvol en zonder fouten gedraaid.</p> <p>De tester ziet op de LetsGrow.com website aan de klimaatcomputer grafiek dat er de afgelopen 24 uur desbetreffende waardes zijn binnengekomen in de klimaatcomputer.</p>	Voor het eerste test draait het prototype zeer stabiel. Complimenten!

M Gesprek Opdrachtgever

Afstudeerproject: Data wegschrijven naar de klimaatcomputers

Notulen vergadering 12-2-2016 11:00

Type vergadering: Projectopdracht verhelderen.

Voorzitter: Eray Soy

Aanwezigen: Eray Soy, Peter Hendriks

I. Opening vergadering

Vergadering geopend om 11:00 uur

II. Vaststelling presentielijst

Geen.

III. Goedkeuring van notulen van vorige vergadering

Geen.

IV. Openstaande punten

- *Projectopdracht verduidelijken.*
 - Wat is het probleem: LetsGrow heeft geen mogelijkheid om de gegevens direct naar de klimaatcomputers te kunnen sturen zonder tussenkomst van de klimaatcomputer leveranciers. LetsGrow kan alleen naar Hoogendoorn klimaatcomputers sturen!
 - Wat is de aanleiding van het probleem: Er is geen functionaliteit hiervoor beschikbaar.
 - Wie heeft het probleem: LetsGrow.com/Klanten
 - Waarom is het een probleem: Nu te veel afhankelijk van Hoogendoorn klimaatcomputer.
 - Welke motivaties voor het probleem op te kunnen lossen: Met de tijd meegaan, klanten extra en aanvullende dienst bieden. Minder afhankelijkheid.
 - Waar doet het probleem zich voor: Bij de klanten in zijn kas. Bij de klimaatcomputers. Bij Letsgrow.com
 - Peter Hendriks wil weten of het mogelijk is om door middel van het prototype(pi2 en shield) de klimaatcomputers te kunnen beïnvloeden?
 - Indien mogelijk, dan wil hij een 'werkende' prototype zien!
- **Kosten/Baten**
 - Prototype mag niet meer dan 300 euro kosten, inclusief alle hardware + software licentiekosten.
 - Peter denkt dat dit prototype ervoor kan zorgen dat LetsGrow.com meer in de persberichten kan komen. Dit zorgt dan ook weer voor dat er meer toekomstige klanten interesse kunnen hebben/tonen voor LetsGrow.com diensten.

-
V. WVTTK

Geen punten

VI. Volgende vergadering

Niet ingepland, indien nodig via mail inplannen.

VII. Sluiting vergadering

De vergadering is om 12:30 uur gesloten.

N Gesprek Christiaan Posthumus Meyes

Datum: 10-3-2016

Gesproken met: Christiaan Posthumus Meyes

Tijd: 13.00-15.00

Emailadres: cpm@letsgrow.com

Onderwerp: Wat is de huidige situatie en wat zijn klimaatcomputers regelingen?

Huidige situatie:

LetsGrow.com stuurt nu zijn data door middel van internet naar Hoogendoorn klimaatcomputers. Een klimaatcomputer is een computer die binnen een tuinbouwbedrijf verantwoordelijk is voor de aansturing en controleren van allerlei processen zoals het klimaat(Temperatuur, Co2 enz.). De Hoogendoorn computer zorgt ervoor dat de klant zijn klimaat in zijn kas (watergift, warmtekrachtkoppeling (WKK's) ed.) kan controleren en indien nodig kan sturen. De configuratie die de klant instelt, wordt doorgestuurd naar het regelstation. Dit gebeurt door de bedieningscomputer die alle data door middel van een ethernet doorstuurt naar het regelstation.

De functie van het regelstation is het uitvoeren van de regelingen. Om te kunnen regelen zijn er metingen en sturingen nodig. Hiervoor zorgt het I/O-systeem.(om meer te weten komen kan er interne document van Anthon van Linden (EtherCan) doorgenomen worden!).

Metingen komen via het I/O-systeem bij het regelstation binnen, het regelstation bepaalt de actie aan de hand van de meting en het setpoint. De actie wordt in een sturing omgezet die via het I/O-systeem naar het regelorgaan gaat. Bijvoorbeeld een motor om de ramen open te sturen.

Deze modules worden op de Hoogendoorn computer zelf geconfigureerd. Dit houdt in dat wanneer een input of een output nodig is een servicemonteur de desbetreffende module aan de CAN kabel gekoppeld wordt. Dit verschilt dus per gekoppelde sensor.

Hoogendoorn klimaatcomputer specificaties zijn 100% procent bekend, en alle anderen geven geen openheid van zaken(logisch is concurrent) en hierdoor kan alleen LetsGrow.com een 'setpoint' (dit wordt oftewel een sturing waarde genoemd) sturen.

Wat is klimaatcomputer en hoe ziet hij eruit:

Bestaat uit 2 onderdelen, een normale pc (als bedieningscomputer) en het regelstation(embedded pc) met speciale behuizing (in eerste verdieping hebben ze eentje staan!). Nu zit er Windows 7 en embedded versie van Windows 7 erop. Om de regelingen te kunnen uitvoeren is het regelstation benodigd. Deze voert de regelingen aan de hand van gegevens verkregen van bedieningscomputer uit. Deze zijn aan elkaar verbonden met een fysiek internet(ethernet) kabel. Op de pc draait natuurlijk een Hoogendoorn software. Gegevens worden van deze pc verstuurd naar het regelstation. Het regelstation regelt de regelingen en indien er nog sensoren gestuurd moeten worden, dan sturen ze deze weer door naar de ingangen en uitgangen modules (remote input/output modules)

Wat is een regeling:

Een regeling is niets anders een model van berekeningen. Om te kunnen regelen zijn er metingen (meting waardes) en sturingen (sturing waardes) nodig. Hiervoor zorgt het Input/Output-systeem (Ingang/Uitgang-systeem). Het I/O-systeem bestaat uit een EtherCan welke de gateway is tussen ethernet en CAN-bus, vervolgens zijn er 'Remote Input/Output modules' die decentraal opgesteld worden en aan de CAN-bus hangen. Belangrijk om te weten is dat er aan de 'remote I/O modules' de sensoren en actuatoren verbonden worden. Waardes kunnen dus vanuit deze sensoren het regelstation binnenkomen en dus kan er van de andere kant een communicatie opgebouwd worden.

Hoe kan een regeling beïnvloedt worden?

Als de sturing en meting waardes beïnvloed kunnen worden, dan kan de regeling ook beïnvloed worden! Dus data van klimaatcomputer en data vanuit de remote input/output modules. Hoe het precies gaat kan in vierde verdieping aan Jack Schipper(die bezig houdt met de hardware) gevraagd worden, omdat hij het opgebouwd heeft. Daarnaast is er ook een documentatie (wel beveiligd en intern bedoeld) geraadpleegd worden. (iets in de richting values various sensor pdf; samengesteld door Jack Schipper).

O Gesprek Edwin Vis

Datum: 11-3-2016

Gesproken met: Edwin Vis over de communicatie

Tijd: 9.00-9.45

Emailadres: ev@letsgrow.com

Onderwerp: communicatie LetsGrow en remote apparaten(hardware)

Hoe communiceert Letsgrow.com op dit moment met zijn hardware componenten.

Vroeger gebeurde het door middel van de SOAP protocol. Er is na 2010 overgegaan om een API te bouwen (met het REST protocol) en deze te gebruiken voor de toekomstige projecten. Deze kan via api.letsgrow.com opgevraagd worden(pdf document) hoe geïmplementeerd kan worden. Dus om met het toekomstige prototype te kunnen communiceren is het gewenst om gebruik te maken van deze Web API. (andere methodes(zoals SOAP) worden met de tijd mee langzaam uit gefaseerd. Om deze Web API te kunnen uitbreiden is er wel kennis vereist van het ASP.NET Web API (van Microsoft!).

Notitie: Indien er meer vragen zijn kan er mondeling met Edwin gesproken worden.

P Interview Jack Schipper

Interview Jack Schipper Werking klimaatcomputers

Datum: 17-maart-2016: 2:30-3:30

Wat is uw afdeling:

Mijn afdeling is Research & Development binnen Hoogendoorn.

Wat is uw functie?

Mijn functie is het in elkaar zetten van de hardware die gebruikt gaat worden in de huidige maar ook de toekomstige klimaatcomputer.

Wat doen de klimaatcomputers?

Haha, Christiaan (je collega bij LetsGrow.com) weet er ook redelijk van, maar ik zal het je uitleggen. Klimaatcomputer is niets anders dan een normale PC met onze eigen software erop. Deze computers zorgen ervoor dat onze klanten al hun gehele klimaat binnen in hun kas kunnen regelen. Klimaatcomputer bestaat uit een bedieningscomputer en een Embedded computer (ook wel het regelstation genoemd). Deze zaken zijn met elkaar verbonden, door middel van een internet kabel. Constant krijgt klimaatcomputer gegevens door middel ons 'eigen' protocol die geïmplementeerd is in onze software.

Wat is een regeling?

Een regeling is een model voor het berekenen van een 'nieuwe' sturing waarde door middel van een sturing en een meting waarde. Als je een sturing waarde hebt (volgens mij kan LetsGrow.com naar onze klimaatcomputers zulke waardes sturen), moet je ook een meting waarde hebben om een 'nieuwe' sturing waarde te kunnen berekenen. Je gaat zeker vragen hoe je deze kan beïnvloeden, lachte hij. (Dit wist hij namelijk, omdat ik mijn project eerder aan hem had verteld.)

Om een regeling te kunnen beïnvloeden moet je een bepaalde spanning in mA of V hebben. Dit kan je opzoeken in de interne PDF bestand die ik had samengesteld. Alleen is dit PDF document geheim, dus we willen dat je 'zorgvuldig' mee om gaat.

Hoe kan een regeling beïnvloedt worden?

Maar terugkomend op de vraag hoe kan een regeling beïnvloedt worden. Dit kan door middel van een analoge waarde te kunnen sturen naar het regelstation. Dit kan je doen door bijvoorbeeld een sensor van (ik zeg maar wat nu) 10 V te hangen. Deze sensor gaat dan een bepaalde voltage produceren. Deze voltage komt door middel van een 'remote input/output module' (een soort Bus terminal) bij het regelstation terecht. Hier is ook een 'sturing waarde' ingekomen van de bedieningscomputer. Aan de hand van deze 2 waardes kan dus de regeling beïnvloedt worden. Ook is er bij PRIVA mogelijk om vanuit deze "remote input/output modules" de regelingen te kunnen beïnvloeden. Het werkt op een soortgelijke manier als Hoogendoorn klimaatcomputer. Indien het mogelijk is om naar een klimaatcomputer een voltage te kunnen sturen, dan moet je van uit gaan dat je ook de regelingen kan beïnvloeden. Dit zal niet een makkelijke opdracht zijn. Indien je het voor elkaar kan krijgen met een Pi, dan zal dat een heel grote stap kunnen zijn. Maar dit is meer business marketing zaak.

Q Gesprek Bedrijfsmentoren

Afstudeerproject: Data wegschrijven naar de klimaatcomputers

Notulen vergadering 6-4-2016 10:00

Type vergadering: Eisen/Wensen Project afstuderen.

Voorzitter: Eray Soy

Aanwezigen: Eray Soy, Edwin Vis, Leon Batta

I. Opening vergadering

Vergadering geopend om 10:00 uur

II. Vaststelling presentielijst

Geen.

III. Goedkeuring van notulen van vorige vergadering

Geen.

IV. Openstaande punten

- *Eisen aan het prototype.*
 - Edwin Vis en Leon Batta willen dat het prototype de volgende eisen bevat:
 - Prototype moet nadat hij stroom heeft, automatisch opstarten.
 - Prototype moet van binnenuit connectie maken met de LetsGrow.com servers, zodat ze de firewall problemen willen voorkomen. Verleden hadden ze hiermee problemen.
 - Authenticatie moet plaatsvinden aan de hand van een 'uniek code'. Deze mag de afstudeerder zelf bepalen.
 - Prototype moet periodiek "stil" zijn besturingssysteem updaten, zodat het goed beveiligd is en altijd voorzien moet zijn van de nieuwe beveiligingsupdates.
 - Om de 5 minuten moet het prototype connectie maken met de LetsGrow.com servers om gegevens op te kunnen vragen.
 - Prototype moet minstens één analoge output hebben. Liefst willen ze 2 analoge output zien.
 - Prototype moet 24 uur kunnen draaien, zonder de tussenkomst van de klanten/monteurs.
 - Prototype moet min mogelijk soldeerwerk bevatten. Zo min mogelijk houdt in dat er alleen headers/pinnetjes gesoldeerd kunnen worden. Er mag wel doormiddel van een USB kabeltje verbonden worden aan de PI, maar er mag totaal niet met weerstanden/kleine onderdelen gesoldeerd worden(ook geen Breadbordjes!).
 - Hardware van prototype moet binnen 5 dagen leverbaar zijn, zodat ze het gehele prototype inclusief software binnen 10 dagen kunnen leveren aan de klanten.

- Kosten van een shield willen ze niet meer dan 100 euro hebben. Graag eronder blijven, omdat anders de verhoudingen scheef worden. Shield kost dan veel meer vergeleken met het moederbord hardware (PI 2)
 - Status door middel van leds die aanwezig zijn op de PI.
 - **Indien tijd over dan:** Behuizing van het prototype bouwen. (waterdicht!)
 - **Nice to have:** Stroomstoring check door middel van ESD en Leds die een indicatie kunnen tonen van de interne status van het prototype.
- Software eisen
 - Het is een 'zware' voorkeur om met Visual Studio te ontwikkelen, omdat Leon als Edwin daarmee werken.
 - Ontwikkeltaal voorkeur C#, omdat ze zelf ook voor alle LetsGrow applicaties de ontwikkeltaal C# gebruiken. Uit de analyse bleek dat het wel kon, dus ook gebruiken!
 - Als besturingssysteem voor het prototype willen ze Windows IOT gebruiken met Visual Studio en C#. Uit de analyse is het gebleken dat het mogelijk is, dus ook dit besturingssysteem gebruiken!
 - Licentiekosten moeten gratis zijn voor het prototype. Ze willen geen "extra" licentiekosten betalen aan bepaalde software (periodiek). Eenmalig kan het wel, maar zo veel mogelijk gratis/demo software gebruiken.
 - Software moet de kant van embedded opgaan. Oftewel niet echt een UI aan de software. Er mag in de achtergrond een applicatie draaien. Dit mag een ook een.dll bestand zijn. Hoeft niet .exe bestand als het maar draait.
 - Communicatie liefst door middel van LetsGrow API plaatsvinden. Om dit te kunnen realiseren moet er in de software(cliënt) API geïmplementeerd worden. Tevens moet er aan de LetsGrow kant(server) ook bepaalde functies geïmplementeerd worden aan de bestaande API.
 - Aan de kant van de server een logging inbouwen voor errors/onbekende zaken

V. WVTTK

Geen punten

VI. Volgende vergadering

Niet ingepland, indien nodig via mail inplannen of mondeling overleggen.

VII. Sluiting vergadering

De vergadering is om 11:30 uur gesloten.

R Handleiding Prototype

Datum : Mei, 2016

Onderwerp : Handleiding voor het Prototype

Versie 1.0: 2016 mei

- eerste uitgave

Introductie

Deze documentatie beschrijft hoe het prototype kan worden aangesloten op een Hoogendoorn klimaatcomputer en in werking kan worden gesteld. De handleiding kan ook worden gebruikt voor het aansluiten van het prototype op een ander soort klimaatcomputer.

Benodigdheden:

- *UTP kabel*
- *Schroevendraaier*
- *Micro USB Power Supply*
- *Het prototype met behuizing.*

Locatie van plaatsing:

- *In de buurt van de IO klimaatcomputer, niet in volle zon.*

Randvoorwaarden:

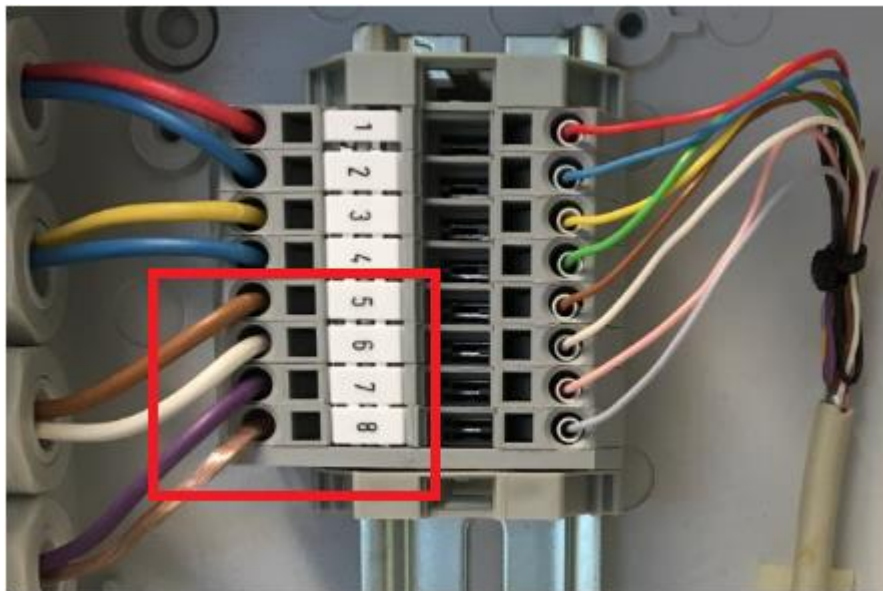
- *Fysieke internetaansluiting.*
- *Stopcontact*

Het Prototype gereedmaken voor gebruik:

Het verkregen prototype bevindt zich in een speciale behuizing. Om dit prototype aan een klimaatcomputer te kunnen koppelen dienen de volgende stappen gevolgd te worden.

1: Schroef de voorkant van de behuizing open.

2: Pak het uiteinde van de bedrading met nummer 5-6 en nummer 7-8 vast. Nummer 5 en nummer 7 zijn de 'plus-kant' en nummer 6 en 8 zijn de 'grond-kant'.

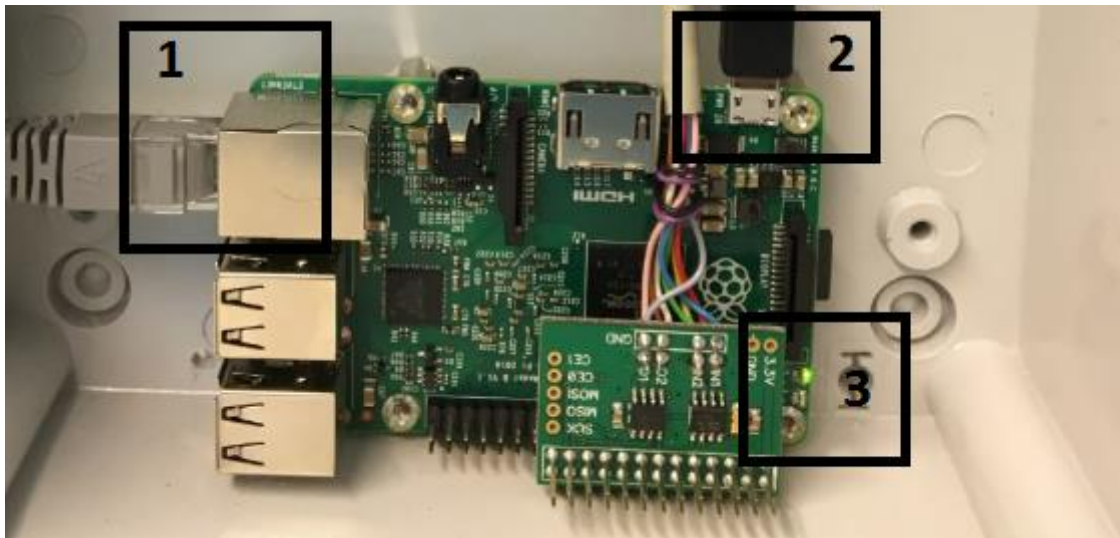


Figuur 1: Prototype bedrading

3: Koppel deze bedrading aan de 'remote input/output module' van 0-2V poorten. Nummer 5 aan de 'plus-kant' en nummer 6 aan de 'grond-kant'. Dit geldt ook voor de nummers 7 en 8.



Figuur 2: 'remote input/output module'



Figuur 3: Raspberry Pi 2 + ADC-DAC Pi Shield

4: Koppel eerst een 'ethernet-kabel' aan de 'ethernet-poort (nummer 1) van de Raspberry Pi 2.

5: Koppel daarna de 'Micro USB Power Supply' aan de USB Micro ingang (nummer 2) van de Raspberry Pi 2.

6: Wacht 4 a 5 minuten af om te kijken of het prototype goed is opgestart. Dit kan afgelezen worden aan het ledje (nummer 3) van de Raspberry Pi 2. Als deze groen wordt is het prototype goed opgestart. Indien beide leds branden (zowel groen als rood) dan betekent dat dat er geen actieve internetverbinding aanwezig is. Indien alleen het rode ledje brand, dan betekent dat er tijdens het wegschrijven van een waarde naar de klimaatcomputer iets mis is gegaan. In het laatste geval dient er contact opgenomen te worden met de LetsGrow.com klantenservice.

7: Het prototype is nu gereedgemaakt voor zelfstandige werking. De voorkant van de behuizing kan nu dichtgeschroefd worden.

PAS OP, Indien het prototype niet naar behoren werkt, dan moet er contact opgenomen worden met LetsGrow.com. Zonder de toestemming van LetsGrow.com mag er aan dit prototype 'NIETS' gewijzigd worden!

Copyright © LetsGrow.com 2016

Reproduction or duplication of this product or any part thereof (incl. the software) is prohibited. We have taken utmost care in the development and creation of this product. Due to its nature (electronics, software, communication), no full guarantee can be granted however as to its continued correct function.

This type of product is inevitably susceptible to interferences from outside influences. If this product is being used in critical processes an external means of monitoring the process separate from this product will therefore always be required.

This certainly also includes personal control by humans at a frequency suited to the respective process. The delivery agreement for this product is subject to the general terms of delivery, ALIB 2007, of UNETO-VNI, explicitly excluding any possible general terms of the purchaser/user or reseller. With your acceptance via the button below, you declare that you have read these ALIB 2007 terms; that you have noted and accepted these; and that you are also authorized to do so on behalf of the purchaser. The ALIB 2007 terms can be obtained from LetsGrow.com.