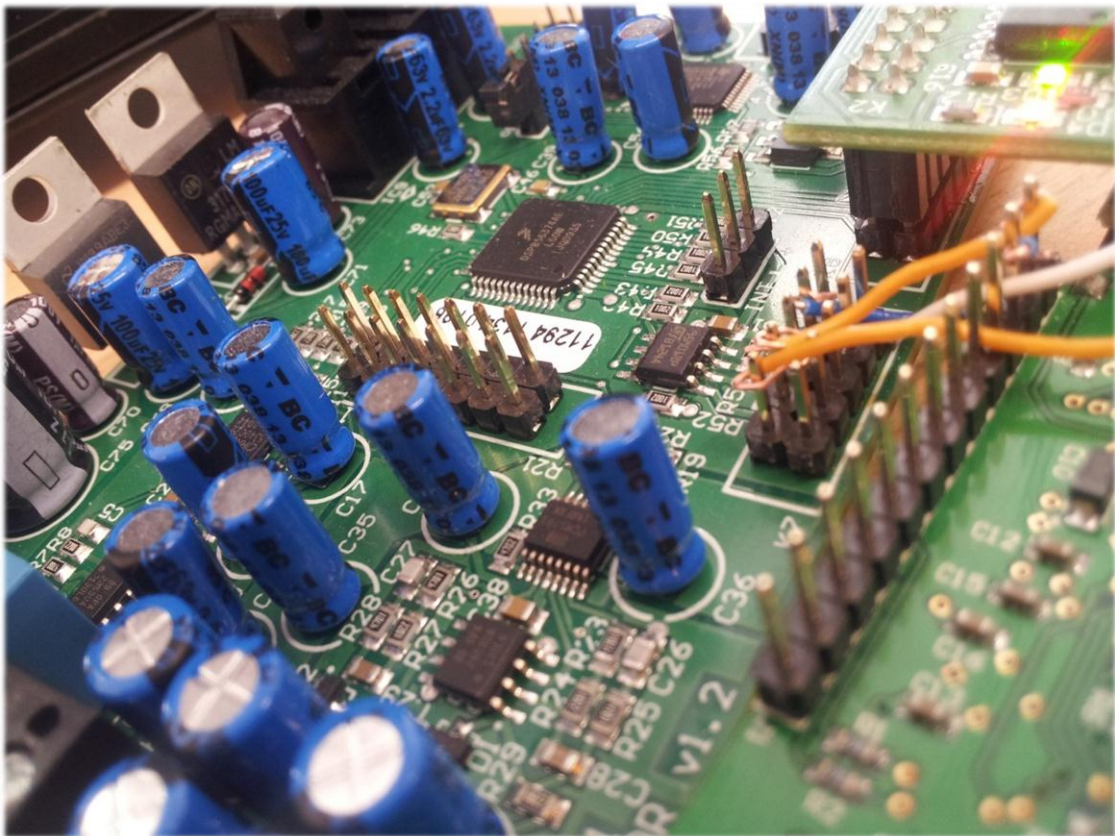


Scriptie

Onderzoeken hoe een DSP kan worden gebruikt om een zo realistisch mogelijk signaal te simuleren.



Roy de kok(1546848)
Eerste Examiner: Peter van Rooijen
Technische informatica

Versie: 1.2.0

VOORWOORD

Voor je ligt een scriptie welke het resultaat is van een onderzoek hoe een DSP kan worden gebruikt om een zo realistisch mogelijk signaal te simuleren. Met deze scriptie zal ik de opleiding Technische informatica aan de Hogeschool Utrecht afronden.

Ik wil nog een aantal mensen bedanken. In het bijzonder wil mijn beide stagebegeleiders, Bas Knopper en Peter van Rooijen, bedanken voor alle tijd die ze erin hebben gestoken om dit onderzoek zo succesvol mogelijk af te sluiten. Verder wil ik de mensen bedanken die mij hebben geholpen met alle uitdagingen die dit onderzoek met zich mee bracht. Zonder hun had ik sommige problemen waarschijnlijk nooit weten op te lossen. Als laatste wil ik nog de mensen bedanken die me hebben gesteund tijdens deze stage opdracht. Zeker op de momenten dat het wat meer tegen zat, heb ik hier veel steun bij kunnen vinden.

MANAGEMENTSAMENVATTING

De NCIM Groep ontdekte dat er nog geen realistische simulator was om radaroperators goed te kunnen opleiden. De NCIM-Groep wil het realisme ten opzichte van huidige simulatoren verhogen door radarsignalen te gaan simuleren in plaats van een vliegtuig op de juiste plek op een radarscherm te tekenen, zoals overige simulatoren het doen. Het realistisch simuleren van radarsignalen kost echter ook veel rekenkracht. De benodigde rekenkracht, zeker als er rekening gehouden moet worden met realistische elementen zoals ruis, is voor een normale computer te veel. Hierdoor zal er gebruik gemaakt moeten worden van een andere techniek of hardware om dit probleem af te vangen. Uiteindelijk is er gekozen om gebruik te maken van een Digitale signaalprocessor (DSP). De vraag die in scriptie beantwoord gaat worden is:

Hoe kan een DSP worden gebruikt om een zo realistische mogelijk signaal te simuleren, zonder dat de snelheid van de simulator te nadelig wordt beïnvloed?

Deze onderzoeksvraag gaat er van uit dat het doormiddel van een DSP mogelijk is om een radar signaal te verzenden, ontvangen en succesvol te verwerken in de simulator. Door een aantal onverwachte uitdagingen is het project achter op schema gaan lopen. Hierdoor zijn van de meeste DSP componenten alleen maar losse resultaten beschikbaar. De resultaten van de losse componenten voldoen echter aan de verwachte resultaten. Hierom wordt er dan ook vanuit gegaan dat een DSP gebruikt kan worden om een zo realistisch mogelijk signaal te simuleren. De resultaten laten verder ook zien dat de snelheid niet te nadelig wordt beïnvloed. De extra vertraging die de DSP met zich brengt is enkele milliseconde. Dit is voor het menselijke oog verwaarloosbaar. Ook voor het gebruik in de code zelf maakt die paar milliseconde niet uit.

Kortom, het is mogelijk om met behulp van een DSP een zo realistisch mogelijk signaal te simuleren, zonder dat de snelheid van de simulator te nadelig wordt beïnvloed.

INHOUDSOPGAVE

1	INLEIDING	1
1.1	BESCHRIJVING VAN HET BEDRIJF	1
1.2	AANLEIDING TOT DE OPDRACHT	1
1.2.1	<i>Werking Radar</i>	3
1.3	RELATIE MET DE ORGANISATIE	3
1.4	WAAR BEN IK TERECHT GEKOMEN	3
2	PROBLEEMSTELLING	5
3	UITGANGSPOSITIE.....	6
3.1	FPGA	6
3.2	RADAR SIGNALEN AANPASSEN OP BESCHIKBARE HARDWARE	6
4	ANALYSE VAN DE PROBLEMEN	7
4.1	VERSCHILLENDE HARDWARE MOGELIJKHEDEN BEKIJKEN	7
4.2	ONDERZOEK NAAR DE WERKING VAN EEN DSP.....	7
4.3	DE OPDRACHT.....	7
4.3.1	<i>DSP</i>	7
4.3.2	<i>Communicatie computer</i>	8
4.3.3	<i>Implementatie simulator</i>	8
4.4	PROJECT METHODE	9
5	DOELSTELLINGEN	10
5.1	ONDERZOEKSVRAGEN.....	11
5.2	OP TE LEVEREN PRODUCTEN.....	11
5.2.1	<i>Moscow</i>	11
6	ONTWERP	13
6.1	COMMUNICATIE MET DSP	13
6.1.1	<i>SPI</i>	13
6.1.2	<i>Communicatie via SPI</i>	14
7	PROTOTYPE.....	15
7.1	SCENARIO SETUP (1)	15
7.2	DE USER INTERFACE (3 & 4)	15
7.3	LUCIAD(5).....	15
7.4	LOGGER EN INSTALLER (7 & 8)	16
7.5	SIMULATOR (2)	16
7.6	SIGNAAL API (6)	16
7.6.1	<i>Signaal pad</i>	17
7.6.2	<i>Signal controller</i>	17
7.6.3	<i>PathNode</i>	18
7.7	DSP.....	19
7.7.1	<i>Verskillende soorten DSP code</i>	20
8	UITDAGINGEN.....	22
8.1	UITDAGING IN HET VERTRAGEN VAN DE TIJD	22
8.1.1	<i>Delay in hardware</i>	22
8.1.2	<i>Digitaal signaal</i>	23

8.1.3	Conclusie.....	23
8.2	SPI UITDAGINGEN.....	23
8.2.1	Zoeken naar oplossingen.....	24
8.3	BOOTLOADER.....	25
9	RESULTATEN	26
9.1	OPGELEVERDE PRODUCTEN.....	26
9.1.1	Implementatie met de simulator.....	26
9.1.2	DSP	27
9.2	WERKING MET DE SIMULATOR.....	30
9.2.1	Signaal sterkte.....	31
9.2.2	Simulator snelheid.....	31
9.2.3	Complete systeem	32
9.2.4	Benodigde DSP's.....	32
10	CONCLUSIES.....	33
10.1	HOE KAN EEN DSP WORDEN GEBRUIKT OM EEN ZO REALISTISCHE MOGELIJK SIGNAAL TE SIMULEREN, ZONDER DAT DE SNELHEID VAN DE SIMULATOR TE NADELIG WORDT BEÏNVLOED?	33
10.1.1	Is het signaal realistischer geworden?.....	33
10.1.2	Weersvariabelen	34
10.1.3	Is snelheid niet te nadelig beïnvloed?	34
10.2	IN HOEVERRE KOMT HET SIGNAAL IN DE BUURT VAN DE WERKELIJKHEID?.....	34
10.3	VOOR- EN NADELEN	34
10.3.1	Voordelen.....	34
10.3.2	Nadelen.....	34
10.4	IS HET MOGELIJK OM EEN DSP TE GEBRUIKEN BINNEN HET EWSIM PROJECT?.....	35
11	AANBEVELINGEN	36
12	EVALUATIE VAN DE PROCESGANG	37
12.1	HOE BEN IK BEGONNEN?	37
12.2	HOE ZAG HET VERVOLG ERUIT?	37
12.3	WAT HEB IK GELEERD?.....	37
12.4	WAT VOND IK LASTIG?.....	38
12.5	WAT VOND IK VAN HET EINDRESULTAAT?	38
12.6	WAT VOND IK VAN DE STAGE?	38
13	BRONVERMELDINGEN	39
BIJLAGE A	PLAN VAN AANPAK	A-1
BIJLAGE B	EVALUATIE VAN HET EIGEN FUNCTIONEREN	B-1
BIJLAGE C	HARDWARE INTERFACE	C-1
	RADARSEND (COMPUTER → DSP)	C-1
	OBJECT (COMPUTER → DSP)	C-2
	SPACE (COMPUTER → DSP).....	C-3
	RADAR RETURN(DSP → COMPUTER).....	C-4

BEGRIPPENLIJST

In deze begrippenlijst leg ik kort enkele begrippen uit die mogelijk niet bekend zijn bij de lezer van dit verslag. Belangrijke begrippen die voorkomen en mogelijk onduidelijk zijn:

- **DSP**: Een Digitale signaalprocessor(DSP) is een microprocessor die speciaal ontworpen is voor het bewerken van (continue) digitale signalen.(Digitale signaalprocessor, 2011)
- **FPGA**: Een field-programmable gate array(FPGA) is een geïntegreerde schakeling bestaande uit programmeerbare logische componenten. Deze logische componenten kunnen geprogrammeerd worden met behulp van logische functies zoals AND, XOR, enzovoorts. Deze functies kunnen bijvoorbeeld decoders of eenvoudige wiskundige functies zijn.(Field-programmable gate array, 2012)
- **Radar Cross Section**: De radar cross section (RCS) is een maat voor de mate waarin een obstakel, object of constructie de radiogolven reflecteert in dezelfde richting als waar zij vandaan komen. (TeTech)
- **Chaff**: Chaff is een antiradartechniek waarbij door militaire vliegtuigen of schepen grote hoeveelheden foliereepjes uitgeworpen worden, die gemaakt zijn van tin, aluminium of een kunststof met een opgedampte laag metaal. (Antiradarsnieuw, 2011)
- **Jamming**: Jamming is het sturen van stoorsignalen om zo het radarbeeld te vervormen en daarmee anderen (of vijanden) te misleiden.**EWSim**: EWSim is de afkorting voor het overkoepelende project van deze afstudeerstage genaamd Electronic Warfare simulator.

1 INLEIDING

In dit hoofdstuk wordt een uitleg gegeven over de context waarbinnen het afstudeerproject zal plaatsvinden. Als eerste komt er een beschrijving van de afdeling waar de opdracht wordt uitgevoerd. Hierin wordt uitgelegd waar het project wordt uitgevoerd en met welke omstandigheden. Hierna zal het gaan over de relatie van het afstudeerproject met andere projecten. Als laatste gaat het ook nog over de relatie van het afstudeerproject met de afdeling en de organisatie.

1.1 BESCHRIJVING VAN HET BEDRIJF

Mijn stage vindt plaats bij het bedrijf de NCIM-groep in Leidschendam. De NCIM-groep is een detacheringsbedrijf dat naast detachering ook innoveert. Deze innoverende projecten worden meestal uitgevoerd door afstudeerders en professionals die op dit moment geen detacheringsoopdracht hebben.



Figuur 1: Hoofdkantoor NCIM-Groep

De NCIM-groep richt zich voornamelijk op technisch complexe projecten in de volgende sectoren:

- Defensie en Veiligheid;
- Energie en Utilities;
- Telecom, Transport en Media;
- Technische Automatisering.

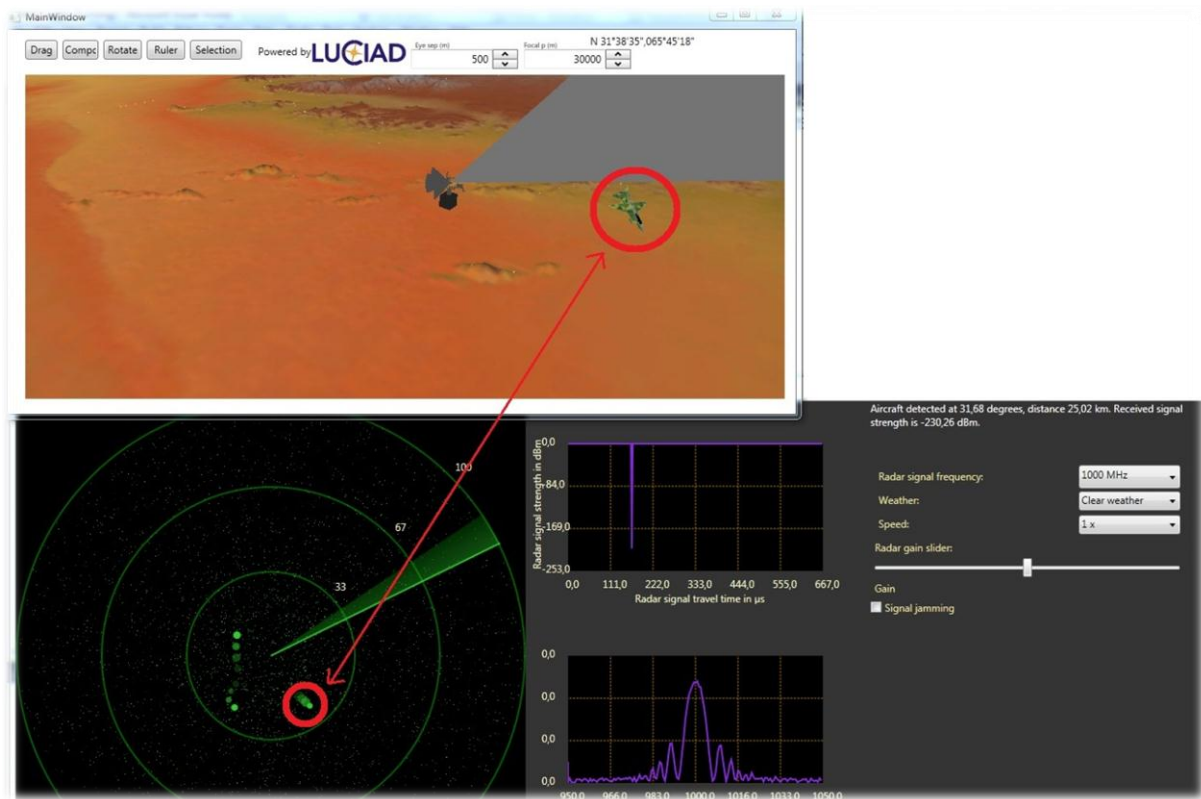
Het project waar ik mijn afstudeerstage zal uitvoeren behoort tot een innoverend project. Deze projecten worden meestal uitgevoerd op het hoofdkantoor in Leidschendam.

De duur van mijn afstudeerstage is drie en halve maand en loopt van 13 februari 2012 tot en met 29 mei 2012.

1.2 AANLEIDING TOT DE OPDRACHT

De NCIM Groep is momenteel bezig met het maken van een Electronic Warfare simulator. Deze simulator, te zien in Figuur 2, is bedoeld om een radarsysteem te simuleren. De simulator moet het mogelijk maken om radaroperators zo realistisch mogelijk te kunnen opleiden. Dit wordt bereikt doordat je realistische scenario's kan inladen. In deze scenario's kan je alles instellen, van het weer tot en met objecten die zich kunnen verplaatsten. Al deze instellingen zijn nodig

om zo goed mogelijk de realiteit te kunnen simuleren. Doordat de simulator op de werkelijkheid lijkt, is het mogelijk om radaroperators zo goed mogelijk te kunnen opleiden aan de hand van verschillende gesimuleerde omstandigheden.



Figuur 2: Electronic Warfare Simulator

De radar is een belangrijk onderdeel in het realistischer maken van de Electronic Warfare simulator. Want de radar speelt namelijk een belangrijke rol in de hedendaagse oorlogsvoering en dus ook in de Electronic warfare simulator. De radar is (feitelijk) de enige manier om een vijandig toestel bijtijds op te merken. Zoals je misschien wel eens in films of documentaires hebt gezien, werd de radar al gebruikt in de tweede wereld oorlog. Maar waarom is deze “oude” techniek nou zo belangrijk voor de Electronic Warfare Simulator? Het antwoord op deze vraag ligt voornamelijk bij de vijand. De vijand, die niet gezien wil worden op de radar, gebruikt steeds meer technieken om niet of zo min mogelijk op het radarbeeld te verschijnen of zelfs het radarbeeld te verstoren. Deze technieken zijn bijvoorbeeld toegepast op stealthvliegtuigen van het type F117 of B2. Het is ook mogelijk om een radarbeeld te verstoren door bijvoorbeeld Chaff¹ of andere verstoringmethodes. Het wordt hierdoor steeds moeilijker voor bijvoorbeeld een radaroperator om vijandige toestellen tijdig te zien aankomen. De radaroperator heeft een belangrijke taak als het gaat om de veiligheid van een schip of een land. Het is daarom belangrijk dat de radaroperator zo realistisch mogelijk wordt opgeleid voor diverse scenario's. Omdat dit in veel gevallen onmogelijk is in de echte wereld, moet er gebruik gemaakt worden van een radarsimulator.

¹ Zie ook de begrippenlijst

1.2.1 WERKING RADAR

Een radar werkt doordat er een radarsignaal de lucht in wordt gestuurd. Dit signaal botst tegen een object op en reflecteert onder andere terug naar de radar. Echter de meeste radar simulatoren maken geen gebruik van gesimuleerde radarsignalen. Doordat deze simulatoren precies weten waar de objecten zich bevinden, worden deze op de goede locatie op de radar getoond. Echter klopt dit niet met de werkelijkheid, omdat er in de werkelijkheid het signaal verandert aan de hand van verschillende oorzaken. Zo zorgt een regenbui ervoor dat het signaal wordt afgebogen. Dit kan er voor zorgen dat bijvoorbeeld een vliegtuig op de radar ergens anders lijkt te vliegen dan in de (gesimuleerde) werkelijkheid. De Electronic Warfare Simulator is hierin anders. De simulator gaat in plaats van de positie te simuleren, de radarsignalen simuleren. Hierdoor kan er beter rekening gehouden worden met de omgevingsvariabelen zoals het weer en andere objecten. Voor de radaroperator het beeld zou er zo realistischer eruit komen te zien. Hierdoor kan er dus beter getraind worden op het gebruik van een radar.

1.3 RELATIE MET DE ORGANISATIE

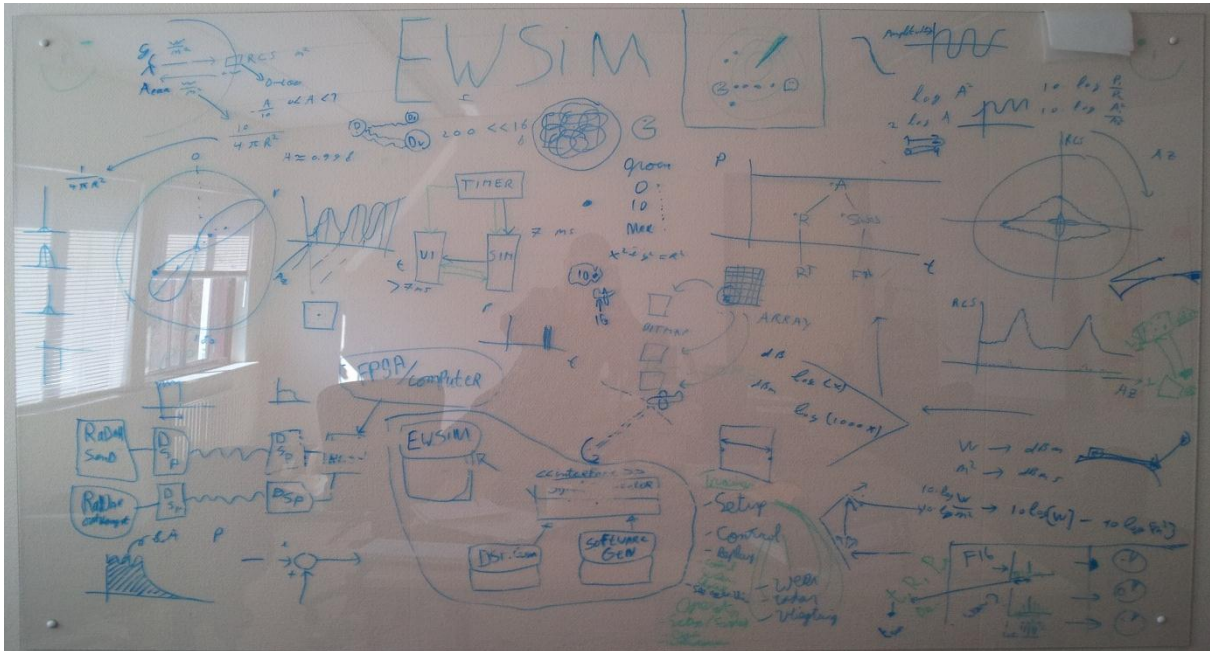
Het afstudeerproject heeft twee belangrijke relaties met het bedrijf. Als eerste is dit project bedoeld als onderzoek naar nieuwe technieken. Zeker het gebruik van DSP technieken is voor het bedrijf nieuw. Met een project zoals dit wil het bedrijf innoveren om zo mee te kunnen concurreren met andere bedrijven.

Als tweede wil het bedrijf ook aan hun klanten laten zien wat ze kunnen. Zonder werkende demo's zullen klanten minder snel met het bedrijf in zee gaan. Een werkende demo geeft de klant vertrouwen dat een bedrijf die bepaalde taak kan uitvoeren.

1.4 WAAR BEN IK TERECHT GEKOMEN

Binnen het bedrijf heb je twee verschillende soorten afdelingen. Eén van die afdelingen heet "operations". Zoals de naam misschien al doet vermoeden, zijn deze mensen voornamelijk bij de klanten aan het werk. De andere afdeling is bezig met innoverende projecten. Het doel van deze projecten is om kennis op te doen en prototypes of demo's te maken. De tweede afdeling bevindt zich voornamelijk op het hoofdkantoor van de NCIM Groep.

Dit afstudeerproject wordt gezien als een innoverend project en zal dus op het hoofdkantoor van de NCIM Groep worden uitgevoerd. Op deze locatie wordt naast dit afstudeerproject ook het overkoepelde project, waar dit afstudeerproject een onderdeel van is, uitgevoerd. Aan dit project werken nog vijf andere personen mee. Alle teamleden van het project bevinden zich op één kamer. Hierdoor worden we in staat gesteld makkelijk te kunnen communiceren met elkaar, en hebben we ruimte om projectvoorwerpen op te slaan. Deze kamer heeft een eigen aantekeningen bord zoals te zien is in Figuur 3. Op dit bord worden aantekeningen gemaakt en dingen uitgelegd die met het project te maken hebben.



Figuur 3: Aantekeningen bord

2 PROBLEEMSTELLING

Het afstudeerbedrijf, de NCIM Groep, wil graag voorop lopen met de huidige technieken. Om dit doel te bereiken blijft het bedrijf innoveren en hierdoor kennis opdoen van nieuwe technieken of de huidige technieken verder te ontwikkelen.

De NCIM-Groep innoveert voornamelijk op technieken waar vraag naar is. Zo ontdekten ze dat er nog geen realistische simulator was om radaroperators goed te kunnen opleiden. Er zijn wel een aantal simulatoren, maar hierbij is het realisme niet hoog genoeg om de radaroperators goed op te kunnen leiden. Deze simulatoren kunnen de kleine verschillen in het signaal vaak niet goed genoeg simuleren. Deze kleine verschillen zijn vaak essentieel om het radarsignaal goed te kunnen interpreteren. Dit heeft als gevolg dat de radaroperators belangrijke details alleen maar in het echt kunnen leren. In het echt is het echter moeilijk en soms onmogelijk op verschillende scenario's te leren. De radaroperator is hierdoor niet bekend met sommige situaties die hij tegen kan komen. Dit kan een groot risico vormen voor de veiligheid bijvoorbeeld op een schip of zelfs voor een land, zoals Nederland.

Om dit probleem op te vangen werken ze bij de NCIM Groep aan een realistische radarsimulator. De simulator gaat de radarsignalen simuleren zodat deze zo realistisch mogelijk wordt. Om een signaal te simuleren, zal je het signaal eerst moeten genereren. Het genereren van een digitaal signaal is voor een computer vaak te rekenintensief. Hij heeft niet de goede componenten om zonder een te groot snelheidsverlies een digitaal signaal te genereren. Zeker niet als er ook nog rekening moet worden gehouden met overige (realistische) elementen, zoals ruis. Dit wordt helemaal een probleem als een signaal tegen een object op kaatst en vervolgens in meerdere richtingen verspreidt. Hierdoor zal er gebruik gemaakt moeten worden van een andere techniek om dit probleem af te vangen.

3 UITGANGSPOSITIE

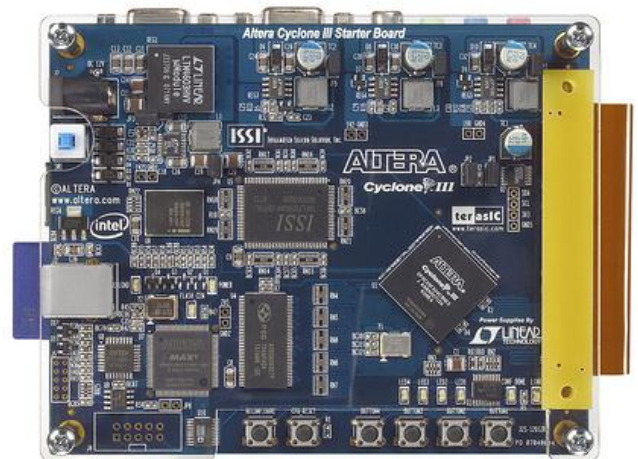
3.1 FPGA

De huidige Electronic Warfare simulator moet hardware matig versneld worden. In eerste instantie was het plan om een FPGA (field-programmable gate array) te gebruiken om berekeningen, die nodig zijn om het radarsignaal te simuleren, parallel te kunnen uitvoeren en zo tijdswinst en effectief meer rekenkracht te verkrijgen. Echter uit een eerder uitgevoerd onderzoek op NCIM, bleek dat er geen snelheidswinst te behalen is op de aanwezige FPGA. Dit komt voornamelijk omdat de huidige FPGA te weinig capaciteit heeft. Dit heeft onder andere als gevolg dat de communicatie tussen de computer en de FPGA te traag is. Een oplossing zou zijn om een betere FPGA aan te schaffen, maar de product owner, Toon Akkermans, vindt hiervoor de prioriteit te laag en ziet liever dat er naar andere technieken wordt gezocht. Mede door de wensen van de product owner en de resultaten van het eerdere onderzoek is uiteindelijk besloten om de FPGA links te laten liggen.

In Figuur 4 en Figuur 5 staat de hierboven besproken FPGA afgebeeld.



Figuur 5: FPGA overzicht



Figuur 4: FPGA achterkant

3.2 RADAR SIGNALLEN AANPASSEN OP BESCHIKBARE HARDWARE

Een echte radar opereert op hoge frequenties van 3 MHz tot 12 GHz. Dit zijn zulke hoge frequenties dat hier speciale en dure hardware voor aangeschaft moet worden. Dit komt omdat normale audio hardware niet in de buurt komen van die frequenties. Om een vergelijking te geven, audio werkt op maximaal 192 KHz. 192 KHz is meer dan tien keer kleiner dan de kleinste radar frequentie van 3 MHz. Om het systeem enigszins betaalbaar te houden, zal de werkelijke frequentie aangepast moeten worden. Hierbij wordt er van uit gegaan dat deze aanpassing geen groot effect heeft op het uiteindelijke resultaat.

4 ANALYSE VAN DE PROBLEMEN

Zoals aangegeven in hoofdstuk 2, heeft de computer veel rekenkracht nodig om een radarsignaal realistisch te kunnen simuleren. De hoeveelheid rekenkracht die nodig is om een goed radarsignaal te simuleren is echter zo groot dat de meeste computers de simulator nooit zullen bijhouden. Als de computer de simulator niet meer realtime kan bijhouden, kunnen er timing problemen ontstaan. Deze problemen zullen zich voornamelijk voordoen als er meerdere computers of hardware aan de simulator zijn aangesloten, omdat dan de verschillende versies van de simulator niet meer synchroon draaien. Verder voelt het voor de student niet realistisch aan of kan het zelfs als vervelend worden beschouwd als de simulator niet realtime draait. Om wat van de rekenkracht uit handen te nemen, moet er gekeken worden naar alternatieven voor de computer.

4.1 VERSCHILLENDE HARDWARE MOGELIJKHEDEN BEKIJKEN

Er zijn twee verschillende alternatieven die in aanmerking komen voor deze taak. Zo is er een FPGA (field-programmable gate array) en een DSP (Digitale signaalprocessor)². Deze twee alternatieven, waar in de volgende hoofdstukken dieper op in wordt gegaan, hebben beide hun sterke en zwakke kanten. De architectuur en de functie van beide apparaten zijn verschillend. Hierom is het voor de stage opdracht belangrijk om eerst te onderzoeken welke van de twee het beste past bij de opdracht. Zoals al in hoofdstuk 3 te lezen is, is de FPGA geen optie voor deze stage. Hierdoor blijft er nog maar één optie over, de DSP. Om de mogelijkheden van de DSP zo goed mogelijk te benutten, zal er eerst een vervolg onderzoek moeten komen over de DSP.

4.2 ONDERZOEK NAAR DE WERKING VAN EEN DSP

Nadat besloten is dat de DSP gebruikt gaat worden, is het belangrijk om dit apparaat dieper te gaan onderzoeken. Hierin moet duidelijk worden wat het apparaat allemaal kan, hoe die werkt en hoe die geprogrammeerd moet worden. Dit moet er voor zorgen dat er voldoende kennis aanwezig is om het risico op kleine “beginners”-fouten te minimaliseren. Verder moet dit onderzoek er ook voor zorgen dat problemen met betrekking tot hardware tijdig worden gedetecteerd en opgelost kunnen worden. Dit is nodig omdat het werken met hardware vaak de nodige problemen kan opleveren. Zoals hierboven is genoemd kunnen sommige van deze problemen, zoals hardware limieten en bekende problemen van andere gebruikers, tijdig gedetecteerd worden. Echter, er kunnen altijd nog onbekende problemen opduiken die nog niet bekend zijn en dus ook niet in het onderzoek naar voren komen.

4.3 DE OPDRACHT

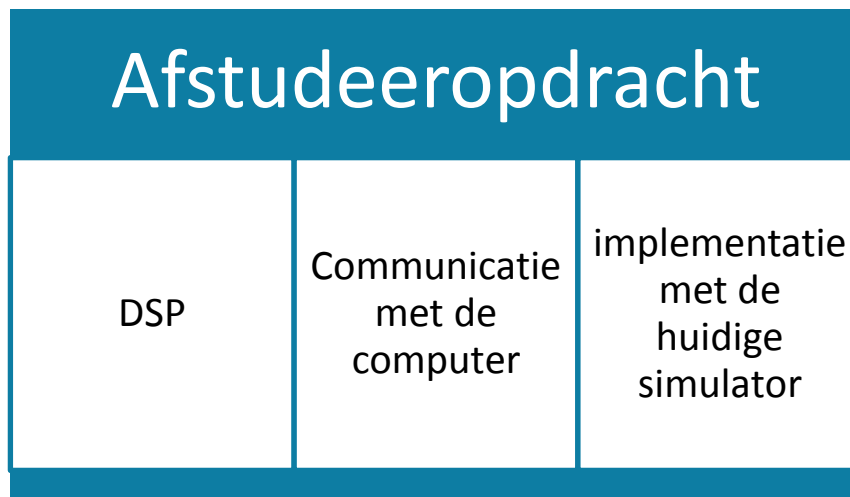
Nadat het onderzoek uit hoofdstuk 4.2 is beëindigd, kan de gekozen hardware gebruikt worden om een radarsignaal te simuleren. De DSP moet door een computer aangestuurd en uitgelezen worden. Het is van belang dat dit tijdig gebeurt. Duurt dit te lang, dan bestaat er een kans dat er timing problemen gaan optreden.

4.3.1 DSP

De opdracht kan in drie belangrijke stukken worden opgedeeld. Zoals te zien is in Figuur 6 bestaat het eerste onderdeel van de opdracht uit het schrijven van de code voor de DSP. De code wordt waarschijnlijk de grootste uitdaging van de opdracht. Het is namelijk op een DSP niet

² Zie ook de begrippenlijst

gebruikelijk om gebruik te maken van een hogere programmeertaal zoals C of Java. Op een DSP heb je namelijk minimale resources tot je beschikking. Hierdoor moet een applicatie zo goed mogelijk geoptimaliseerd zijn. Met een hogere programmeertaal heb je geen tot nauwelijks grip op deze precisie optimalisaties. Verder heeft de DSP een krachtige instructieset waarbij je bijvoorbeeld sommige instructies parallel kan laten uitvoeren. Voor een hogere programmeertaal is het meestal onmogelijk om deze instructies optimaal te benutten. Programma's geschreven in Assembly zijn hierdoor meestal veel kleiner, een paar honderd lijnen in Assembly tegen tienduizend lijnen in een hogere programmeertaal. Hierdoor zal code die geschreven is in een hogere programmeertaal vaak meer instructies bevatten, langzamer zijn en meer geheugen gebruiken (Smith, 1997). Verder is er voor de te gebruiken DSP geen ondersteuning voor een hogere programmeertaal vanuit de fabrikant. Voor dit project zal er dan ook op instructie niveau, oftewel Assembly, moeten worden geprogrammeerd. De grootste moeilijkheid van Assembly zit hem in het feit dat een instructieset op elke chip anders is. Voor elke nieuwe instructieset, zal opnieuw moeten worden gekeken wat er allemaal mee kan. Ook het vinden van hulp is vaak lastig, omdat het vaak om een specifieke en hardware afhankelijke vraag gaat.



Figuur 6: indeling opdracht

4.3.2 COMMUNICATIE COMPUTER

Het volgende punt in Figuur 6 is de communicatie met de computer. Tijdens dit deel van de stage moet er worden uitgezocht hoe de communicatie met de computer het best kan verlopen zodat er geen timing problemen ontstaan. Dit punt is waarschijnlijk het kleinste en eenvoudigste punt van mijn afstudeeropdracht.

4.3.3 IMPLEMENTATIE SIMULATOR

Het laatste punt is de implementatie met de huidige simulator. De huidige simulator code biedt nog geen ondersteuning voor een DSP en dit is wel noodzakelijk voor het project. Waarschijnlijk hoeft er weinig te worden aangepast in de huidige simulator code. Het belangrijkste onderdeel dat veranderd moet worden, is dat de simulator niet meer verantwoordelijk is voor de simulatie van het signaal. Deze verantwoordelijkheid moet verplaatst worden naar de DSP. Dit is waarschijnlijk niet meer werk dan enkele functies aanpassen in de simulator. Als bijkomend voordeel is dit het enige punt waarbij de teamleden van het EWsim project ook mee kunnen

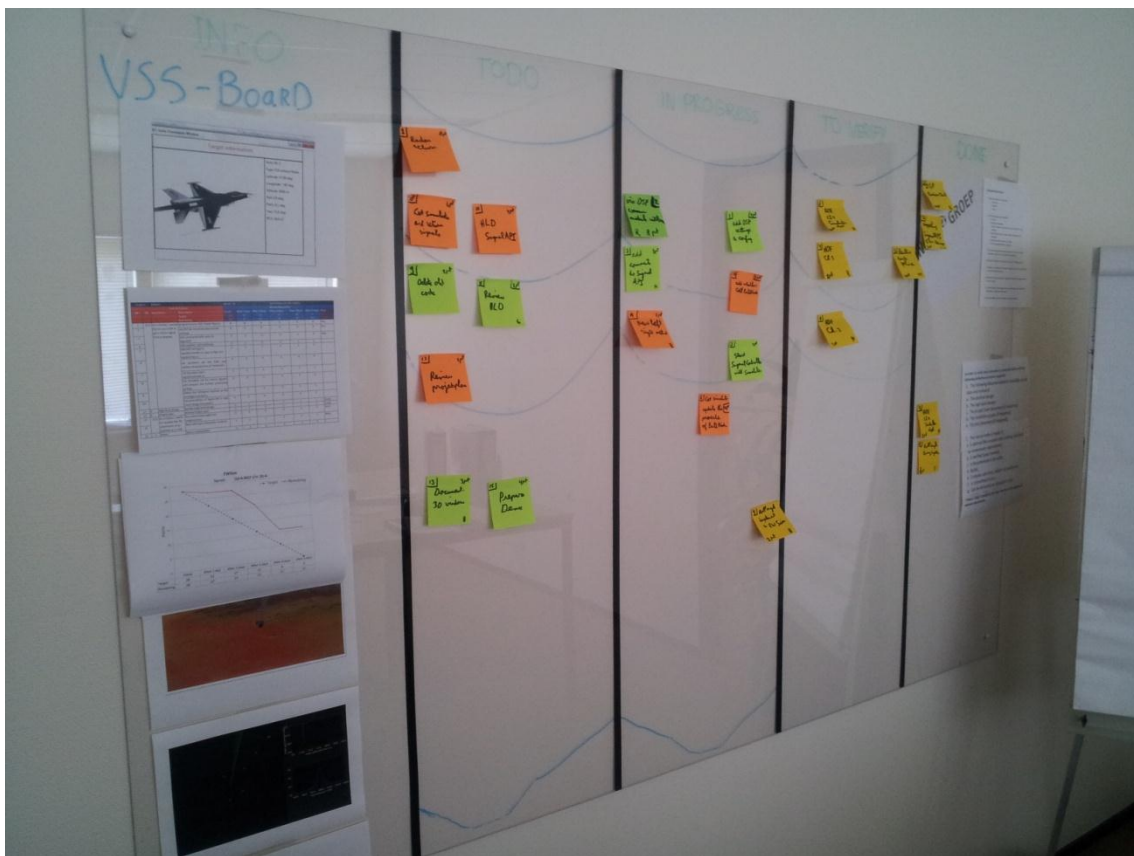
helpen. Dit zou het risico op problemen moeten verkleinen, omdat er meerdere mensen aan de code kunnen zitten. Meer mensen betekent namelijk meer mankracht en denkkracht.

4.4 PROJECT METHODE

Om de kwaliteit van de software te garanderen en de kans op problemen zo laag mogelijk te houden, wordt er gebruik gemaakt van de project methode scrum. Scrum is een raamwerk voor agile management ("flexibel beheren") van softwareontwikkeling. Er wordt gewerkt in multidisciplinaire teams die in korte sprints, van één week, werkende software opleveren. Tijdens deze sprints wordt een takenpakket voor die sprint opgesteld. De vooruitgang van de taken worden vaak bijgehouden op een scrumbord zoals te zien is in Figuur 7. Bij scrum worden de experts uit de verschillende fasen bij elkaar in één team gezet. Het team wordt geleid door de "scrum-master" en houdt vrijwel dagelijks bij aanvang van de werkdag een zogenaamde "scrum-meeting" (ook wel "standup-meeting" genoemd). In deze ontmoeting die ongeveer 15 minuten duurt, beantwoordt elk teamlid de volgende drie vragen:

- Wat heb je gedaan?
- Wat ga je doen?
- Wat zijn je problemen?

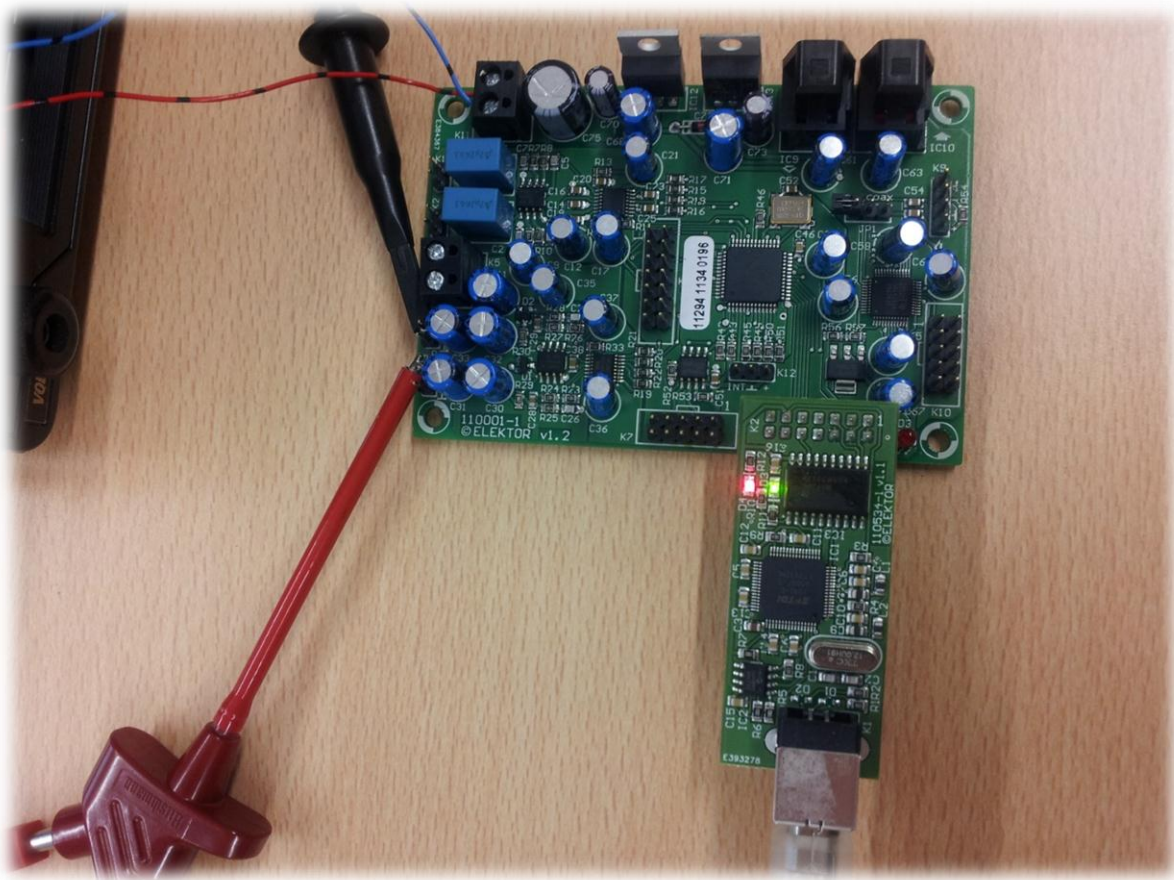
Daarna gaat de expert weer aan het werk met zijn eigen team om de opdracht te volbrengen. De personen werken veel samen en pakken het project met zijn allen tegelijkertijd aan. (Scrum (softwareontwikkelmethode), 2012)



Figuur 7: Scrum bord

5 DOELSTELLINGEN

In hoofdstuk 4.2 is naar voren gekomen om een DSP te gaan gebruiken. De verwachting is dat de DSP sneller signalen kan genereren dan een computer. Het genereren van signalen is belangrijk voor het realisme in het systeem. Niet alleen het signaal kan aan de hand van de omgevingsfactoren, realistischer bewerkt worden. Maar ook voor de radaroperator, die het signaal moet interpreteren, ziet het signaal er realistischer eruit. Verder wilde de product owner dat het bedrijf meer ervaring krijgt met DSP's. Om deze redenen zal het gebruik van een DSP goed moeten passen in dit project. In Figuur 8 bevindt zich de DSP die uiteindelijk gebruikt gaat worden in dit project.



Figuur 8: DSP

Het realistischere genereren van signalen zonder dat de snelheid te nadelig wordt beïnvloed is met een gewone computer onmogelijk. Hiervoor moet er gekeken worden naar een alternatieve techniek, zoals in het geval van deze stage een DSP. De doelstelling van deze stage is daarom ook om uit te vinden hoe een DSP kan worden gebruikt om een zo realistisch mogelijk signaal te simuleren, zonder dat de snelheid van de simulator te nadelig wordt beïnvloed.

5.1 ONDERZOEKSVRAGEN

De doelstelling is om de volgende onderzoeksvragen te beantwoorden:

- Hoe kan een DSP worden gebruikt om een zo realistische mogelijk signaal te simuleren, zonder dat de snelheid van de simulator te nadelig wordt beïnvloed?
 - Hoe heeft het genereren van het teruggekaatste radarsignaal invloed gehad op het realistischer maken van het signaal?
 - Hoe heeft de weersvariabelen, zoals de luchtdruk, temperatuur en luchtdichtheid, invloed gehad op het realistischer maken van het signaal?
 - In hoeverre heeft de DSP invloed gehad op de snelheid van de simulatie?
- In hoeverre komt het gesimuleerde signaal in de buurt van de werkelijkheid?
 - Welke verschillende omgevingsfactoren uit de Moscow zijn er geïmplementeerd?
 - Welke van deze factoren zijn belangrijk geweest voor het realistischer maken van het radarsignaal?
 - Welke omgevingsfactoren uit de Moscow zijn nog niet geïmplementeerd?
 - In hoeverre zouden deze factoren invloed hebben op het realistischer maken van het radarsignaal?
- Is het mogelijk om een DSP binnen het EWSim project te gebruiken voor het genereren van de signalen?
- Wat zijn de voor- en nadelen van een DSP?
 - Wat zijn de voordelen?
 - Wat zijn de nadelen?

5.2 OP TE LEVEREN PRODUCTEN

In dit subhoofdstuk volgt er een uitleg over de producten die naar verwachting worden opgeleverd. Omdat het bedrijf via scrum werkt, en dus dat er elke week opnieuw gekeken wordt naar wat er moet worden gemaakt, is het moeilijk om de op te leveren producten echt in detail te bespreken. Om deze reden is er een Moscow tabel ingevoegd met de mogelijk op te leveren producten.

5.2.1 MOSCOW

5.2.1.1 MUST HAVE

- Een bekende Radar Cross Section gebruiken om het te genereren signaal van de weerkaatsing te berekenen met behulp van een DSP.
- Met behulp van een DSP een signaal te genereren die is beïnvloed door de luchtdruk, temperatuur, luchtdichtheid (uniform over aardoppervlak).
- EWSim code aanpassen zodat een DSP kan worden geïmplementeerd.
- Invloed van neerslag, bewolking en onweer (uniform over het aardoppervlak) op het radarsignaal genereren met behulp van een DSP. Aanname: weer heeft uniforme Radar Cross Section.
- Documentatie van de code.
- Korte handleiding.

5.2.1.2 SHOULD HAVE

- Invloed op de radarontvangst van Chaff genereren met behulp van een DSP.
- Invloed op de radarontvangst van Windmolens genereren met behulp van een DSP.

- Non-uniforme weersinvloeden met eigen Radar Cross section genereren met behulp van een DSP.
- De DSP gebruiken om ruis in het signaal te genereren. bijvoorbeeld ruis dat wordt veroorzaakt door een stad.

5.2.1.3 COULD HAVE

- Invloed op de radarontvangst van Jamming genereren met behulp van een DSP.
- Invloed op de radarontvangst van het terrein genereren met behulp van een DSP.
- Invloed van overige objecten, zoals bomen en huizen, genereren met behulp van een DSP.

5.2.1.4 WOULD HAVE

- De Radar Cross section berekenen met behulp van een DSP/computer(bij de overige punten ga ik er van uit dat de Radar Cross section al bekend is.)

Zoals te zien in de Moscow is de DSP en de radarsignalen een erg belangrijk onderwerp. Daarom is met de hulp van een DSP de radarsignalen te genereren, het hoofdproduct dat wordt opgeleverd.

Naast de DSP is het voor het bedrijf ook erg belangrijk dat alles goed gedocumenteerd wordt. Dit project is een innoverend project. Dit wil zeggen dat het bedrijf dit project gebruikt om ervaring op te doen. Als dan de documentatie ontbreekt, dan is het eigenlijk allemaal voor niets gedaan. Daarom staat dit bij de must have.

6 ONTWERP

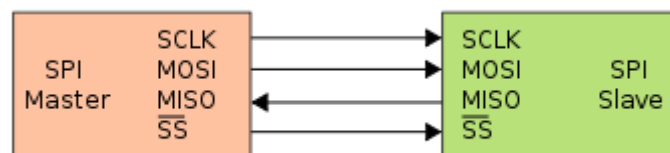
6.1 COMMUNICATIE MET DSP

De Communicatie met de computer is een belangrijk onderdeel van de DSP. Door de communicatie wordt het mogelijk om realtime verschillende waarden in de DSP te veranderen. Dit is noodzakelijk om de verschillende DSP's juist te kunnen instellen.

6.1.1 SPI

De enige manier om via data te kunnen communiceren met de huidige DSP is via een SPI (Serial Peripheral Interface Bus). SPI is een synchroon serial data link standaard die werkt in full duplex mode.

De SPI bus wordt ook soms "4 wire bus" genoemd, omdat een SPI schakeling bestaat uit vier verschillende lijnen. Deze verschillende lijnen zijn terug te vinden in Figuur 9 en worden hieronder verder toegelicht.



Figuur 9: Verschillende SPI lijnen

Uitleg van de 4 logische signalen die in Figuur 9 staan afgebeeld:

- SCLK: Serial clock. Gegenereerd door master.
- MOSI: Master output , slave input
- MISO: Master input , slave output
- SS: Slave select. (actief laag)

Andere veel gebruikte benamingen voor de 4 draden:

- CLK, SCK : Serial clock
- SDI, DI, SI : Serial data in
- SDO, DO, SO Serial data out
- nCS, CS, CSB, CSN, nSS, STE: chip select

Het systeem bestaat uit maximaal één master en één of meerdere slave apparaten.

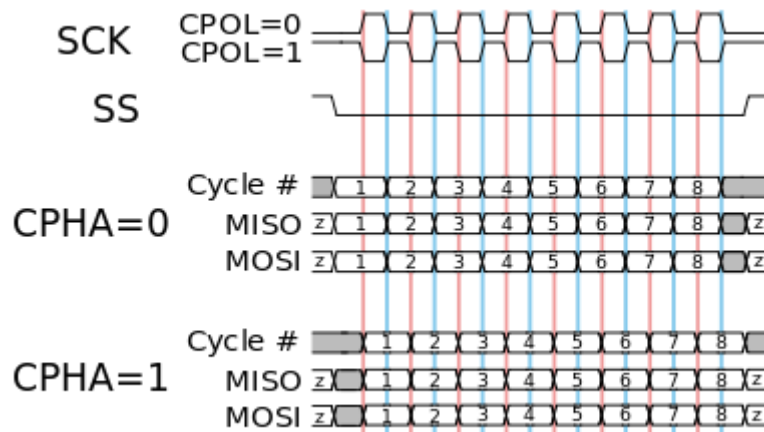
6.1.1.1 WERKING SPI

Als er een slave geselecteerd is door middel van de SS, dan mag die slave luisteren en schrijven op de MOSI en MISO lijnen. Mochten er meerdere slaves zijn aangesloten, dan mogen deze niet luisteren totdat ze geselecteerd worden door de master.

SPI kent enkele verschillende modes waar de SPI in kan werken. Je kunt bijvoorbeeld de CPOL en CPHA modes instellen. Zoals in Figuur 10: timing SPI te zien is heeft CPOL te maken met de

klok(SCLK) en CPHA met de MOSI en MISO. Als CPOL op 0 wordt gezet, dan is de SCLK lijn standaard laag. Bij elke klok puls wordt deze lijn hoog. Bij CPOL 1 is dit precies het omgekeerde.

CPHA geeft aan wanneer de SPI-lijnen uitgelezen kan worden. Zoals te zien in het figuur hieronder begin je bij CPHA 0 in het midden van een verzonden bit en eindig je bij het einde van een bit. Bij CPHA 1 is dit precies omgekeerd. Hierbij begin je bij het begin van de bit en eindig je in het midden.



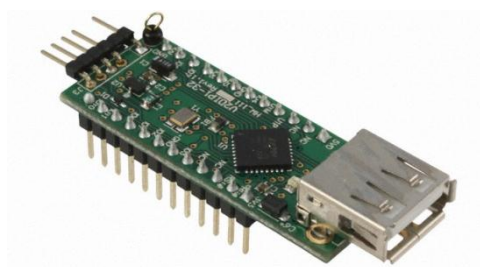
Figuur 10: timing SPI

6.1.2 COMMUNICATIE VIA SPI

De meest voor de hand liggende oplossing om te communiceren met meerdere DSP's is doormiddel van een microcontroller. De microcontroller is dan het master device en communiceert met de DSP's die in slave mode draait. Dit bleek echter geen oplossing, omdat de huidige DSP door hardware beperkingen nooit in slave mode gezet kan worden. De microcontroller in slave modes zetten en de verschillende DSP's in master modes zetten is ook geen optie omdat een SPI lijn maar één master kan hebben.

6.1.2.1 VINCULUM-II

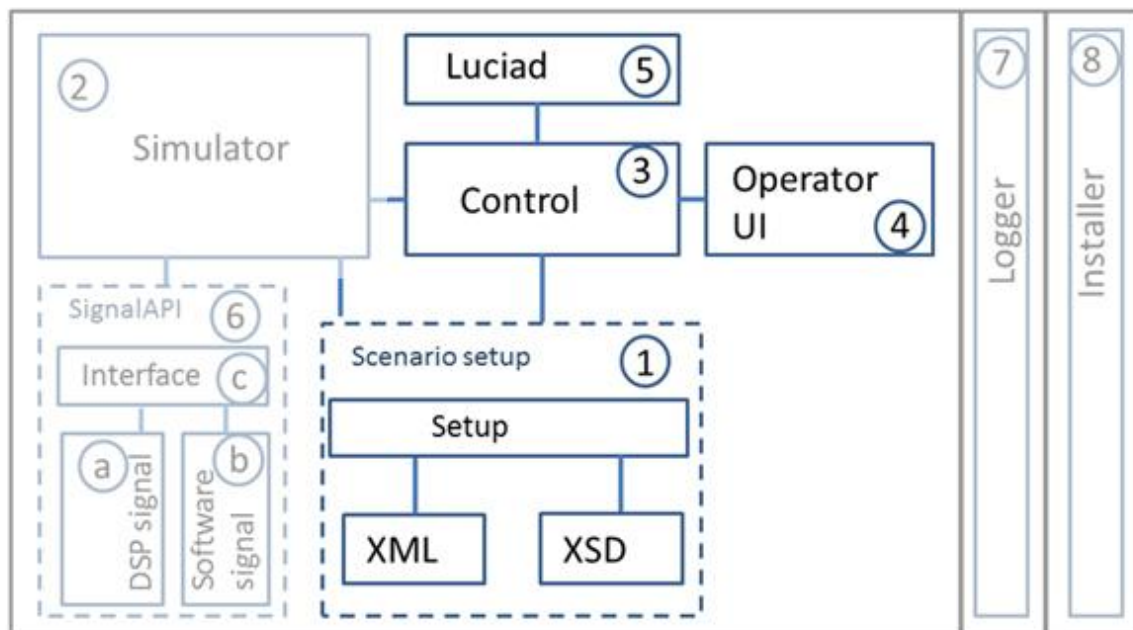
De oplossing moest dus ergens anders gezocht worden. Het volgende idee is om met behulp van een USB naar SPI converter de SPI lijn op een USB port aan te sluiten. Hier bleken verschillende chips voor beschikbaar te zijn. Uiteindelijk hebben we gekozen om Vinculum-II chip van FTDI, te zien in Figuur 11, te gaan gebruiken. Deze chip bleek een USB naar SPI (en omgekeerd) converter te hebben. Verder is er genoeg informatie, zoals voorbeeld code en documentatie, over deze chip te vinden. Gezien de hiervoor genoemde argumenten en het feit dat hij gemakkelijk verkrijgbaar is, hebben we besloten om deze chip te gaan gebruiken voor de communicatie met de DSP.



Figuur 11: De Vinculum-II

7 PROTOTYPE

In dit hoofdstuk wordt uitgelegd hoe de hardware en code is opgebouwd. In Figuur 12 bevindt zich een systeem overzicht van het hele EWsim project. Er zitten onderdelen in die buiten de scope van dit afstudeerproject vallen, maar zullen in de volgende hoofdstukken toch kort toegelicht worden.



Figuur 12: Systeem overzicht: Setup venster, Luciad maps, control- en operatorvenster

7.1 SCENARIO SETUP (1)

Punt nummer één uit Figuur 12 is het Scenario setup scherm. Dit is het startscherm van de applicatie. In dit scherm kan de gebruiker een bepaald scenario kiezen dat is opgebouwd in XML. In dit scenario staan alle gegevens die tijdens de simulator moet worden uitgevoerd. Na het inladen van een scenario, kan de gebruiker alle gegevens van het scenario aanpassen en uitlezen. Mocht alles goed staan ingesteld, dan kan de gebruiker de simulator starten door op een startknop te klikken.

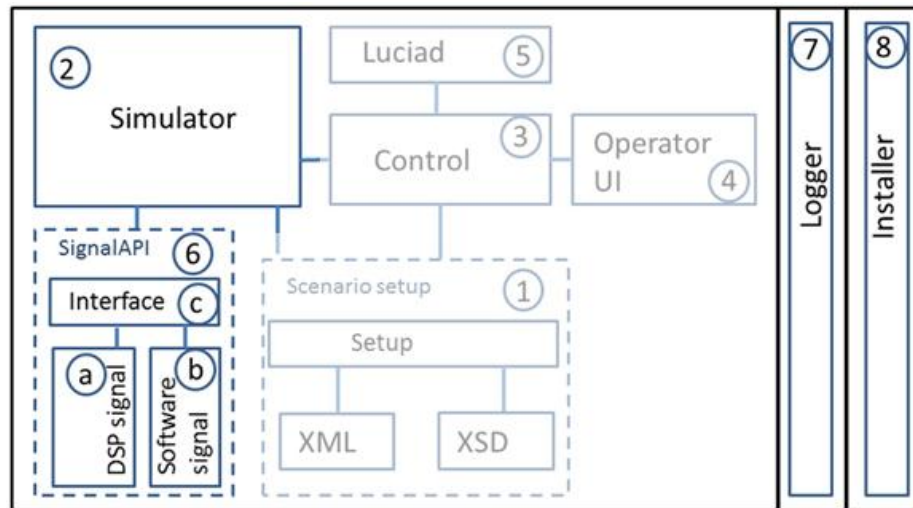
7.2 DE USER INTERFACE (3 & 4)

Als de simulator wordt gestart, dan wordt er een control venster (punt 3) en operatorvenster opgestart (punt 4). In het control venster kan de gebruiker realtime bepaalde gegevens in de simulator aanpassen. Zo kan hij het op een bepaalde plaats bijvoorbeeld laten regenen. Het operatorvenster bestaat alleen uit een radarscherm. Op dit radarscherm zijn de gedetecteerde objecten te zien. Zoals te zien in Figuur 12 heeft de operatorvenster geen direct contact met de simulator. Alle communicatie met de simulator is dan ook de verantwoordelijkheid van het control venster.

7.3 LUCIAD(5)

Verder is te zien in Figuur 12 dat het control venster ook een connectie heeft met Luciad (punt 5). Luciad is een leverancier van 2D en 3D kaarten van de wereld. Deze kaarten worden

vaak gebruikt in militaire toepassingen. Luciad is een toepassing die is gemaakt in Java. De simulator is zelf in C# gemaakt. Hierdoor zal er een brug van Java naar C# moet worden geschreven om Luciad te kunnen gebruiken. Deze brug moet nog geschreven worden. Hierdoor is Luciad tijdens het schrijven van dit verslag nog niet geïmplementeerd. Lucid is al wel opgenomen in Figuur 12, omdat de verwachting is dat Lucid in de nabije toekomst geïmplementeerd gaat worden.



Figuur 13: Systeem overzicht: logger, installer, simulator en signalAPI

7.4 LOGGER EN INSTALLER (7 & 8)

Aan de rechterkant van Figuur 13 staan nog de twee onderwerpen die betrekking hebben op het hele EWSim project. Dit zijn de onderwerpen “Logger” (punt 7) en “Installer” (punt 8). De logger, zoals de naam al doet vermoeden, is verantwoordelijk voor de logging van de applicatie. In deze applicatie wordt gebruik gemaakt van de logger NLog. Deze is geschreven voor C# en zeer uitgebreid. Het andere onderwerp, de installer, is een standaard Windows installer die wordt gebruikt om de applicatie te kunnen installeren. Deze installer is voornamelijk bedoeld om eenvoudig een demo van de applicatie te kunnen geven. Het enige wat nodig is, is een USB-stick of een CD met de installer erop.

7.5 SIMULATOR (2)

In de overige twee punten, de simulator (punt 2) en de signaal API (punt 6), zijn in de huidige afstudeerstage aanbod gekomen. De simulator is het hart van de applicatie. De simulator is namelijk verantwoordelijk voor alles wat er in de applicatie gebeurt op het gebied van de simulatie. Dit betekent onder andere dat de simulator verantwoordelijk is voor het updaten van de vliegtuigen, weer, ect... De simulator was in eerste instantie ook verantwoordelijk voor het simuleren van het radarsignaal. Door de komst van de DSP moest dit echter veranderen. Hierdoor is punt 6, de signaal API, ontstaan. De simulator blijft echter nog wel verantwoordelijk voor de verschillende eigenschappen die invloed hebben over het radarsignaal. Alleen het verzenden en simuleren van het radarsignaal is nu in handen van de signaal API.

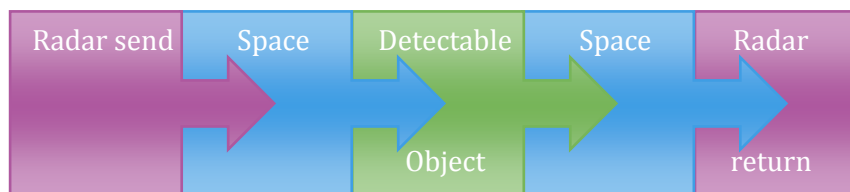
7.6 SIGNAAL API (6)

Zoals beschreven in het vorige hoofdstuk, is de signaal API verantwoordelijk voor het simuleren van het radarsignaal. Het was een vereiste dat de signaal API zowel softwarematig als via een

DSP te gebruiken is. Dit is nodig omdat het anders voor de overige teamleden onmogelijk is om de simulator te testen zonder DSP.

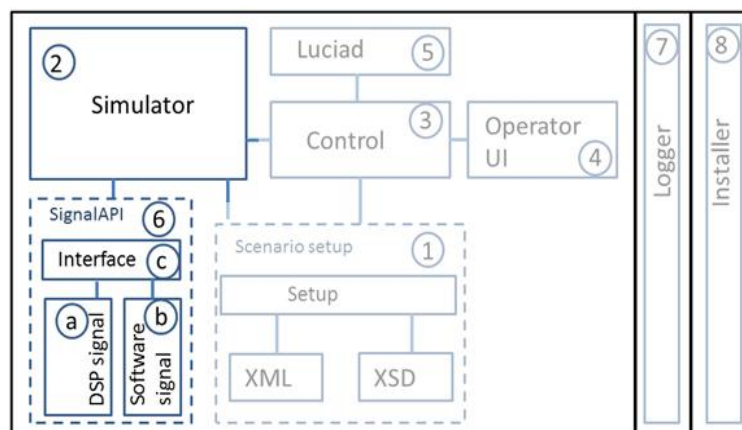
7.6.1 SIGNAAL PAD

De signaal API bestaat uit verschillende pad nodes. Een pad node kan gezien worden als een puzzelstukjes die je in elkaar kan klikken. Meer informatie hierover volgt later in dit hoofdstuk. Iedere pad node heeft zijn eigen functionaliteit. Zo is er een “radar send”-node die een signaal verzendt. Verder is er bijvoorbeeld ook een “detectable object”-node die een bepaalde RCS toepast. Door de nodes aan elkaar te koppelen ontstaat er een signaal pad. Een dergelijk pad kan worden gezien als puzzelstukjes die je in elkaar kan zetten. Dit principe staat afgebeeld in Figuur 14. Zoals bij elke puzzel heb je een begin en een eind stukje. Aan zo’n stukje, hier de radar send en de radar return, kan maar aan één kant iets gekoppeld worden. Deze stukjes zijn dan ook het begin en het einde van het signaal. Tussen deze twee stukjes, kunnen zo veel stukjes geplaatst worden als nodig is. Dit betekent dat het eenvoudig is om bijvoorbeeld een regen bui ergens tussen te plaatsten. Het enige wat moet gebeuren is het stukje los halen en het nieuwe stukje ertussen plaatsten. Echter als de DSP gebruikt wordt om het signaal te simuleren, dan stelt elk stukje een eigen DSP voor. Dit betekent dat het maximale aantal stukjes dat je kunt gebruiken, gelijk staat aan het aantal beschikbare DSP’s. Ook moeten de DSP aan elkaar gekoppeld zoals de stukjes in elkaar zitten. Dit betekent dat er hardware matige veranderingen nodig zijn om een stukje erin te plaatsten of eruit te halen.



Figuur 14: Signaal pad

7.6.2 SIGNAL CONTROLLER

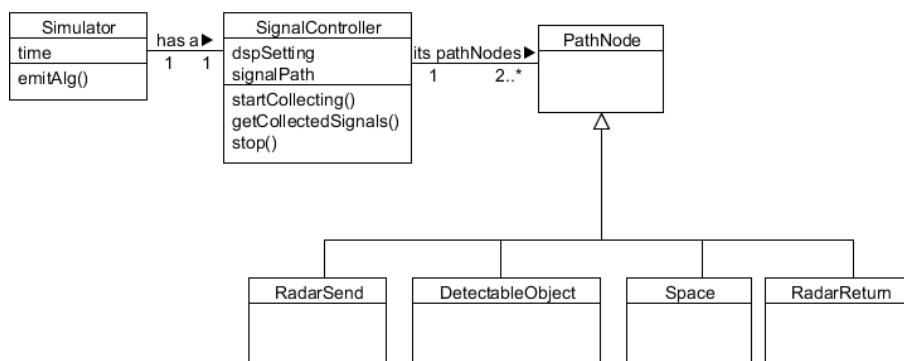


Figuur 15: Systeem overzicht: simulator en signalAPI

Het signaal pad wordt gecontroleerd door de signal controller. De signal controller is de enige aanwezige controller in de Signal API. De signal controller is onder andere verantwoordelijk voor het aanmaken van het signaal pad. Zoals te zien is in Figuur 16 wordt de controller door de simulator aangeroepen. De simulator geeft aan de controller een lijstje door hoe hij het signaal pad wil hebben. Vervolgens gaat de controller proberen om dit pad te maken.

Dit pad kan op twee verschillende manieren gemaakt worden. De eerste mogelijkheid is om alles via de DSP te simuleren. Als tweede kan ook alles via een softwarematige simulatie gebeuren. De gebruiker kan in een configuratie bestand kiezen welke manier hij wil gebruiken. Hierbij heeft de gebruiker de keuze uit drie verschillende standen:

1. **ForceDSP** – hierbij moet er altijd gebruik worden gemaakt van DSP's. Mochten er niet voldoende DSP's aanwezig zijn, dan zal de simulatie een foutmelding geven.
2. **TryDSP** – Hierbij wordt er geprobeerd om DSP's te gebruiken. Mocht dit niet lukken, dan zal de controller overschakelen op softwarematige simulatie.
3. **Software** – Hierbij wordt er nooit geprobeerd om DSP's te gebruiken. Er zal dus altijd gebruik worden gemaakt van softwarematige simulatie.



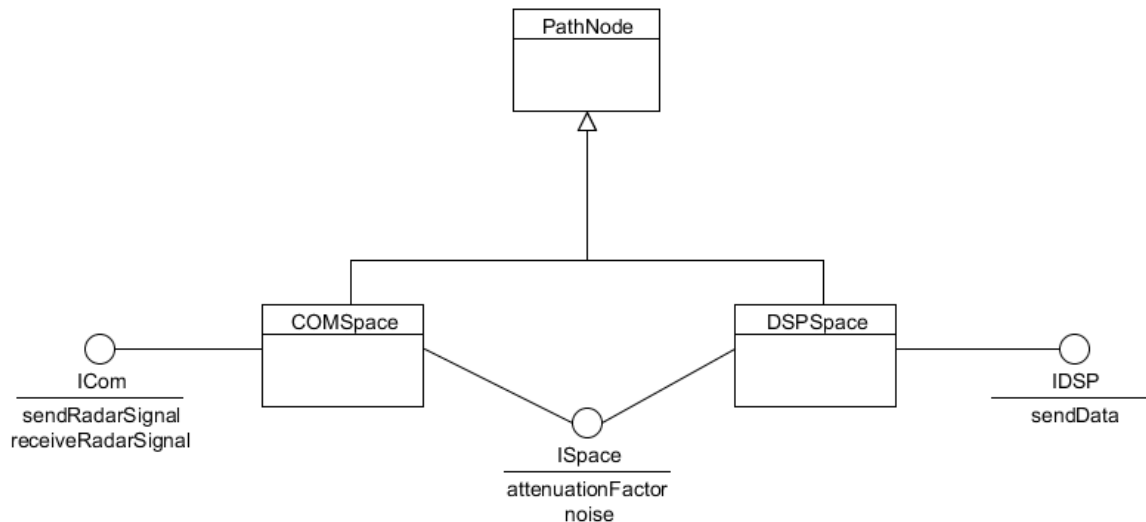
Figuur 16: Klassen diagram Signal API

Standaard staat de “TryDSP” stand aan. Deze stand geeft namelijk het beste van de twee werelden. Mochten er een DSP aangesloten zijn, dan zal hij deze gebruiken. Als er niks is aangesloten, dan gaat hij over op softwarematige simulatie.

7.6.3 PATHNODE

Het volgende onderwerp dat in deze scriptie wordt behandeld is een PathNode. Zoals in het vorige hoofdstuk al beschreven is, is de controller verantwoordelijk voor het maken van een pad. Dit pad wordt gemaakt door verschillende PathNodes aan elkaar te koppelen. Een PathNode is de basis van het puzzelstukje wat in hoofdstuk 7.6.1 is beschreven. Een PathNode heeft alleen de functionaliteit om verschillende stukjes aan elkaar te koppelen. De onderlinge communicatie en overige functionaliteit staan niet in de PathNode beschreven. Om in het puzzel voorbeeld te blijven, beschrijft de PathNode alleen hoe een stukje neergelegd kan worden. Het “plaatje” op het puzzelstukje staat hier dus niet beschreven.

Het “plaatje” op het puzzelstukje staat beschreven in de klassen die de PathNode’s extenden. In het geval van Figuur 16 zijn dit de “RadarSend”, “DetectableObject”, “Space” en “RadarReturn”.



Figuur 17: PathNode overzicht

Om wat dieper op deze klassen in te gaan, is in Figuur 17 het space object eruit gepakt om dit beter toe te lichten. Zoals te zien is in Figuur 17, is het Space object nu is opgedeeld in twee verschillende objecten, namelijk COMSpace en DSPSpace. Het verschil tussen de twee objecten heeft te maken met het type simulatie dat gebruikt moet worden. Verder is te zien dat er enkele interface objecten erbij zijn gekomen.

De Com in COMSpace staat voor computer oftewel een softwarematige simulatie. De DSP in DSPSpace betekent dat er gebruik wordt gemaakt van een DSP voor de simulatie. Omdat deze twee verschillende soorten simulatie, andere functionaliteit nodig hebben, is het nodig om hier onderscheid in te maken. Zoals je al kan zien bij de ICom interface, heeft deze twee functies genaamd sendRadarSignal en receiveRadarSignal. Deze twee functies zijn nodig het radarsignaal door te sturen naar de volgende PathNode. In deze functies moeten de nodige berekeningen gebeuren om het te simuleren signaal aan te passen. Als er vervolgens wordt gekeken naar de IDSP, dan is te zien dat deze interface geen gebruik maakt van deze twee functies. De DSP zitten namelijk hardware matig aan elkaar gekoppeld waardoor het door zenden van het signaal “automatisch” gebeurt. De IDSP heeft echter wel een andere functie nodig, genaamd sendData. Het DSP object moet data, simulatie informatie, doorsturen naar de DSP’s. Dit is een functie die niet nodig is bij het softwarematige deel. Bij softwarematig hoeft er namelijk niks naar hardware door te worden gestuurd.

Het laatste nog niet behandelde interface uit Figuur 17 is ISpace. ISpace is een interface voor beide objecten. De interface beschrijft de eigenschappen van een Space object. Door gebruik te maken van een interface voor beide objecten, maakt het voor de controller niet uit of hij praat met een COMSpace of een DSPSpace. De functies die betrekking hebben op het Space object, blijven gelijk. Natuurlijk kan de implementatie van die functie in beide objecten anders zijn.

7.7 DSP

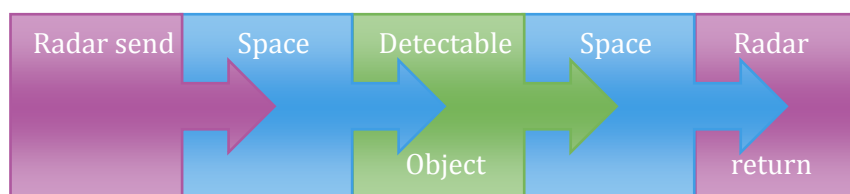
Zoals in het vorige hoofdstuk te lezen is, moet er data naar de DSP gestuurd worden. De dataoverdracht naar de DSP is, zoals te lezen in hoofdstuk 6.1, alleen mogelijk via een SPI-lijn.

Om deze dataoverdracht te versimpelen, hebben we, zoals te lezen in hoofdstuk 6.1.2, er voor gekozen om een USB naar SPI converter te gebruiken.

Elke DSP in de schakeling heeft zijn eigen verantwoordelijkheden. Op de elke DSP's staat ook verschillende stukjes software. Zo is er bijvoorbeeld een DSP die het radarsignaal verzendt en opvangt, maar ook bijvoorbeeld het "space"-object heeft zijn eigen DSP. Het is niet mogelijk om de functionaliteit van een DSP in realtime te veranderen.

Elk stukje code in de DSP is op dezelfde manier opgebouwd. In het eerste deel van de applicatie worden alle instellingen voor het gebruik van de DSP ingesteld. Hier wordt bijvoorbeeld de klok en de te gebruiken pinnen ingesteld. Vervolgens worden alle standaard waarden voor de variabele gezet. Als dit allemaal gebeurd is, komt de applicatie in een oneindige lus. In deze lus worden eerst de audio input lijnen uitgelezen. Vervolgens doet de DSP het ding waar hij verantwoordelijk voor is, bijvoorbeeld een bepaalde delay toepassen. Mocht het nog nodig zijn dan verstuurt hij het aangepaste signaal vervolgens weer door naar de volgende DSP. Als laatste worden er eventueel waarden teruggestuurd of uitgelezen van en naar de computer. Met behulp van deze waarden kunnen de DSP en de computer met elkaar communiceren.

7.7.1 VERSCHILLENDE SOORTEN DSP CODE



Figuur 18: Pad van het radar signaal

7.7.1.1 RADAR

Het eerste stukje DSP code dat hier wordt besproken, is de Radar. De radar code heeft twee verschillende taken. In de eerste taak word er een signaal verzonden. Als het signaal verzonden is, dan gaat de DSP het terugkerende signaal verwerken. Deze twee taken zijn in Figuur 18 beschreven als "RadarSend" en "RadarReturn". De "RadarSend" is verantwoordelijk voor de eerste taak, het zenden van het signaal, van de radar DSP.

De RadarSend kent verschillende eigenschappen die ingesteld kunnen worden. Zo kan bij een radar de frequentie en de zendsterkte worden aangepast. Verder is de duur van het signaal dat wordt verzonden en de duur dat er wordt gewacht op het terug kerende signaal erg belangrijk³. Deze verschillende eigenschappen kunnen echter nog niet realtime worden aangepast. Tijdens het schrijven van deze scriptie, accepteert de RadarSend alleen een signaal dat er moet worden begonnen met het zenden van een radar signaal. De overige eigenschappen kunnen alleen, doormiddel van variabele, handmatige in de code worden aangepast. Door de vele uitdagingen en leermomenten, is deze stage opdracht achter gaan lopen op schema. Om een zo goed mogelijk product af te leveren, is er voor gekozen om alleen de prioriteit op de belangrijkste eigenschappen te leggen.

Net als in een echte radar, kan een DSP niet zenden en ontvangen tegelijkertijd. Mocht de DSP aan het zenden zijn, dan worden alle inkomende radar signalen genegeerd. Als het radarsignaal verzonden is, dan gaat de DSP van de RadarSend taak naar de RadarReturn taak. In deze taak luistert de DSP naar alle signalen die binnen komen. Mocht er een signaal binnen komen, dan

³ In Bijlage C kunnen deze verschillende eigenschappen terug gevonden worden.

wordt aan de hand van dit signaal de daarbij horende dBm berekent. Wanneer de dBm berekent is, wordt deze naar de computer gestuurd. Verder wordt ook de tijd sinds de RadarReturn actief is door gestuurd. Aan de hand van deze twee waardes, kan de simulator vervolgens berekenen waar het object gedetecteerd is en hoe sterk het signaal is. Deze waardes zijn dus nodig om een stip op het radarscherm te kunnen zetten.

7.7.1.2 SPACE

De Space is verantwoordelijk voor de simulatie van de lucht. De space kent twee belangrijke eigenschappen. Ten eerste wordt er een bepaalde vertraging op het binnen gekomen signaal toegepast. Deze vertraging is nodig om de afgelegde afstand te simuleren. Als tweede wordt het binnen gekomen signaal afgezwakt.

7.7.1.3 DETECTABLE OBJECT

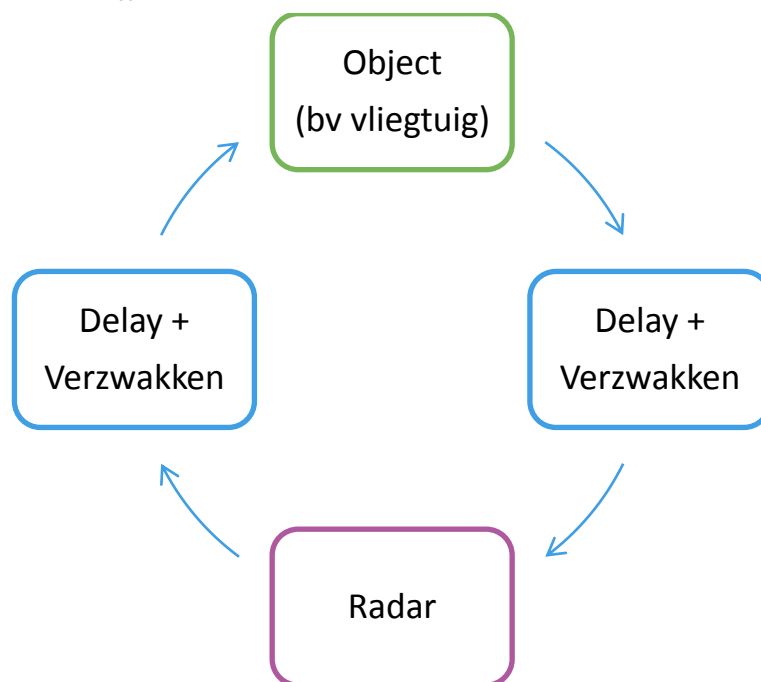
Detectable Object is verantwoordelijk voor het simuleren van een gedetecteerd object. Zo'n object kan bijvoorbeeld een vliegtuig die door de lucht vliegt zijn. Al deze objecten hebben een bepaalde RCS die constant veranderen. Hoe hogere RCS een bepaald object heeft, hoe meer het signaal versterkt moet worden.

8 UITDAGINGEN

8.1 UITDAGING IN HET VERTRAGEN VAN DE TIJD

Een radarsignaal verplaatst zich met ongeveer de snelheid van het licht. Voor dit rekenvoorbeeld rekenen we gemakshalve met ronde (en makkelijke) getallen. Hiervoor stellen we dat de snelheid van het radarsignaal 300.000 km/s is. In dit verslag gaan we ervan uit dat de maximale range van de radar 300 km is. Dit betekent dat het 1 milliseconde duurt voordat het radarsignaal 300 kilometer heeft afgelegd. In $3,33 \cdot 10^{-6}$ seconde is het radarsignaal 1 kilometer verder.

8.1.1 DELAY IN HARDWARE



Figuur 19: Schematische weergave van de baan van het radarsignaal

De huidige vertraging in de huidige DSP(DSP56374) is groot. Als een ADC(CS5340) en een DAC(PCM1781) wordt gebruikt, dan is de vertraging 0,65 milliseconde bij een sample frequentie van 44.1 kHz. Deze vertraging is afhankelijk van de gebruikte frequentie. Als er een hogere frequentie wordt gebruikt, bijvoorbeeld 192 kHz, dan zou de vertraging lager moeten zijn.

Als we voor elke blokje uit een DSP gaan gebruiken, dan hebben we 4 DSP's nodig. De vertraging van de hele schakeling zou dan $4 \cdot 0,65\text{ms} = 2,60\text{ms}$ zijn. Als we bij deze tijd nog de tijd van de maximale range van radar, de 1 milliseconde, hierbij optellen dan komen we uit op 4,60 milliseconde ($2,60 + 1\text{ms} + 1\text{ms}$). De vertraging van alle DSP's bij elkaar, is groter dan de maximale range van de radar.

Om het signaal dan op tijd aan te laten komen bij de computer, moet je het signaal eigenlijk 2,60ms eerder versturen. Hierdoor zou de hardware ook eerder aangestuurd moeten gaan worden. Dit is een probleem omdat de simulator, die de gehele radar simuleert, niet in de toekomst kan kijken om de hardware eerder aan te sturen.

8.1.2 DIGITAAL SIGNAAL

Naar aanleiding hiervan ben ik opzoek gegaan naar alternatieven om de vertraging te verkleinen. Ik heb hiervoor gekeken naar de digitale uitgangen op dezelfde DSP. Om de digitale uitgang te gebruiken moet de DSP communiceren met een SRC chip genaamd "SRC4392". Deze chip heeft een instelbare sample delay om filters zoals Anti-Aliasing toe te kunnen passen. Je kunt hierbij kiezen vanaf 8 samples tot en met 64 samples. Bij het minimale aantal samples (8) is de vertraging van het digitale gedeelte 1,08 milliseconde. De vertraging van elke sample is 20 microseconde, omdat we ook bij deze test een sample rate van 44,1 kHz gebruiken ($1/44,1\text{kHz} = 0,02$ milliseconde). Bij alle metingen met een sample delay van meer dan 8 samples, klopt dit ook. Echter bij 8 samples zou de vertraging dan $8 * 0,02$ milliseconde = 0,16 milliseconde moeten zijn, maar dit blijkt 1,08 milliseconde te zijn. Dit betekent er nog een vertraging is van 0,92 milliseconde die niet bekend is. Het opzoeken van dit probleem (op internet en in de datasheets) heeft geen resultaat opgeleverd en de oorzaak van deze vertraging is nog onduidelijk. Maar zolang deze vertraging er nog is, is het sneller om een analoog signaal te gebruiken.

8.1.2.1 NADELEN DIGITAAL

Buiten het feit om dat de vertraging van een digitaal signaal groter is dan een analoog signaal, is er nog een nadeel van het gebruiken van een digitaal signaal. Het is mogelijk dat er meerdere signalen tegelijk terugkomen bij de radar. In het geval van een analoog signaal kunnen deze signalen gemakkelijk gemengd worden door de signalen over dezelfde lijn te laten gaan. Bij een digitaal signaal gaat dit niet lukken. Als meerdere digitale signalen zich mengen, dan levert dat altijd een signaal op dat niet het gewenste resultaat oplevert.

8.1.3 CONCLUSIE

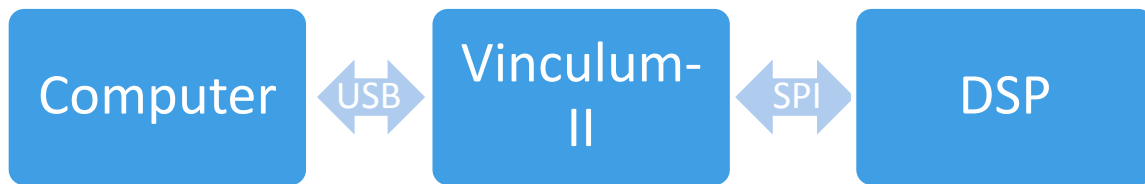
Nadat ik mijzelf de vraag heb gesteld of de delay wel een probleem zou zijn, ben ik tot de conclusie gekomen dat het delay probleem helemaal niet zo groot is als ik had gedacht.

Gezien de beschikbare tijd van mijn afstudeerstage en de constatering dat vertraging klein genoeg is, ben ik van mening dat we gebruik kunnen gaan maken van meerdere DSP's en niet meer hoeven te kijken naar andere hardware.

In de huidige simulator, die om de 7 milliseconde update, vormt de vertraging geen probleem. De maximale delay van de DSP schakeling van 4,6 milliseconde valt ruim binnen de 7 milliseconde die de simulator nodig heeft om te updaten. Zelfs als de simulator sneller zou updaten dan de maximale delay van het signaal, zou dit nog geen groot probleem hoeven te zijn. Als je softwarematig hier rekening mee houdt en een delay toepast op de aansturing, dan is dit probleem goed af te vangen. Verder is een vertraging van enkele milliseconden niet waarneembaar met het blote oog. Pas bij een vertraging van 100 milliseconde zou het waarneembaar kunnen zijn. Dit bevestigt nogmaals dat de vertraging geen groot probleem is.

8.2 SPI UITDAGINGEN

De computer kan via SPI met de DSP communiceren door middel van een communicatie module. Deze module is een V2DIP1-32 van FTDI. Dit is een module die wordt aangedreven door een Vinculum-II chip van FTDI. Hiermee is het mogelijk om een via USB (virtuele com-poort) met de DSP te communiceren. In staat uitgebeeld hoe de communicatie tussen de computer en de DSP gaat.



Figuur 20: communicatie flow

Tijdens de stage periode zijn er veel verschillende problemen geweest met de communicatie tussen de Vinculum-II en de DSP. Veel van deze problemen waren eenvoudig op te lossen. Een probleem heeft echter voor veel vertraging in de vooruitgang van de opdracht gezorgd. Dit probleem treedt alleen op als er data van de Vinculum-II naar de DSP wordt gestuurd. De data die de DSP opvangt is niet het zelfde als de data die wordt verstuurd. Er zit wel een patroon in het probleem. Vanaf de eerste 0, zijn de rest van de bits ook 0. Dit betekent dat als bijvoorbeeld de waarde “01100110” wordt verstuurd, de waarde “01100000” bij de DSP aankomt.

Andere voorbeelden zijn:

“01110000” → “01110000”

“01000010” → “01000000”

“00110010” → “00000000”

“10000000” → “00000001”

“10001100” → “00000001”

Zoals beschreven staat in hoofdstuk 6.1 speelt de communicatie een belangrijke rol in het hele project. Zonder communicatie heeft de DSP geen enkele meer waarde binnen het project. Het is dus noodzakelijk om op zoek te gaan naar een oplossing.

8.2.1 ZOEKEN NAAR OPLOSSINGEN

Om dit probleem op te lossen is er eerst gekeken naar de mogelijke oorzaken die het probleem kunnen veroorzaken. Zo is er geprobeerd om de kloksnelheid extreem te verlagen. Dit kan een mogelijke oorzaak zijn, omdat één van twee apparaten het tempo van de andere niet bij kan houden. Na het verlagen van de kloksnelheid, bleef het probleem bestaan. Er moest dus opzoek worden gegaan naar een andere oorzaak. De volgende oorzaak die bekeken wordt, is het SPI protocol en de verschillende timing die erbij komen kijken. Zoals in hoofdstuk 6.1.1.1 beschreven staat, kent SPI verschillende instellingen. De instellingen van de applicaties leken echter goed te staan, maar voor de zekerheid werden de overige instellingen ook geprobeerd. Dit loste uiteindelijk niks op en de originele instellingen bleken dus al goed te staan.

Om gerichter te kunnen zoeken, moest het duidelijk worden bij welke hardware het probleem ligt. Een manier om hier wat meer informatie over te krijgen, is om twee de zelfde type hardware met elkaar te laten communiceren. Mocht het probleem nogmaals optreden, dan is het bekend waar het probleem zich waarschijnlijk bevindt. Dit idee heeft echter één nadeel. De DSP's kunnen niet aan elkaar gekoppeld worden door middel van SPI. Zoals beschreven is in hoofdstuk 6.1.1, heeft SPI minimaal 1 slave en maximaal 1 master nodig. De DSP's die worden gebruikt in dit project zijn echter hardwarematig op master gezet. Het SPI protocol kan niet overweg met meerdere master en zal dus niet goed werken. Op de communicatie modules was dit probleem niet aanwezig. Deze kunnen in zowel master als slave mode worden gezet. Na het schrijven van een eenvoudig stukje code, was het mogelijk om de communicatie modules te testen. Na

zorgvuldig getest te hebben, was het al snel duidelijk dat het onderling communiceren tussen de twee modules over SPI niet vlekkeloos ging. De miscommunicatie kwam in sommige gevallen overeen met het SPI communicatie probleem. Aan de hand van deze resultaten lijkt de module de schuldige te zijn. Maar omdat er niet op de DSP getest kan worden, zal er nog verder gekeken moeten worden.

Een andere mogelijkheid om het probleem te vinden, is om gebruik te maken van een digitale oscilloscoop. Een oscilloscoop kan een signaal omzetten naar een soort van grafieken. Hiermee wordt het mogelijk om de verschillende signalen te bestuderen. Na het uitvoeren van verschillende testen, kwam er niks onverwachts uit. Tijdens deze testen was het ook duidelijk te zien dat na de eerste 0, de rest van de bits ook 0 zijn, ongeacht wat er werd gestuurd.

Al snel bleek dat dit een probleem zou worden wat niet door mijzelf opgelost kon worden. Hierom is de hulp ingeschakeld van mensen binnen en buiten het bedrijf. Hier kwamen een aantal goede suggesties uit, maar achteraf bleken die ook niet hielpen. De meeste suggesties hadden te maken met de elektronica van het systeem. Zo werd onder andere er voorgesteld om de aarding van het hele systeem goed na te lopen. Ook het controleren van de pull-up of pull-down weerstanden en de compatibiliteit van de logische niveaus werden genoemd. Na een aantal weken te hebben getobd, was het plan om het probleem te accepteren en er omheen te werken. Echter op het allerlaatste moment stelde iemand voor om de draadjes waarmee de module aan de DSP zat een paar centimeter in te korten. Na de draadjes verkort te hebben was het probleem spontaan verdwenen. Het was weer mogelijk om zonder fouten naar de DSP te kunnen communiceren.

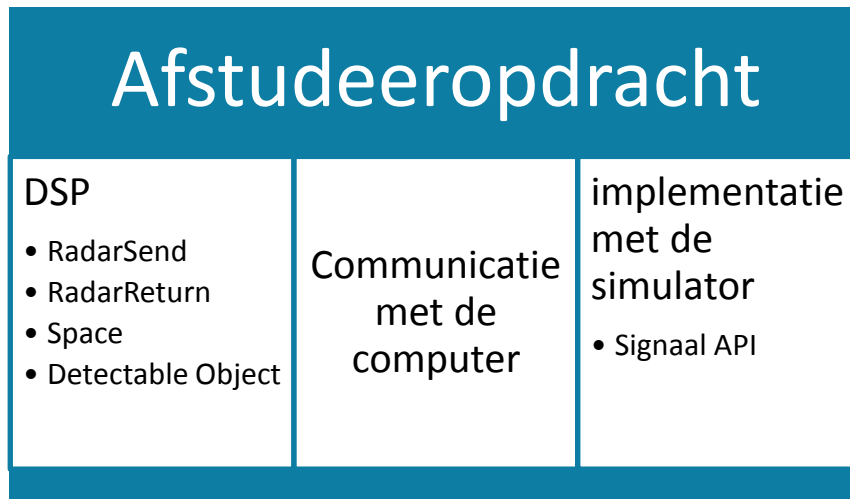
8.3 BOOTLOADER

Elke DSP heeft een bootloader nodig om te kunnen opstarten zonder de hulp van een computer. Deze bootloader haalt het programma, dat staat opgeslagen in een SPI eeprom geheugen, op en plaatst het in zijn eigen geheugen. Deze bootloader staat niet standaard op de DSP waardoor die nog zelf moet mee worden gecompileerd. Dit bleek echter moeilijker dan gedacht, omdat er geen goede voorbeeld code beschikbaar was. Gezien de tijd die we hadden verloren door de verschillende problemen, hebben we besloten om een makkelijkere oplossing te zoeken. De oplossing was uiteindelijk gevonden door het programma via de USB op de DSP te plaatsten en dan de USB eruit te trekken. Hierdoor blijft de applicatie gewoon door draaien en heb je geen bootloader nodig. Dit is uiteindelijk geen goede oplossing, omdat het op deze manier veel tijd kost om alle DSP steeds te flashen.

9 RESULTATEN

In dit hoofdstuk zullen de resultaten van de afstudeerstage beschreven worden. Als eerste worden hier alle opgeleverde producten los besproken. Hierna zal het gaan over de resultaten van het samenvoegen van de DSP met de simulator.

9.1 OPGELEVERDE PRODUCTEN

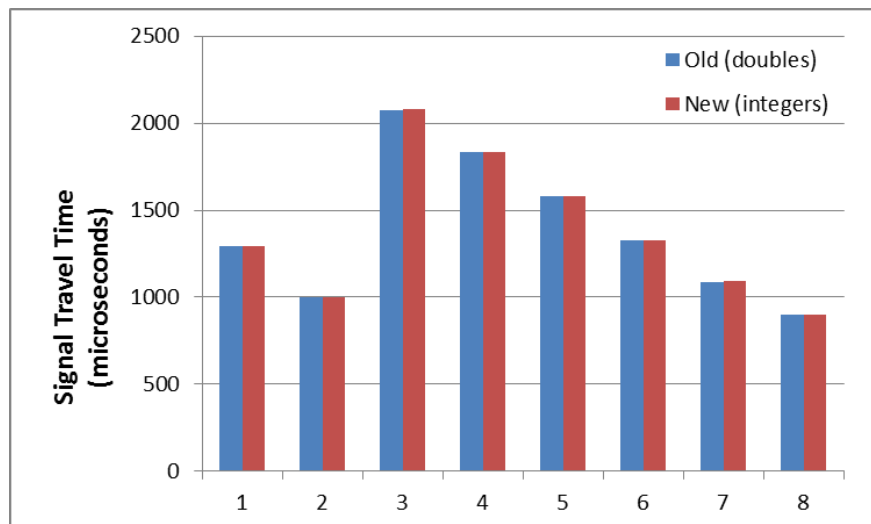


Figuur 21: Opdracht in drie delen

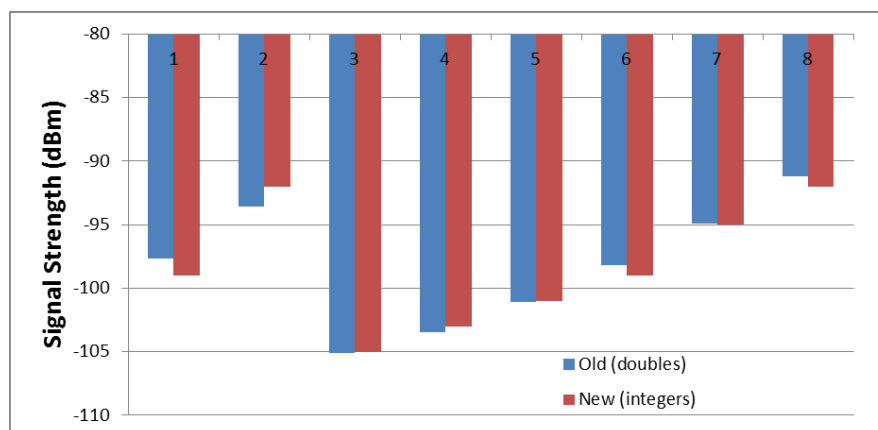
Zoals te zien in Figuur 21 kunnen de opgeleverde producten verdeeld worden in drie verschillende categorieën. Als eerste zal hieronder de implementatie met de simulator besproken worden. Als laatste zal in dit subhoofdstuk de resultaten van de DSP beschreven worden. Het middelste onderwerp uit Figuur 21, de communicatie met de computer, zal in beide hoofdstukken aan bod komen.

9.1.1 IMPLEMENTATIE MET DE SIMULATOR

De implementatie met de simulator kan opgedeeld worden in twee verschillende stukken. Zo is er de softwarematige en de hardwarematige simulatie. Beide simulaties maken gebruik van de nieuwe signalAPI. De softwarematige simulatie kan echter eenvoudig vergeleken worden met de oude versie van de simulator. De uitkomsten van de nieuwe en oude softwarematige simulatie horen namelijk overeen te komen met elkaar. Na het draaien van beide versies, kwam er naar voren dat de resultaten die deze vergelijking opleverde, goed waren. De signalAPI bleek bijna niet af te wijken van de al bestaande simulatie. Het enige verschil zat hem in het feit dat de nieuwe signalAPI nu werkt met integers in plaats van doubles. Hierdoor kunnen er afrondingsfouten ontstaan die de afwijking van nieuwe signalAPI kan verklaren.



Figuur 22: Vergelijking oud met nieuw met de afgelegde tijd



Figuur 23: Vergelijking oud met nieuw met de signaal sterkte

In Figuur 22 en Figuur 23 is te zien wat het verschil is tussen de oude (blauwe lijn) en de nieuwe situatie (rode lijn). Deze resultaten zijn op 8 willekeurige momenten genomen. Op zo'n punt werden de waarden van beide versies berekend. Zoals te zien is in Figuur 22 is het verschil tussen de oude en nieuwe situatie niet merkbaar. In Figuur 23 is er echter een grotere afwijking te zien. Deze afwijking is echter nog steeds zo klein dat het verwaarloosbaar is.

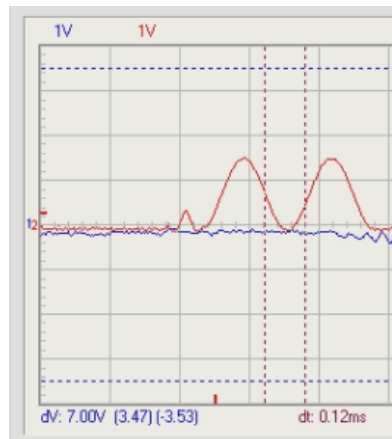
9.1.2 DSP

De verschillende DSP's, zoals beschreven staan in hoofdstuk 9.1.2, kunnen gezien worden als verschillende modules. Deze modules kunnen los getest worden. De resultaten van deze testen kunnen terug worden gevonden in de volgende subhoofdstukken. Alle modules werden door middel van een oscilloscoop getest. Deze oscilloscoop werd gebruikt om de radarsignalen te genereren en uit te lezen. Verder werd er een eigen geschreven computer programma gebruikt om heen en weer te kunnen communiceren met de computer en DSP.

9.1.2.1 RADARSEND

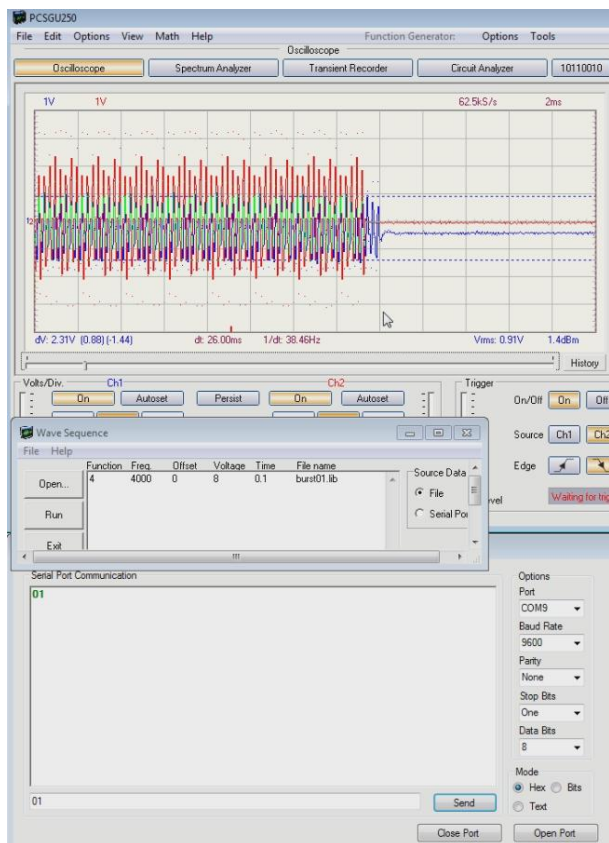
Als eerste wordt de RadarSend behandeld. Zoals beschreven staat in hoofdstuk 7.7.1.1, heeft de RadarSend als verantwoordelijkheid om een bepaald radarsignaal te versturen. Het versturen van dit signaal gebeurt op het moment dat er een hexadecimale waarde naar de DSP wordt

gestuurd. Zolang de waarde niet 0 is, wordt er een radar signaal verstuurd. In Figuur 24 staat een voorbeeld van het signaal dat wordt verstuurd.

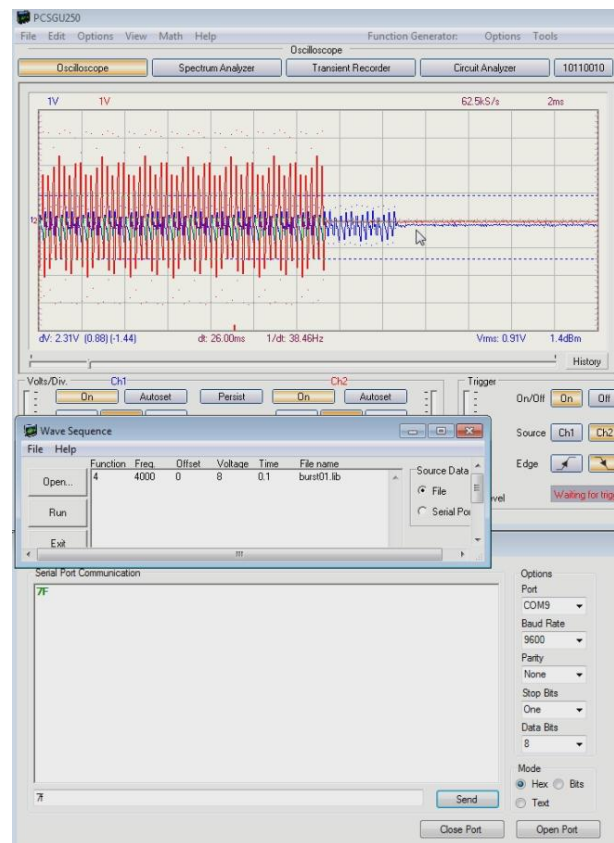


Figuur 24: RadarSend: radar signaal

9.1.2.2 SPACE



Figuur 25: Space: kleine afstand



Figuur 26: Space: grote afstand

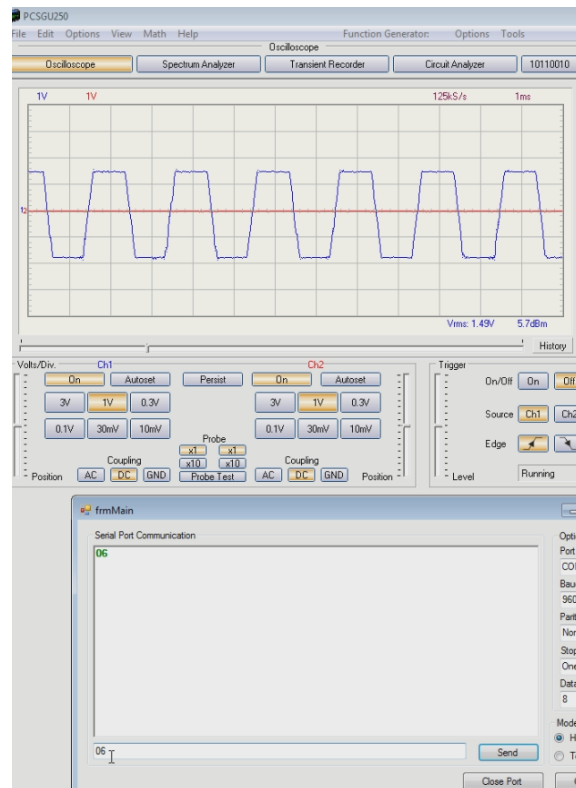
De volgende module die besproken wordt, is de Space. De Space is verantwoordelijk voor het simuleren van de lucht waar het signaal doorheen gaat. Zoals beschreven staat in hoofdstuk 7.7.1.2, heeft een space-DSP twee belangrijke eigenschappen. De space moet een vertraging en een afzwakking simuleren. In Figuur 25 en Figuur 26 staan deze twee verschillende

eigenschappen uitgebeeld. In Figuur 25 wordt de hexadecimale waarde⁴ van 01 naar de DSP gestuurd. Deze verstuurde waarde stelt een korte te simuleren afstand voor. Vervolgens wordt er een kort signaal gegenereerd door middel van de oscilloscoop. De oscilloscoop wordt vervolgens ook gebruikt om deze signalen op te vangen en in een grafiek, zoals te zien in Figuur 25, te plaatsten. De rode lijn in deze grafiek is het signaal dat gegenereerd is door de oscilloscoop. De blauwe lijn is het signaal dat door de space-DSP wordt gesimuleerd. Zoals te zien is in dit figuur, is het blauwe signaal afgezwakt en vertraagd ten opzicht van het rode signaal. In het volgende figuur, Figuur 26, is de waarde 7F naar de DSP gestuurd. Deze waarde stelt een grote te simuleren afstand voor. Na dezelfde handelingen te hebben uitgevoerd als in Figuur 25 is te zien dat de blauwe lijn in Figuur 26 veel meer is afgezwakt en vertraagd ten opzicht van de blauwe lijn uit Figuur 25. Dit komt doordat het signaal een grotere afstand heeft afgelegd en hierdoor dus langer onderweg is.

9.1.2.3 DETECTABLE OBJECT



Figuur 27: Detectable Object: kleine RCS



Figuur 28: Detectable Object: grote RCS

De Detectable Object module is verantwoordelijk voor het simuleren van een detecteerbaar object zoals een vliegtuig. Zoals beschreven in hoofdstuk 7.7.1.3, heeft een Detectable Object maar één belangrijke eigenschap, de RCS⁵. In Figuur 27 wordt de hexadecimale waarde⁶ van 02 naar de DSP gestuurd. Deze waarde stelt een kleine RCS voor. Op dezelfde DSP wordt ook een constant signaal gegenereerd door middel van een oscilloscoop. Dit signaal is niet te zien in de grafieken van Figuur 27 en Figuur 28, maar tijdens het uitvoeren van dit onderzoek is het signaal niet veranderd. In Figuur 28 wordt vervolgens de hexadecimale waarde van 06 naar de

⁴ Deze waarde stelt geen afstand of tijd voor, maar het aantal stappen dat er vertraagt en afgezwakt moet worden.

⁵ Zie begrippenlijst

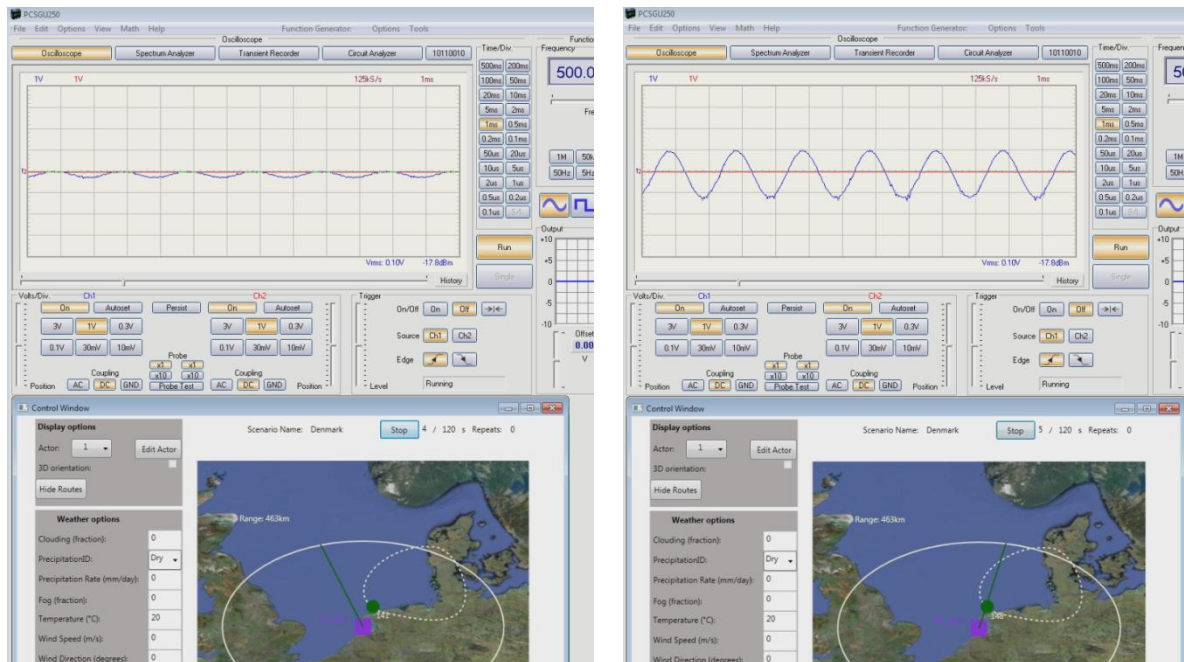
⁶ Deze waarde stelt geen echte RCS voor, maar het aantal stappen dat versterkt moet worden.

DSP gestuurd. Deze waarde stelt een grote RCS voor. Als vervolgens de signalen uit Figuur 27 en Figuur 28 met elkaar worden vergeleken, dan is er een duidelijk verschil te zien in de verschillende signalen. Het signaal met een kleine RCS, is kleiner dan het signaal met een grote RCS. Hierdoor zal de signaalsterkte in het geval van een grote RCS, groter zijn dan de signaalsterkte van de kleinere RCS.

9.1.2.4 RADARRETURN

Zoals beschreven staat in hoofdstuk 7.7.1.1, heeft de RadarReturn als verantwoordelijkheid om het ontvangen radarsignalen terug te zenden naar de simulator.

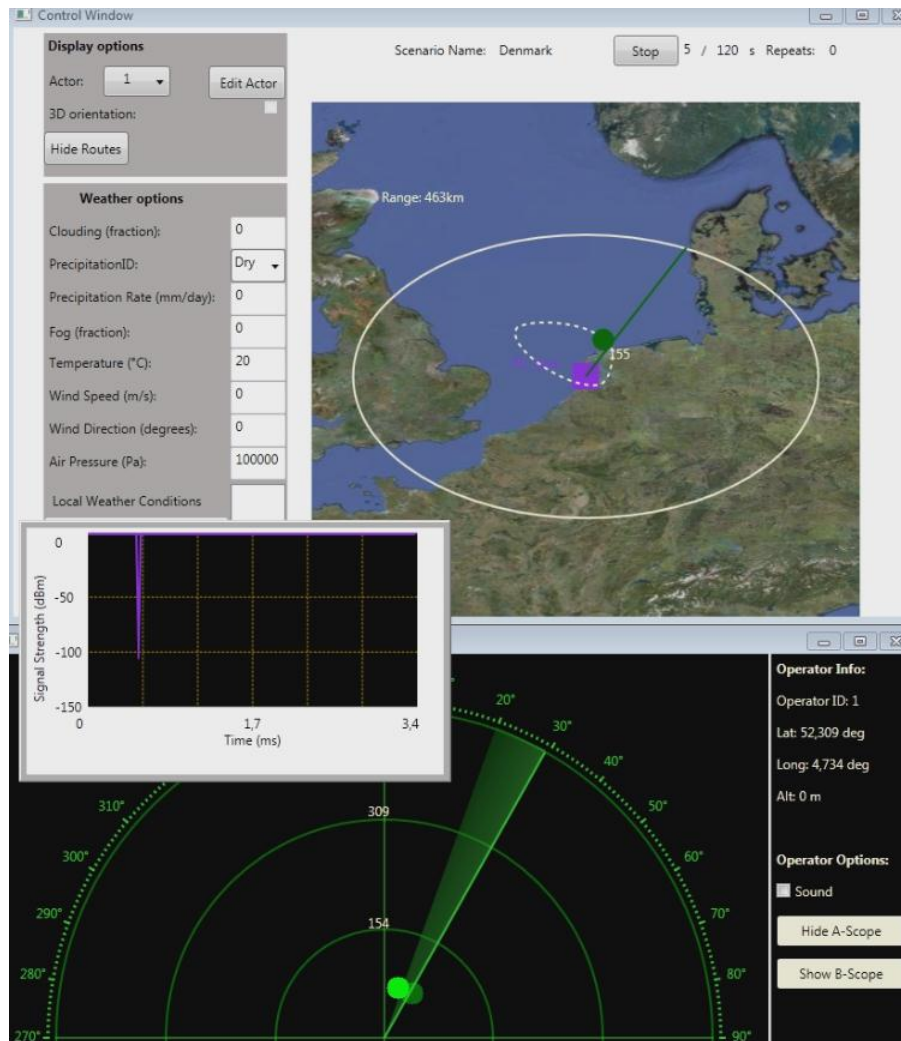
9.2 WERKING MET DE SIMULATOR



Figuur 29: RCS toepassen op een gedetecteerd vliegtuig

Nu de alle resultaten van de losse modules bekend zijn, wordt het tijd om alles aan de simulator te koppelen. Om te beginnen wordt de “Detectable Object” DSP aan de simulator gekoppeld. Zoals te zien in Figuur 29 wordt het vliegtuig zonder problemen gedetecteerd en wordt de RCS hierop aangepast. In dit figuur is ook te zien dat het signaal, net zoals in hoofdstuk 9.1.2.3, continu is. Dit komt omdat er nog geen DSP op de simulator is aangesloten die een signaal genereert. In plaats van de DSP, verzorgt de oscilloscoop nu nog het signaal.

De volgende stap is om de radar DSP aan de simulator te koppelen. Zoals beschreven staat in hoofdstuk 7.7.1.1 bestaat de radar DSP uit zowel de “RadarSend” en de “RadarReturn”. Tussen de “RadarSend” en de “RadarReturn” wordt er vervolgens een “Detectable Object” DSP gekoppeld. Hierdoor kunnen er drie resultaten in één keer binnen komen. Ten eerste kan er getest worden of het versturen van een radarsignaal werkt. Daarnaast kan er getest worden of de koppeling van de radar naar een object werkt. Als laatste kan er ook gekeken worden of de “RadarReturn” gegevens terug stuurt naar de simulator.



Figuur 30: Radar simulator werkend op DSP kracht

Het resultaat van deze test is terug te vinden in Figuur 30. Onderin dit figuur is een venster te zien met een groen radarscherm. Dit is het scherm waarop de gedetecteerde objecten worden geplaatst. Als er een object gedetecteerd is, dan wordt er een groene stip getekend op de gedetecteerde positie. Zoals te zien is in het venster, is er een groene stip getekend. Hieruit kunnen we concluderen dat object gedetecteerd is. Omdat de Space DSP ontbreekt, wordt de afzwakking van het signaal op een eenvoudige, softwarematig manier gecorrigeerd. Hierdoor is het niet mogelijk om met zekerheid te zeggen of het terug kerkende signaal klopt. Ook de afstand waarop het vliegtuig vliegt kan, door het ontbreken van een space-DSP, niet worden berekend.

9.2.1 SIGNAAL STERKTE

Uit de test van Figuur 30 kwam een interessant resultaat naar voren. Het blijkt dat de sterkte van het signaal niet altijd gelijk blijft, ook al staat het gedetecteerde object wel stil. Het verschil in sterkte is niet groot, maar het was net genoeg om op te vallen. Als er gekeken wordt naar het verschil in signaalsterkte, dan blijkt het verschil maximaal 4 dBm te zijn.

9.2.2 SIMULATOR SNELHEID

Tijdens het uitvoeren van de testen met de DSP's aan de simulator gekoppeld, kwam er naar voren dat de simulator te veel aanvragen voor het zenden van een signaal stuurde. Hierdoor

kwam er te veel data bij de simulator binnen en dit zorgde ervoor dat de simulator erg traag draaide.

Hierom werden er twee verschillende maatregelen genomen om dit te verhelpen. Als eerste werd de variabelen met om de hoeveel tijd de simulator een radarsignaal zend, verhoogd van 7 naar 10 milliseconde. Als tweede werd het maximaal aantal keren dan een signaal een object mag raken, op één gezet. Dit betekent dat wanneer een object al gedetecteerd is, hier niet opnieuw een radar signaal heen wordt gestuurd. Deze maatregelen hadden een positief effect op de performance van de simulator.

9.2.3 COMPLETE SYSTEEM

Tijdens het schrijven van deze scriptie, het nog niet gelukt is om een compleet signaal te verzenden en vervolgens te ontvangen. Door de vele uitdagingen en leermomenten, is deze stage opdracht achter gaan lopen op schema. Hierdoor was het, door de verloren tijd, niet meer mogelijk om het complete systeem te testen, kalibreren en eventueel aan te passen. Om deze reden zullen er dan ook geen resultaten beschikbaar zijn over het complete systeem.

9.2.4 BENODIGDE DSP'S

Om een radarsignaal te volledig simuleren, zijn er minimaal drie verschillende DSP's nodig. Zo heb je een DSP voor de radar, space en het object. Mocht de huidige code worden uitgebreid met meer functionaliteit, dan zou het aantal DSP's ook toenemen. De toenamen zal in de meeste gevallen per twee DSP's moeten gebeuren. Omdat het signaal altijd voor een deel door de lucht verplaatst, zal er altijd een Space object nodig zijn.

10 CONCLUSIES

In hoofdstuk 9.2.3 is er beschreven dat, tijdens het schrijven van deze scriptie, het nog niet gelukt is om een compleet signaal te verzenden en vervolgens te ontvangen. Door de vele uitdagingen en leermomenten, is deze stage opdracht achter gaan lopen op schema. Hierdoor was het, door de verloren tijd, niet meer mogelijk om het complete systeem te testen, kalibreren en eventueel aan te passen. Om deze reden zal deze conclusie voornamelijk gebaseerd zijn op de resultaten van de lossen onderdelen. De huidige onderzoeksvragen gaan echter uit van een compleet systeem. Om deze onderzoeksvragen toch te kunnen beantwoorden, gaat deze conclusie naast de resultaten van de lossen onderdelen er ook vanuit dat het mogelijk is om de losse onderdelen goed te kunnen koppelen of dat het eventuele probleem eenvoudig op te lossen is.

10.1 HOE KAN EEN DSP WORDEN GEBRUIKT OM EEN ZO REALISTISCHE MOGELIJK SIGNAAL TE SIMULEREN, ZONDER DAT DE SNELHEID VAN DE SIMULATOR TE NADELIG WORDT BEÏNVLOED?

In hoofdstuk 5.1 is de hoofdvraag hoe een DSP kan worden gebruikt om een zo realistische mogelijk signaal te simuleren, zonder dat de snelheid van de simulator te nadelig wordt beïnvloed. Deze vraag zal hier in dit subhoofdstuk beantwoord en verder toegelicht worden.

10.1.1 IS HET SIGNAAL REALISTISCHER GEWORDEN?

Er zijn veel verschillen tussen de softwarematige en hardware matige signalen. Het softwarematige signaal dat wordt gebruikt in de huidige simulator bestaat uit één piek. Dit is eigenlijk één moment opname van het hele verzonden signaal. Zoals beschreven staat in hoofdstuk 9.1.2.1, is het met behulp van de DSP's wel mogelijk om een bepaald signaal te genereren. Omdat het DSP signaal uit meer dan alleen één piek bestaat, is er meer informatie over het signaal beschikbaar. Hierdoor kunnen de aanpassingen aan het signaal dan ook realistischer worden toegepast.

Zoals beschreven in hoofdstuk 9.2.1, zijn DSP gesimuleerde signalen nooit hetzelfde. Er zitten altijd kleine veranderingen in de signalen die op papier hetzelfde horen te zijn. Dit wordt bijvoorbeeld veroorzaakt door interferentie op de lijnen waar het radarsignaal overheen gaat. Dit lijkt misschien een ongewenst effect, maar dat maakt het signaal juist realistischer. Als een echt radarsignaal de lucht in wordt gestuurd, dan zullen de omstandigheden ook nooit hetzelfde zijn. Zo is bijvoorbeeld mogelijk dat er precies een vogel door het radarsignaal heen vliegt. Hierdoor zal het signaal dat terug komt bij de radar net iets anders zijn, dan als er niks doorheen was gevlogen.

De DSP gebruikt relatief grote stappen om de verschillende eigenschappen te veranderen. Deze stappen zijn merkbaar in het resultaat wat uiteindelijk terugkeert bij de radar. In het echt zijn deze stappen ook niet waarneembaar en hierdoor heeft het een negatieve invloed op het realisme van het radarsignaal. Deze stappen kunnen eventueel later kleiner worden gemaakt door de code van die draait op de DSP's aan te passen.

10.1.2 WEERSVARIABLEN

De onderzoeksvraag over de weersvariabelen kan niet beantwoord worden. Het is in de huidige code niet mogelijk om de weersvariabelen realtime aan te kunnen passen. Het realtime aanpassen van deze waarde is nog niet geïmplementeerd, omdat het werkend krijgen van een complete DSP simulatie, zoals te lezen in hoofdstuk 9.2.3, voorrang had. Er bestaat echter nog wel een mogelijkheid om dit handmatig te aan te passen, maar hier zijn nooit echte resultaten uit gekomen.

10.1.3 IS SNELHEID NIET TE NADELIG BEÏNVLOED?

De snelheid van de simulator is een belangrijk aspect. Zoals beschreven staat in hoofdstuk 4 moet de simulator realtime draaien omdat er anders timing problemen kunnen ontstaan. In hoofdstuk 9.2.2 staat beschreven dat de snelheid van het versturen van een radarsignaal voor de zekerheid van 7 naar 10 milliseconde is verhoogd. Door het verhogen van deze tijd zou er gesteld kunnen worden dat de snelheid is vertraagt. Maar, zoals beschreven staat in hoofdstuk 8.1.3, is een verhoging van enkele milliseconden niet tot nauwelijks merkbaar. Ook op de snelheid van de simulator heeft het geen invloed, omdat de simulator nu alleen iets minder vaak een radarsignaal verstuurd. Er kan dus gesteld worden dat de snelheid niet te nadelig is beïnvloed.

10.2 IN HOEVERRE KOMT HET SIGNAAL IN DE BUURT VAN DE WERKELIJKHEID?

Als er gekeken wordt naar hoofdstuk 5.2.1, dan zijn alleen de must haves geïmplementeerd. Van deze must haves heeft voornamelijk “Een bekende Radar Cross Section gebruiken om het te genereren signaal van de weerkaatsing te berekenen met behulp van een DSP.” Veel effect gehad op de werkelijkheid van de simulator. Hierdoor wordt het namelijk mogelijk om verschillende type vliegtuigen te onderscheiden van elkaar.

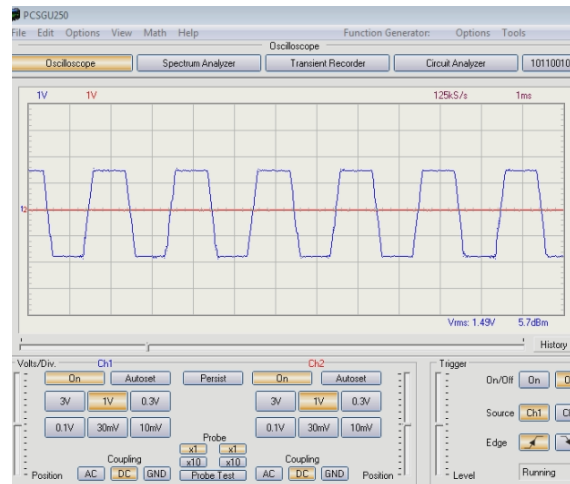
Als er wordt gekeken naar de nog niet geïmplementeerde producten, dan zal voornamelijk noise en het weer een grote rol spelen in het realistischer maken van een signaal. Het weer zal naar alle waarschijnlijkheid de grootste invloed hebben. Het weer zal ook het duidelijkst zichtbaar zijn op de radar. Een goede weercel kan namelijk evengoed zichtbaar zijn als vliegtuig.

10.3 VOOR- EN NADELEN

10.3.1 VOORDELEN

Zoals ook in hoofdstuk 10.1.1 is het beschreven, zorgt de DSP voor meer realisme in radarsignaal. Realisme is, zoals te lezen in hoofdstuk 2, een belangrijk onderdeel van de simulator. Hierdoor kan er wel gesteld worden dat meer realisme het belangrijkste voordeel is. Een ander voordeel wat de DSP met zich mee brengt is, dat de verschillende modules los te controleren zijn. Zo kan er bijvoorbeeld halverwege ergens een oscilloscoop geplaatst en uitgelezen worden. Hierdoor wordt het mogelijk om na te gaan hoe een radarsignaal er bij een vliegtuig eruit ziet.

10.3.2 NADELEN



Figuur 31: Hardware limieten

De gebruikte DSP's hebben echter ook een aantal nadelen. Eén van de grootste nadelen zijn de hardware limieten. In Figuur 31 wordt er een sinussignaal door de DSP heen gestuurd. Wat opvalt is dat het geen sinus(golf)signaal, maar bloksignaal lijkt te zien. Dit komt omdat de DAC en ADC een bepaald bereik hebben. Mocht je buiten dit bereik komen, dan verlies je de informatie van dat deel.

10.4 IS HET MOGELIJK OM EEN DSP TE GEBRUIKEN BINNEN HET EWSIM PROJECT?

Gezien de hele scriptie, kan er wel geconcludeerd worden dat het mogelijk is om een DSP te gebruiken binnen het EWSim project. Maar hiervoor zullen er nog wel aanpassingen nodig zijn. Zo is het, tijdens het schrijven van deze scriptie, nog niet mogelijk om een compleet signaal te versturen en te ontvangen. Verder wordt er nog geen gebruik gemaakt van de volledige capaciteit van de DSP. Mochten er weercellen, ruis of jamming technieken gesimuleerd gaan worden, dan zal het voordeel van de DSP alleen maar groter worden.

11 Aanbevelingen

Zoals beschreven staat in hoofdstuk 8.2 zijn er tijdens het uitvoeren van deze stageopdracht verschillende problemen ontstaan, die nog extra aandacht vragen. Het is daarom belangrijk om nog eens goed te kijken naar de communicatie tussen de DSP en de communicatie module. Dit probleem is wellicht opgelost door kortere draadjes te gebruiken, maar meerdere personen gaven het advies mee om nog eens goed naar dit probleem te kijken. Waarschijnlijk werd dit probleem veroorzaakt door een elektrisch probleem.

Een ander punt wat wel tijd kan kosten is het vastmaken van de communicatie module aan de DSP. De korte lengte van de draadjes zorgen ervoor dat ze snel losschieten. Een draadje hoeft maar half los te zitten en er bestaat een kans dat het draadje geen contact meer maakt. Er zijn meerdere oplossingen voor dit probleem, maar solderen is waarschijnlijk de eenvoudigste manier.

Voor de communicatie naar de DSP wordt er momenteel nog geen gebruik gemaakt van protocol om de data naar de DSP te sturen. Hierdoor is het alleen mogelijk om één bepaalde variabele te veranderen in de DSP. Sommige variabele zullen dus nooit veranderd kunnen worden door de simulator. Voor de huidige objecten is dat niet een heel groot probleem. Maar als er later DSP weercellen komen of de mogelijkheid om noise te creëren in een Space-object, dan wordt dit gemis een probleem. Een eenvoudig eigen bedacht protocol is vaak al genoeg.

In hoofdstuk 5.2.1 bevindt zich de Moscow van deze stageopdracht. De punten in deze Moscow zijn echter nog lang niet allemaal gemaakt. De meeste de punten die in de Moscow staan, zorgen ervoor dat het realisme in het systeem merkbaar wordt verhoogd.

12 Evaluatie van de procesgang

12.1 HOE BEN IK BEGONNEN?

Een van de eerste dingen waarmee ik begon was het kiezen van mijn afstudeeropdracht. Als stagiaire ben je namelijk redelijk vrij om zelf een onderwerp te kiezen dat past bij een bepaald project. Omdat ik had aangegeven dat ik als onderwerp graag met hardware zou willen werken, stelde ze twee mogelijke apparaten voor. Het eerste apparaat was een FPGA en andere een DSP. Al snel kwam er een onderzoek van een vorige projectteam naar voren. In dit onderzoek werd afgeraden om de FPGA te gebruiken die op het bedrijf ligt. Hierom is er uiteindelijk gekozen voor de DSP. Vervolgens ben ik begonnen met het lezen van handleidingen over de DSP. Hierdoor was ik goed voorbereid voor het programmeren op de DSP.

12.2 HOE ZAG HET VERVOLG ERUIT?

Vervolgens ben ik test programma's gaan schrijven voor de DSP. Deze programma's waren bedoeld om te ondervinden hoe de DSP nou precies werkt en wat hij allemaal kan. Hierdoor stuiten ik op een hardware probleem. Het probleem was dat de analoog-digitaal converter te traag is. Hij heeft een latency van ongeveer 0.6 milliseconde en radarbegrippen is dit extreem veel. Uiteindelijk bleek dit probleem niet heel groot te zien. Het was in de simulator namelijk relatief eenvoudig om rekening te houden met deze latency.

De volgende stap in het programmeren van de DSP's is om het te laten communiceren met de computer. De enige manier om via data te kunnen communiceren met de huidige DSP is via een SPI-lijn (Serial Peripheral Interface Bus). De communicatie tussen de computer en de SPI-lijn, werd geregeld door een Vinculum-II chip van FTDI.

De communicatie tussen de DSP en computer heeft voor veel verschillende problemen gezorgd. Het grootste probleem was dat de data die de DSP opvangt is niet het zelfde als de data die wordt verstuurd (zie ook hoofdstuk 8.2). Deze problemen zorgen er uiteindelijk voor dat het schema dusdanig achter liep, dat ik net wat tijd te kort kwam om het hele systeem via DSP's te kunnen testen.

12.3 WAT HEB IK GELEERD?

Ik tijdens deze stage veel verschillende dingen geleerd. Tijdens deze stage stond ik voor een aantal grote uitdaging. Deze uitdagingen hebben ervoor gezorgd dat deze stage een leerzame tijd was.

Tijdens deze stage heb ik het meeste geleerd van hardware en alles wat erbij hoort. Tijdens de opleiding technische informatica zijn wel een aantal hardware project langsgekomen. Maar al deze hardware was van te voren al getest en de eventuele problemen al bekend. Tijdens deze stage had ik dat voordeel niet. Ik moest zelf alles uitzoeken en uitproberen. Dit laat echt een hele andere kant zien van "technische informatica". Ik heb tijdens deze dan ook veel leert over hardware. Zo weet ik nu hoe moeilijk het is om in te schatten hoelang een hardware opdracht duurt. Problemen kunnen uit de meest onverwachtse hoeken komen. Ook zijn deze problemen vaak erg specifiek en levert het internet vaak weinig resultaten op. Ook op het gebied van de standaard dingen, zoals datasheets, protocollen en timing, heb erg veel geleerd. Om sommige problemen te vinden heb ik datasheets vaak helemaal moeten leren begrijpen.

Een ander punt waar ik veel over heb geleerd is de verwerking van audio- en radarsignalen. Omdat ik tijdens de stage veel in aanrakingen ben gekomen met deze signalen, weet ik nu veel beter hoe ze werken en bewerkt worden.

12.4 WAT VOND IK LASTIG?

Zoals in hoofdstuk 12.1 beschreven staat had je als stagiaire veel vrijheid om zelf een onderwerp te kiezen. In eerste instantie vond ik het moeilijk om met de gegeven vrijheid om te gaan. Ik had verwacht dat deze vrijheid veel minder was en dat er al afstudeeropdracht klaar lagen.

Verder vond ik het lastig om met de hardware te werken. Door de verschillende problemen had ik vaak niet het idee dat ik vooruit kwam. Als dit vervolgens een paar keer achterelkaar gebeurt, dan begin je wel aan jezelf te twijfelen. Ik heb soms wel het gevoel gehad dat het nooit zou lukken, omdat het te moeilijk was.

12.5 WAT VOND IK VAN HET EINDRESULTAAT?

Ik heb uiteindelijk wel gemende gevoelens aan het eindresultaat overgehouden. Aan de ene kant ben erg blij met het opgeleverde resultaat, gezien de problemen die ik ben tegen gekomen. Maar aan de andere kant had ik ook graag een systeem gezien wat uiteindelijk gewoon goed werkt.

Ik ben wel erg trots op de producten die afzonderlijk goed werken. Ik vind heel mooi om te zien, dat je als aan de ene kant een audiosignaal plaatst, dat dit signaal dan bewerkt aan de andere kant eruit komt.

12.6 WAT VOND IK VAN DE STAGE?

Ik vond dit een erg leuke stageopdracht. De opdracht was niet gemakkelijk en je moest er echt wat voor doen om een resultaat te behalen. Als je dan een resultaat behaald, dan ben extra blij met dat resultaat. Ik heb altijd al het idee gehad dat ik later wat meer wil doen richting de hardware kant. Maar na deze stage weet ik het wel zeker dat daar mijn ding ligt.

Ik vond het verder ook een leerzame stage. Ik heb veel geleerd over hardware dat nooit voorbij is gekomen op de opleiding. Ik denk ook dat deze stage een belangrijke stap is naar een baan in deze richting.

13 BRONVERMELDINGEN

Antiradarsneeuw. (2011, december 26). Opgeroepen op november 16, 2012, van Wikipedia:
<http://nl.wikipedia.org/wiki/Antiradarsneeuw>

Digitale signaalprocessor. (2011, December 22). Opgeroepen op Februari 16, 2012, van Wikipedia:
http://nl.wikipedia.org/wiki/Digitale_signaalprocessor

Field-programmable gate array. (2012, januari 20). Opgeroepen op november 16, 2012, van Wikipedia: http://nl.wikipedia.org/wiki/Field-programmable_gate_array

Scrum (softwareontwikkelmethode). (2012, April 30). Opgeroepen op Mei 15, 2012, van Wikipedia:
[http://nl.wikipedia.org/wiki/Scrum_\(softwareontwikkelmethode\)](http://nl.wikipedia.org/wiki/Scrum_(softwareontwikkelmethode))

Smith, S. W. (1997). *C versus Assembly*. Opgeroepen op Mei 23, 2012, van dspguide (The Scientist and Engineer's Guide to Digital Signal Processing): <http://www.dspguide.com/ch28/5.htm>

TeTech. (sd). *begrippenlijst Antennes en Propagatie*. Opgeroepen op november 16, 2012, van TeTech:
<http://www.tetech.nl/begripAP.htm>

Bijlage A PLAN VAN AANPAK



Scriptie

Onderzoeken hoe een DSP kan worden gebruikt om een zo realistisch mogelijk signaal te simuleren.

Roy de kok(1546848)
Technische informatica

1 Inhoudsopgave

2	INLEIDING	A-3
3	BEGRIPPENLIJST	V
4	AANLEIDING TOT DE OPDRACHT	A-5
5	CONTEXT WAARBINNEN HET AFSTUDEERPROJECT WORDT UITGEVOERD	A-7
5.1	BESCHRIJVING VAN DE AFDELING.....	A-7
5.2	RELATIE MET ANDERE PROJECTEN.....	A-7
5.3	RELATIE MET DE ORGANISATIE	1
6	PROBLEEMSTELLING	A-8
7	DOELSTELLING	A-8
8	AFSTUDEEROPDRACHT	A-9
9	OP TE LEVEREN PRODUCTEN.....	A-10
9.1	MOSCOW	11
9.1.1	<i>Must have</i>	11
9.1.2	<i>Should have</i>	11
9.1.3	<i>Could have</i>	12
9.1.4	<i>Would have</i>	12
10	PROJECT DEFINITIE	A-11
10.1	SCOPE	A-11
10.2	AANNAMES	A-11
10.3	RANDVOORWAARDEN	A-11
11	PROJECTACTIVITEITEN	A-12
11.1	FASE 1: DE EERSTE TWEE WEKEN	A-12
11.2	FASE 2: ONDERZOEK DSP (WEEK 3-5)	A-12
11.3	FASE 3: IMPLEMENTATIE DSP (WEEK 6-15)	A-12
11.4	FASE 4: AFRONDINGS- EN PRESENTATIEFASE (WEEK 16-19)	A-12
12	DE TE GEBRUIKEN METHODEN/TECHNIEKEN	A-13
13	KWALITEITSBEWAKING.....	A-13
14	RISICO'S	A-14
15	BEDRIJFS-/PERSOONSgegevens	A-15
15.1	BEDRIJFSgegevens	A-15
15.2	PERSOONSgegevens	A-15
16	BIJLAGE.....	FOUT! BLADWIJZER NIET GEDEFINIEERD.
16.1	BIJLAGE A: STROKENPLANNING	FOUT! BLADWIJZER NIET GEDEFINIEERD.

2 INLEIDING

Ik ben een vierdejaars student Technische Informatica aan de Hogeschool Utrecht. Mijn stage vindt plaats bij het bedrijf NCIM-groep in Leidschendam. NCIM-groep is een detacheringsbedrijf dat naast detachering ook innoveert. Deze innoverende projecten worden meestal uitgevoerd door afstudeerders en professionals die op dit moment geen detacheringsovername hebben.

NCIM-groep richt zich voornamelijk op technisch complexe projecten in de volgende sectoren:

- Defensie en Veiligheid;
- Energie en Utilities;
- Telecom, Transport en Media;
- Technische Automatisering.

Het project waar ik mijn afstudeerstage zal uitvoeren behoort tot een innoverend project. Deze projecten worden meestal uitgevoerd op het hoofdkantoor in Leidschendam.

De duur van mijn afstudeerstage is drie en halve maand en loopt van 13 februari 2012 tot en met 29 mei 2012. De inhoud van dit plan van aanpak bestaat uit een plan voor het succesvol afronden van mijn afstudeeropdracht bij NCIM-groep.

In dit plan van aanpak bevindt zich achtergrond informatie en uitgebreide informatie over de opdracht. Verder komt de planning die tijdens dit project gebruikt gaat worden en overige zaken die relatie hebben met dit project aan bod.

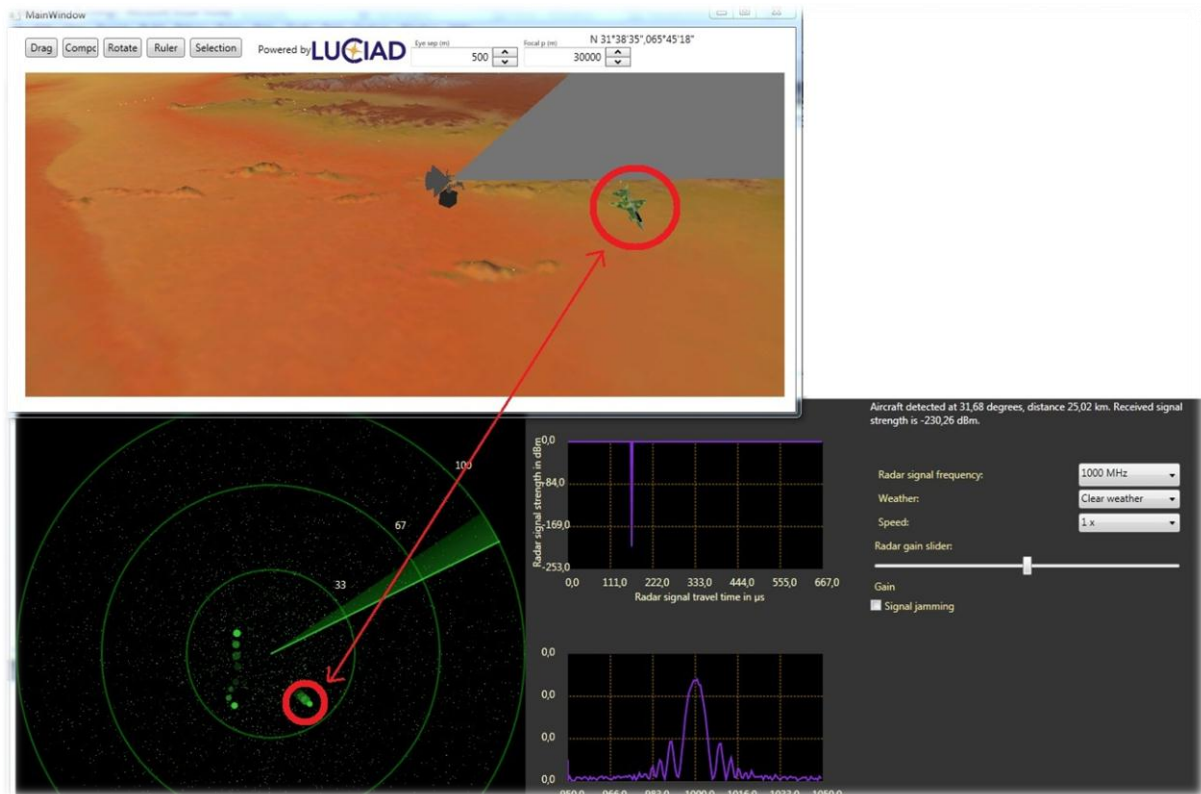
3 BEGRIPPENLIJST

In deze begrippenlijst leg ik kort enkele begrippen uit die mogelijk niet bekend zijn bij de lezer van dit verslag. Belangrijke begrippen die voorkomen en mogelijk onduidelijk zijn:

- **DSP**: Een Digitale signaalprocessor(DSP) is een microprocessor die speciaal ontworpen is voor het bewerken van (continue) digitale signalen. (Digitale signaalprocessor, 2011)
- **FPGA**: Een field-programmable gate array(FPGA) is een geïntegreerde schakeling bestaande uit programmeerbare logische componenten. Deze logische componenten kunnen geprogrammeerd worden met behulp van logische functies zoals AND, XOR, enzovoorts. Deze functies kunnen bijvoorbeeld decoders of eenvoudige wiskundige functies zijn. (Field-programmable gate array, 2012)
- **Radar Cross Section**: De radar cross section is een maat voor de mate waarin een obstakel, object of constructie de radiogolven reflecteert in dezelfde richting als waar zij vandaan komen. Dit is vooral van belang bij RADAR. (TeTech)
- **Chaff**: Chaff is een antiradartechniek waarbij door militaire vliegtuigen of schepen grote hoeveelheden foliereepjes uitgeworpen worden, die gemaakt zijn van tin, aluminium of een kunststof met een opgedampte laag metaal. (Antiradarsnieuw, 2011)
- **Jamming**: Jamming is het sturen van stoorsignalen om zo het radarbeeld negatief te beïnvloeden.
- **EWSim**: EWSim is de afkorting voor het overkoepelende project van deze afstudeerstage genaamd Electronic Warfare simulator.

4 AANLEIDING TOT DE OPDRACHT

NCIM-groep is momenteel bezig met het maken van een Electronic Warfare simulator. Deze simulator, te zien in Figuur 32, is bedoeld om een radarsysteem te simuleren. De simulator moet het mogelijk maken om radaroperators zo realistisch mogelijk te kunnen opleiden. Dit wordt bereikt doordat je realistische scenario's kan inladen. In deze scenario's kan je alles instellen, van het weer tot en met objecten die zich kunnen verplaatsen. Al deze instellingen zijn nodig om zo goed mogelijk de realiteit te kunnen simuleren. Doordat de simulator op de werkelijkheid lijkt, is het mogelijk om radaroperators zo goed mogelijk te kunnen opleiden aan de hand van verschillende gesimuleerde omstandigheden.



Figuur 32: Electronic Warfare Simulator

De radar is een belangrijk onderdeel in het realistischer maken van de Electronic Warfare simulator. Want de radar speelt namelijk een belangrijke rol in de hedendaagse oorlogsvoering en dus ook in de Electronic warfare simulator. De radar is (feitelijk) de enige manier om een vijandig toestel bijtijds op te merken. Zoals je misschien wel eens in films of documentaires hebt gezien, werd de radar al gebruikt in de tweede wereld oorlog. Maar waarom is deze “oude” techniek nou zo belangrijk voor de Electronic Warfare Simulator? Het antwoord op deze vraag ligt voornamelijk bij de vijand. De vijand, die niet gezien wil worden op de radar, gebruikt steeds meer technieken om niet of zo min mogelijk op het radarbeeld te verschijnen of zelfs het radarbeeld te verstoren. Deze technieken zijn bijvoorbeeld toegepast op stealthvliegtuigen van het type F117 of B2. Het is ook mogelijk om een radarbeeld te verstoren door bijvoorbeeld Chaff of andere verstoringsmethodes. Het wordt hierdoor steeds moeilijker voor bijvoorbeeld een radaroperator om vijandige toestellen tijdig te zien aankomen. De radaroperator heeft een belangrijke taak als het gaat om de veiligheid van een schip of een land. Het is daarom belangrijk dat de radaroperator zo realistisch mogelijk wordt opgeleid voor diverse scenario's. Omdat dit

in veel gevallen onmogelijk is in de echte wereld, moet er gebruik gemaakt worden van een radarsimulator.

Een radar werkt doordat er een radarsignaal de lucht in wordt gestuurd. Dit signaal botst tegen een object op en reflecteert onder andere terug naar de radar. Echter de meeste radar simulatoren maken geen gebruik van gesimuleerde radarsignalen. Doordat deze simulatoren precies weten waar de objecten zich bevinden, worden deze op de goede locatie op de radar getoond. Echter klopt dit niet met de werkelijkheid, omdat er in de werkelijkheid het signaal verandert aan de hand van verschillende oorzaken. Zo zorgt een regenbui ervoor dat het signaal wordt afgebogen. Dit kan er voor zorgen dat bijvoorbeeld een vliegtuig op de radar ergens anders lijkt te vliegen dan in de (gesimuleerde) werkelijkheid. De Electronic Warfare Simulator is hierin anders. De simulator gaat in plaats van de positie te simuleren, de radarsignalen simuleren. Hierdoor kan er beter rekening gehouden worden met de omgevingsvariabelen zoals het weer en andere objecten. Voor de radaroperator het beeld zou er zo realistischer eruit komen te zien. Hierdoor kan er dus beter getraind worden op het gebruik van een radar.

Realistischere berekeningen vergen ook meer rekenkracht. Voor een computer zullen deze berekeningen te veel rekenkracht kosten. Dit komt omdat er veel verschillende factoren een rol spelen tijdens het verplaatsen van het signaal. Zo geeft zonnig weer een ander resultaat terug dan regen. Ook blijft het signaal nooit één enkel signaal. Het signaal kaatst vaak af in meerdere richtingen. Die signalen kunnen vervolgens ook weer afkaatsen in verschillende richtingen. Verder geeft bijvoorbeeld een stad meer ruis op het signaal, dan een dorp. Het genereren van die signalen kost te veel rekenkracht voor één of zelfs meerdere computers. Hierom moest er dus een alternatief komen dat het genereren van die signalen wel aan zou kunnen zonder dat het ten koste gaat van de snelheid. Omdat het om een digitaal signaal gaat is een DSP waarschijnlijk de beste optie voor deze taak. Een DSP is een microprocessor die speciaal ontworpen is voor het bewerken van (continue) digitale signalen. Om deze reden zou een DSP ideaal moeten zijn om hem te koppelen aan het Electronic Warfare project.

5 CONTEXT WAARBINNEN HET AFSTUDEERPROJECT WORDT UITGEVOERD

In dit hoofdstuk wordt een uitleg gegeven over de context waarbinnen het afstudeerproject zal plaatsvinden. Als eerste komt er een beschrijving van de afdeling waar de opdracht wordt uitgevoerd. Hierin wordt uitgelegd waar het project wordt uitgevoerd en met welke omstandigheden. Hierna zal het gaan over de relatie van het afstudeerproject met andere projecten. Als laatste gaat het ook nog over de relatie van het afstudeerproject met de afdeling en de organisatie.

5.1 BESCHRIJVING VAN DE AFDELING

Binnen het bedrijf heb je twee verschillende soorten afdelingen. Eén van die afdelingen heet “operations”. Zoals de naam misschien al doet vermoeden, zijn deze mensen voornamelijk bij de klanten aan het werk. De andere afdeling is bezig met innoverende projecten. Het doel van deze projecten is om kennis op te doen en prototypes of demo's te maken. De tweede afdeling bevindt zich voornamelijk op het hoofdkantoor van de NCIM-groep.

Dit afstudeerproject wordt gezien als een innoverend project en zal dus op het hoofdkantoor van NCIM-groep worden uitgevoerd. Op deze locatie wordt naast dit afstudeerproject ook het overkoepelde project, waar dit afstudeerproject een onderdeel van is, uitgevoerd. Aan dit project werken nog vijf andere personen mee.

5.2 RELATIE MET ANDERE PROJECTEN

Het afstudeerproject uit dit verslag heeft een relatie met een ander project. In dit subhoofdstuk volgt er een uitleg over de relaties met dit afstudeerproject.

De afstudeeropdracht, beschreven in dit verslag, maakt deel uit van een overkoepelend project, genaamd Electronic Warfare Simulator. De afstudeeropdracht moet met behulp van een DSP ondersteuning bieden aan het overkoepelende Electronic Warfare Simulator project. Dit project heeft als doel om met een zo realistisch mogelijke radarsimulator, radaroperators te kunnen opleiden. Dit wordt bereikt door het simuleren van de radarsignalen om zo te bepalen wat er op het radar scherm moet komen te staan. Naast de operator user interface voor de radaroperator in opleiding, komt er ook nog een teacher user interface. Hier kan de leraar, die de radaroperator opleidt, alles in de simulator instellen. Zo kan bijvoorbeeld het weer en de andere objecten worden aangepast of worden toegevoegd.

5.3 RELATIE MET DE ORGANISATIE

Het afstudeerproject heeft twee belangrijke relaties met het bedrijf. Als eerste is dit project bedoeld als onderzoek naar nieuwe technieken. Zeker het gebruik van DSP technieken is voor het bedrijf nieuw. Met een project zoals dit wil het bedrijf innoveren om zo mee te kunnen concurreren met andere bedrijven.

Als tweede wil het bedrijf ook aan hun klanten laten zien wat ze kunnen. Zonder werkende demo's zullen klanten minder snel met het bedrijf in zee gaan. Een werkende demo geeft de klant vertrouwen dat een bedrijf die bepaalde taak kan uitvoeren.

6 PROBLEEMSTELLING

Het afstudeerbedrijf, NCIM-Groep, wil graag voorop lopen met de huidige technieken. Om dit doel te bereiken moet het bedrijf gaan innoveren en hierdoor kennis opdoen van nieuwe technieken of de huidige technieken verder te ontwikkelen.

NCIM-Groep innoveert voornamelijk op technieken waar vraag naar is. Ze ontdekten dat er nog geen realistische simulator was om radaroperators goed te kunnen opleiden. Er zijn wel een aantal simulatoren, maar hierbij is het realisme niet hoog genoeg om de radaroperators goed op te kunnen leiden. Deze simulatoren kunnen de kleine verschillen in het signaal vaak niet goed genoeg simuleren. Deze kleine verschillen zijn vaak essentieel om het radarsignaal goed te kunnen interpreteren. Dit heeft als gevolg dat de radaroperators belangrijke details alleen maar in het echt kunnen leren. In het echt is het echter moeilijk en soms onmogelijk op verschillende scenario's te leren. De radaroperator is hierdoor niet bekend met sommige situaties die hij tegen kan komen. Dit kan een groot risico vormen voor de veiligheid bijvoorbeeld op een schip of zelf voor een land, zoals Nederland.

Om dit probleem op te vangen werken ze bij NCIM-groep aan een realistische radarsimulator. De simulator gaat de radarsignalen simuleren zodat deze zo realistisch mogelijk wordt. Om een signaal te simuleren, zal je het signaal eerst moeten genereren. Het genereren van een digitaal signaal is voor een computer vaak te intensief. Hij heeft niet de goede componenten om deze zonder een te grote snelheidsverlies een digitaal signaal te genereren. Zeker niet als er ook nog rekening moet worden gehouden met overige elementen, zoals ruis. Dit wordt helemaal een probleem als een signaal meerdere keren opsplijt en meerdere signalen op het zelfde moment verspreid. Hierdoor zal er gebruik gemaakt moeten worden van een andere techniek om dit probleem af te vangen.

7 DOELSTELLING

Het realistischere genereren van signalen zonder dat de snelheid te nadelig wordt beïnvloed is met een gewone computer onmogelijk. Hiervoor moet er gekeken worden naar een alternatieve techniek, zoals in het geval van deze stage een DSP. De doelstelling van deze stage is daarom ook om uit te vinden hoe een DSP kan worden gebruikt om een zo realistisch mogelijk signaal te simuleren, zonder dat de snelheid van de simulator te nadelig wordt beïnvloed.

8 AFSTUDEEROPDRACHT

In dit hoofdstuk wordt er kernachtig beschreven wat de afstudeeropdracht precies inhoud en welke onderzoeksvragen er beantwoord moeten worden.

De afstudeeropdracht heeft een relatie met de huidige Electronic Warfare simulator. Deze simulator moet hardware matig versneld worden. In eerste instantie was het plan om een FPGA(field-programmable gate array) te gebruiken om berekeningen, die nodig zijn om het radarsignaal te simuleren, parallel te kunnen uitvoeren en zo tijdswinst en effectief meer rekenkracht te verkrijgen. Echter uit een eerder uitgevoerd onderzoek op NCIM, bleek dat er geen snelheidswinst te behalen is op de aanwezige FPGA. Dit komt voornamelijk omdat de huidige FPGA te weinig capaciteit heeft. Dit heeft onder andere als gevolg dat de communicatie tussen de computer en de FPGA te traag is. Een oplossing zou zijn om een betere FPGA aan te schaffen, maar de product owner, Toon Akkermans, vindt hiervoor de prioriteit te laag en ziet liever dat er naar andere technieken wordt gezocht. Mede door de wensen van de product owner en de resultaten van het eerdere onderzoek is uiteindelijk besloten om de FPGA links te laten liggen.

Uiteindelijk is gekozen om een DSP te gaan gebruiken. De verwachting is dat de DSP sneller signalen kan genereren dan een computer. Het genereren van signalen is belangrijk voor het realisme in het systeem. Niet alleen het signaal kan aan de hand van de omgevingsfactoren, realistischer bewerkt worden. Maar ook voor de radaroperator, die het signaal moet interpreteren, ziet het signaal er realistischer eruit. Verder wilde de product owner dat het bedrijf meer ervaring krijgt met DSP's. Om deze redenen zal het gebruik van een DSP goed moeten passen in dit project.

Gedurende het onderzoek hoop ik de volgende onderzoeksvragen te kunnen beantwoorden:

- Hoe kan een DSP worden gebruikt om een zo realistische mogelijk signaal te simuleren, zonder dat de snelheid van de simulator te nadelig wordt beïnvloed?
 - Hoe heeft het genereren van het teruggekaatste radarsignaal invloed gehad op het realistischer maken van het signaal?
 - Hoe heeft de weersvariabelen, zoals de luchtdruk, temperatuur en luchtdichtheid, invloed gehad op het realistischer maken van het signaal?
 - In hoeverre heeft de DSP invloed gehad op de snelheid van de simulatie?
- In hoeverre komt het gesimuleerde signaal in de buurt van de werkelijkheid?
 - Welke verschillende omgevingsfactoren uit de Moscow zijn er geïmplementeerd?
 - Welke van deze factoren zijn belangrijk geweest voor het realistischer maken van het radarsignaal?
 - Welke omgevingsfactoren uit de Moscow zijn nog niet geïmplementeerd?
 - In hoeverre zouden deze factoren invloed hebben op het realistischer maken van het radarsignaal?
- Is het mogelijk om een DSP binnen het EWSim project te gebruiken voor het genereren van de signalen?
- Wat zijn de voor- en nadelen van een DSP?
 - Wat zijn de voordelen?
 - Wat zijn de nadelen?

9 OP TE LEVEREN PRODUCTEN

In dit hoofdstuk volgt er een uitleg over de producten die naar verwachting worden opgeleverd. Omdat het bedrijf via scrum werkt, en dus dat er elke week opnieuw gekeken wordt naar wat er moet worden gemaakt, is het moeilijk om de op te leveren producten echt in detail te bespreken. Om deze reden is er een Moscow tabel ingevoegd met de mogelijk op te leveren producten.

9.1 MOSCOW

9.1.1 *MUST HAVE*

- Een bekende Radar Cross Section gebruiken om het te genereren signaal van de weerkaatsing te berekenen met behulp van een DSP.
- Met behulp van een DSP een signaal te genereren die is beïnvloed door de luchtdruk, temperatuur, luchtdichtheid (uniform over aardoppervlak).
- EWSim code aanpassen zodat een DSP kan worden geïmplementeerd.
- Invloed van neerslag, bewolking en onweer (uniform over het aardoppervlak) op het radarsignaal genereren met behulp van een DSP. Aanname: weer heeft uniforme Radar Cross Section.
- Documentatie van de code.
- Korte handleiding.

9.1.2 *SHOULD HAVE*

- Invloed op de radarontvangst van Chaff genereren met behulp van een DSP.
- Invloed op de radarontvangst van Windmolens genereren met behulp van een DSP.
- Non-uniforme weersinvloeden met eigen Radar Cross section genereren met behulp van een DSP.
- De DSP gebruiken om ruis in het signaal te genereren. bijvoorbeeld ruis dat wordt veroorzaakt door een stad.

9.1.3 *COULD HAVE*

- Invloed op de radarontvangst van Jamming genereren met behulp van een DSP.
- Invloed op de radarontvangst van het terrein genereren met behulp van een DSP.
- Invloed van overige objecten, zoals bomen en huizen, genereren met behulp van een DSP.

9.1.4 *WOULD HAVE*

- De Radar Cross section berekenen met behulp van een DSP/computer(bij de overige punten ga ik er van uit dat de Radar Cross section al bekend is.)

Zoals te zien in de Moscow is de DSP en de radarsignalen een erg belangrijk onderwerp. Daarom is met de hulp van een DSP de radarsignalen te genereren, het hoofdproduct dat wordt opgeleverd.

Naast de DSP is het voor het bedrijf ook erg belangrijk dat alles goed gedocumenteerd wordt. Dit project is een innoverend project. Dit wil zeggen dat het bedrijf dit project gebruikt om ervaring op te doen. Als dan de documentatie ontbreekt, dan is het eigenlijk allemaal voor niets gedaan. Daarom staat dit bij de must have.

10 PROJECT DEFINITIE

10.1 SCOPE

Deze afstudeeropdracht omvat alleen het ontwikkelen van de code voor de DSP. De DSP wordt alleen ter ondersteuning van het overkoepelende project gebruikt. Het is tijdens deze stage niet bedoeld om veel aan de huidige Electronic Warfare simulator te veranderen. Kleine aanpassingen zodat de DSP gebruikt kan worden in de simulator horen hier echter wel bij.

Dit project bestaat uit een tijdsbestek van drie en halve maand en loopt van 13 februari 2012 tot en met 29 mei 2012.

10.2 AANNAMES

- Er is een DSP aanwezig om de applicatie op te kunnen ontwikkelen.
- Er is een beschikbare en geschikte werkplek om mijn project te kunnen uitvoeren.
- Er is duidelijke documentatie aanwezig m.b.t. het overkoepelende project.
- Er is genoeg tijd om dit project uit te voeren.
- De applicatie waarmee de DSP gaat communiceren is volledig genoeg zodat de DSP kan worden geïmplementeerd.

10.3 RANDVOORWAARDEN

- Het project moet voltooid zijn voor/op de gestelde einddatum.
- Het project moet een werkend prototype opleveren.
- Het project moet minimaal aan de Must Haves van de Moscow regels voldoen.
- Tijdens het project moet er zoveel mogelijk gebruik gemaakt worden van programma's die beschikbaar zijn voor Microsoft Windows.

11 PROJECTACTIVITEITEN

Dit hoofdstuk bevat een puntsgewijze opsomming van de uit te voeren activiteiten op volgorde gezet binnen de fasering van ons onderzoek. Deze fasering bestaat uit vier fasen.

Het project bestaat uit scrum sessies van een week. Elke week wordt er bepaald wat er die week af moet worden gemaakt. Parallel met het project loopt het maken van de scriptie. Het is door scrum onmogelijk om een precieze planning te maken, daarom zal hier een planning volgen hoe het ongeveer eruit kan komen te zien.

Voor een strokenplanning van deze uit te voeren activiteiten, zie hoofdstuk **Fout! Verwijzingsbron niet gevonden..**

De volgende belangrijke activiteiten moeten tijdens het onderzoek plaatsvinden en staan gesorteerd op chronologische volgorde, van aan het begin van het project tot het einde van het project.

11.1 FASE 1: DE EERSTE TWEE WEKEN

- Met alle stakeholders spreken om meer te weten te komen over het onderzoek
- Onderzoeksvragen bedenken
- Plan van aanpak opstellen en inleveren (uiterste inlever datum: 16-03-2012)
- Globale informatie lezen over de werking van radars.
- Globale indruk krijgen over de huidige software van het overkoepelende project.

11.2 FASE 2: ONDERZOEK DSP (WEEK 3-5)

- Globale informatie lezen over de werking van radars.
- Globale indruk krijgen over de huidige software van het overkoepelende project.
- Globale indruk krijgen van de gebruikte wiskunde.
- Onderzoeken hoe DSP's werken en waarvoor ze het beste kunnen worden gebruikt.
- Onderzoeken hoe DSP's geprogrammeerd moeten worden.
- Begin maken van de scriptie.

11.3 FASE 3: IMPLEMENTATIE DSP (WEEK 6-15)

- Het programmeren van de DSP's aan de hand van de genoemde Moscow regels die hier in dit verslag zijn genoemd.
- Verwerken van de uitkomsten van de deelvragen in de scriptie.
- Uiterste Inleverdatum van de Scriptie is op 29-05-2012 om 12.00u.

11.4 FASE 4: AFRONDINGS- EN PRESENTATIEFASE (WEEK 16-19)

- Uitloophase van het project.
- Bedrijfsbeoordeling inleveren op 08-06-2012.
- Presentatie maken. (afstudeerzitting: 11-6 t/m 22-6 2012)

12 DE TE GEBRUIKEN METHODEN/TECHNIEKEN

Om het project succesvol te laten slagen wordt er gebruik gemaakt van de volgende methoden en technieken:

- Een digitale signaalprocessor(DSP).
- De DSP maakt gebruik van C/C++ of Assembler.
- De DSP moet gebruik maken van wiskundige formules om een signaal te kunnen genereren.
- C#/.net wordt gebruikt in het overkoepelende project.

13 KWALITEITSBEWAKING

Binnen het onderzoek zal de kwaliteit bewaakt worden door middel van voortgangsgesprekken met de begeleider, het beoordelen van elkaars code en documenten en het stellen van kwaliteitseisen aan het eindproduct.

De kwaliteitsbewaking voor de documenten vindt plaats doordat externe mensen, zoals de stagebegeleider en overige teamleden, de documenten door te laten lezen om zo de kwaliteit te waarborgen.

Voor het beheren van de broncode gebruiken wij het versiebeheersysteem SVN. Hiermee is het gemakkelijk om elkaars wijzigingen in te zien en te beoordelen. Ook maakt dit het mogelijk om oude code terug te halen indien nodig. Door de wijze waarop wij dit doen kunnen wij elkaars code gemakkelijk doorlezen en elkaar daar dan indien nodig op aanspreken.

De kwaliteitseisen voor de software zullen vooral gebaseerd zijn op een testsuite die controleert of er in de simulatie niets is verandert.

14 RISICO'S

In de oriënterende fase heb ik een inventarisatie van de risico's gemaakt die tijdens ons onderzoek een rol zouden kunnen spelen. In de volgende tabel staan deze vermeld, met de bijbehorende prioriteit. Daarnaast hebben we mogelijke tegenmaatregelen bedacht die we kunnen inzetten in het geval dat het betreffende risico optreedt.

Status	Beschrijving risico	Kans	Impact	Prioriteit	Proximity	Maatregel
open	Uitval door ziekte	0,1	2	0,2	Hele project	Taken overdragen aan andere projectleden
open	Technische complicaties	0,15	3	0,45	Ontwikkeling	Alternatief gebruiken/Technische ondersteuning zoeken.
open	Uitloop planning	0,1	2	0,2	Ontwikkeling	Extra tijd erin steken, ruimer plannen
open	Te weinig kennis vooraf	0,1	3	0,3	Ontwikkeling	Vooraf goede kennis opdoen via bijvoorbeeld internet en/of boeken.
open	Product voldoet niet aan de eisen	0,2	4	0,8	Hele project	ik kan dit verminderen door wekelijks contact te houden met de begeleider.
Open	Het overkoepelde project is nog niet ver genoeg. Hierdoor kunnen niet alle deelvragen beantwoord worden.	0,1	3	0,3	Hele project	Genoeg status updates krijgen van het groten project zodat ik op tijd kan bepalen wanneer bepaalde dingen niet op tijd af komen. Mocht dit het geval zijn, dan moet ik dit op tijd opmerken en aanpassingen maken.

15 BEDRIJFS-/PERSOONSgegevens

15.1 BEDRIJFSgegevens

NCIM GROEP HOOFDKANTOOR

Overgoo 11

2266 JZ Leidschendam

Tel: +31 (0)70 - 3501554

Fax: +31 (0)70 - 3545515

NCIM GROEP EINDHOVEN

RUYSDAELBAAN 31B

5613 DX EINDHOVEN

TEL: +31 (0)40 - 2433488

FAX: +31 (0)70 - 3545515

15.2 PERSOONSgegevens

STAGEBEGELEIDER EN TECHNISCH AANSPREEKPUNT

BAS KNOPPER

MAIL: B.KNOPPER@NCIM.NL

TEL: +31 (0)6 - 24384288

ALGEMEEN AANSPREEKPUNT

Debby Voordenhout

Mail: D.Voordenhout@ncim.nl

Tel: +31 (0)6 - 51123131

STAGIAIRE

Roy de Kok

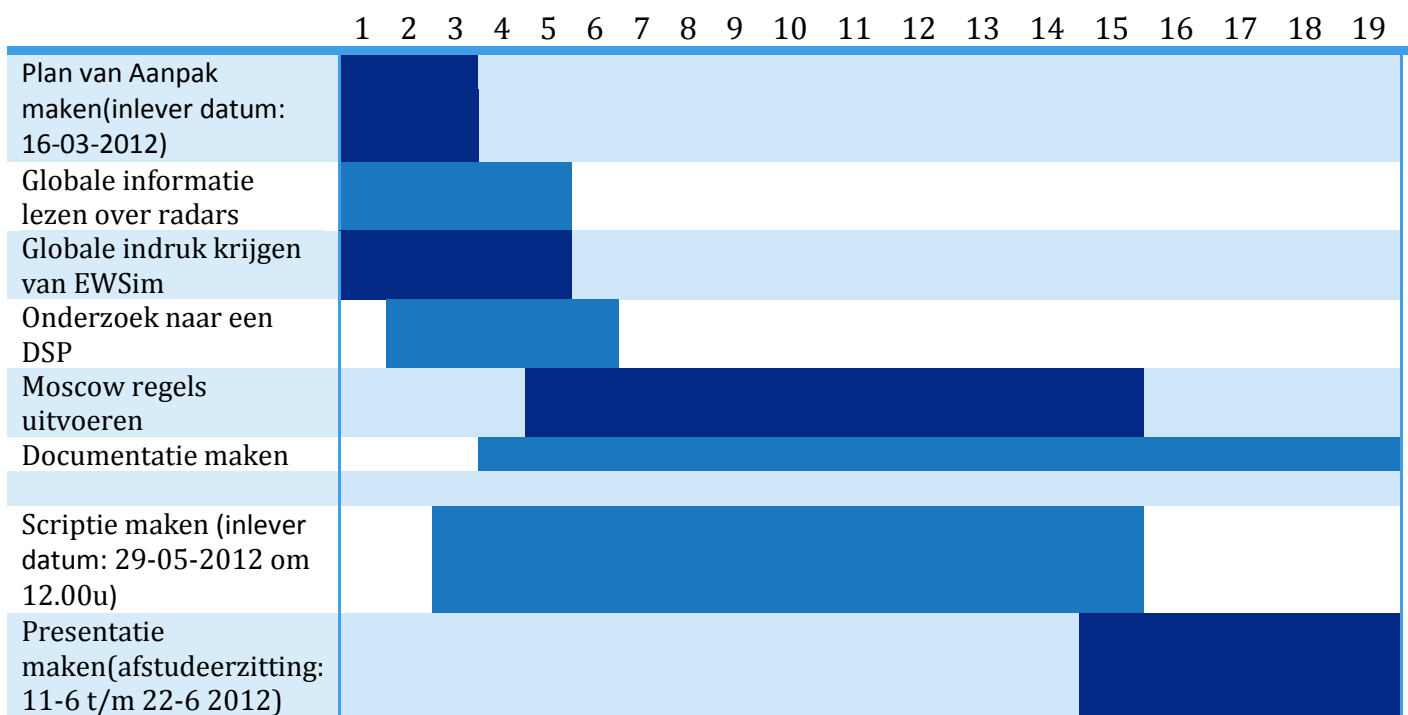
Student nr.: 1546848

Mail: R.de.Kok@ncim.nl

Tel: +31 (0)6 - 46557426

16 BIJLAGE

16.1 BIJLAGE A: STROKENPLANNING



Bijlage B EVALUATIE VAN HET EIGEN FUNCTIONEREN

Bijlage C HARDWARE INTERFACE

RADARSEND (COMPUTER → DSP)

Naam	ID	Min waarde	Max. waarde	Waarde eenheid	Waarde bereik	Datatype	Woordlengte	Uitlijning	Communicatie weg	Opmerking
Synchronisatie burst	N.V.T.	N.V.T.	N.V.T.	N.V.T.	N.V.T.	N.V.T.	N.V.T.	N.V.T.	D → D	Een synchronisatie burst om de te kunnen corrigeren voor de vertraging van de hele schakeling
Frequentie	1	1	96	KHz	[1...96]	Integer	24	Rechts	C → D	De frequentie van het te versturen signaal. Dit is maximaal 96, omdat dit de helft van 192 is. 192 is de maximale sample rate van de analoog digitaal convertor.
Gain	2	0	50	dBm	[0...50]	Integer	24	Rechts	C → D	Afhankelijk van het type radar of de “sideloop” van de radar.
Amplitude	3					Integer	24	Rechts	C → D	
Burst time	4	0,1	10	ms	[1...100]	Integer	24	Rechts	C → D	De tijd van het te generen signaal
Time Next burst	5	1	20	ms	[1...20]	Integer	24	Rechts	C → D	De tijd die het duurt voordat er een nieuw signaal wordt verstuurd. Let op: Dit is inclusief de burst time!
Enable	6	0	1	N.V.T.	[0,1]	Integer	24	Rechts	C → D	Hiermee kan je de DSP laten draaien. Als hij niet geënabled is, stuurt hij ook geen signaal. Mocht hij geënabled worden, dan stuurt hij een eerst een synchronisatie burst.

OBJECT (COMPUTER → DSP)

Naam	ID	Min waarde	Max. waarde	Waarde eenheid	Waarde bereik	Datatype	Woordlengte	Uitlijning	Communicatie weg	Opmerking
RCS	1	0	100	N.V.T.	[0...100]	Integer	24	Rechts	C → D	De RCS waarde van het object. Dit moet vermenigvuldigd worden met het signaal. 0 is niet zichtbaar en 100 is goed zichtbaar
Enable	2	0	1	N.V.T.	[0,1]	Integer	24	Rechts	C → D	Hiermee kan je aangeven of het object gedetecteerd wordt. Als hij gedisabled is, dan stuurt hij het binnen gekomen signaal niet door.

SPACE (COMPUTER → DSP)

Naam	ID	Min waarde	Max. waarde	Waarde eenheid	Waarde bereik	Datatype	Woordlengte	Uitlijning	Communicatie weg	Opmerking		
RCS	1	-6	-13	dBm	[0...70]	Integer	24	Rechts	C → D	De RCS waarde van de verzwakking door de lucht.		
Delay	2	0	300	Samples	[0...300]	Integer	24	Rechts	C → D	De delay van de space, gemeten in het aantal samples.		
Noise	3	0	100	N.V.T.	[0...100]	Integer	24	Rechts	C → D	De hoeveelheid noise die de space moet creëren. Bij 0 wordt er geen noise toegepast.		
Enable + Early return	4	0	1	N.V.T.	[0,1,2,3]	Integer	24	Rechts	C → D	Als Early return bit is geactiveerd, dan stuurt hij om een ingestelde tijd een afgezwakt signaal terug over het andere uitgangskanaal. Hierdoor wordt het mogelijk om een signaal om een bepaalde tijd terug te laten sturen naar een ander object zoals de “radar return”. Hierdoor is het bijvoorbeeld mogelijk om “noise” te creëren van een regenbui.		
											Bit 1	Bit 0 (LSB)
											Actief bit	Early return bit
Early return time	5	reserved	for	future	use					Om de hoeveelheid tijd hij een early return moet versturen.		
Early return amount	6	reserved	for	future	use					Hoe vaak hij een early return moet sturen.		

RADAR RETURN(DSP → COMPUTER)

Naam	ID	Min waarde	Max. waarde	Waarde eenheid	Waarde bereik	Datatype	Woordlengte	Uitlijning	Communicatie weg	Opmerking
Synchronisatie	N.V.T.	N.V.T.	N.V.T.	N.V.T.	N.V.T.	N.V.T.	N.V.T.	N.V.T.	D → D	Het terug krijgen van de synchronisatie burst om de te kunnen corrigeren voor de vertraging van de hele schakeling.
Return Signal strength	1	0	150	dBm	[0...150]	Integer	24	Rechts	D → C	Het signaal dat wordt ontvangen, wordt doorgegeven aan de computer. Dit gebeurt in combinatie met “return signal strength” en “return signal time”
Return Signal strength	2	0	150	samples	[0...700]	Integer	24	Rechts	D → C	Het signaal dat wordt ontvangen, wordt doorgegeven aan de computer. Dit gebeurt in combinatie met “return signal strength” en “return signal time”
Enable	3	0	1	N.V.T.	[0,1]	Integer	24	Rechts	C → D	Als hij gedisable is, dan kijkt hij niet naar de ontvangen signalen.
Aaaa	4	0	?	?	?	Integer	24	Rechts	C → D	

