

**RADAR**

# Verkeersteller

**Door: Louis Van Lommel**

**Hogeschoolpromoter: Pedro Wyns**

**Bedrijfspromoter: Paul Dierckx**

**PBA Elektronica-ICT**  
Afstudeerrichting Elektronica

**Bachelorproef voorgedragen tot het behalen  
van de graad en het diploma van bachelor**

**Campus De Nayer**

# Voorwoord

Al sinds klein af aan ben ik bezig met het maken van dingen, zo maakte ik als kind knex auto's die konden rijden op basis van een opgeblazen ballon en raketten die ik de lucht in kon sturen door op een plastic fles te springen. Thuis sta ik dus sinds kinds af aan bekend als de "handy man".

Ik begon mijn middelbare schoolcarrière in Industriële wetenschappen en rolde in mijn derde jaar de Elektriciteit-Elektronica wereld in. Daar maakte ik kennis met mijn eerste microcontrollers, FPGA's, voedingen... Na mijn middelbare schoolcarrière was het voor mij duidelijk dat ik verder wou studeren in de richting van Hardware-Elektronica, daar is uiteindelijk de richting Embedded-Hardware uitgekomen.

In mijn tweede jaar Embedded-Hardware waren mijn interesses al snel duidelijk, FPGA, RF en PCB-design. Ik kan met trots zeggen dat ik in mijn hoge schoolcarrière individueel deze interesses heb kunnen uitbouwen zo maakte ik in het tweede jaar een draadloze deurbel met een PCB antenne en in het derde jaar mijn eigen FPGA (Field Programmable Gate Array) development board. Nu kan ik mijn hoge schoolcarrière afsluiten met een bachelorproef die deze 3 interesses combineert.

Tot slot wil ik graag enkele personen bedanken voor hun hulp en begeleiding. Graag bedank ik mijn docenten, medestudenten, hogeschool promotor: Pedro Wyns, bedrijfspromotor: Paul Dierckx en mijn collega's: Bram Kennis en Jonas Aertgeerts.

# Samenvatting

Het doel van deze bachelorproef is om een prototype verkeersteller te ontwikkelen dat in staat is om de snelheid en het type van voertuigen te registreren in beide rijrichtingen op basis van radar en FPGA technologie.

Deze verkeersteller is een project van de lokale/federale politiediensten dat opgedragen is aan het stagebedrijf OAC Electronics te Herentals.

Er is onderzoek gedaan naar de verschillende modulatietechnieken die toegepast worden in een radarsysteem en wat de voordelen zijn van een FPGA tegenover de klassieke microcontroller.

Uit onderzoek is gebleken dat de beste modulatietechniek voor het monitoren van verkeer Frequency Modulated Continuous Wave-radar (FMCW-radar) is, hierbij is het mogelijk om de afstand, snelheid, rijrichting en plaats van verschillende objecten te bepalen op een snelle maar zeer complexe manier.

Vervolgens is hiervoor een prototype PCB ontwikkeld met de nodige hardware. Het volledige bestukingsproces van de PCB is ter plaatse gebeurd volgens de regels van het vak met de EMS-productielijn van OAC Electronics.

Tot slot is er onderzoek gedaan naar de nodige software om een FMCW-radar systeem softwarematig te kunnen realiseren, hierbij komen verschillende wiskundige algoritmes zoals de Fast Fourier Transform (FFT) en dergelijke aan bod.

# Inhoudstafel

|   |           |
|---|-----------|
| <b>1. INLEIDING</b>                                 | <b>9</b>  |
| <b>2. OAC-ELECTRONICS</b>                           | <b>10</b> |
| <b>3. OPDRACHT</b>                                  | <b>11</b> |
| 3.1. Doelstelling                                   | 11        |
| 3.2. Onderzoek                                      | 11        |
| 3.3. Uitdagingen                                    | 11        |
| <b>4. RADAR: THEORIE</b>                            | <b>12</b> |
| 4.1. Introductie                                    | 12        |
| 4.1.1. Elektromagnetische golven                    | 12        |
| 4.2. Radar  | 14        |
| 4.2.1. Doppler (snelheid)                           | 14        |
| 4.2.2. IQ-signaal (Richting)                        | 15        |
| 4.2.3. Modulatie (Afstand)                          | 16        |
| 4.2.3.1. <i>Continuous Wave</i>                     | 16        |
| 4.2.3.2. <i>Chirping</i>                            | 16        |
| 4.2.3.3. <i>Frequency Modulated Continuous Wave</i> | 18        |
| <b>5. PROCESSING: THEORIE</b>                       | <b>20</b> |
| 5.1. Digital Signal Processing                      | 20        |
| 5.2. Sampling ADC                                   | 21        |
| 5.3. Fourier transformation                         | 22        |
| 5.3.1. Discrete Fourier Transform (DFT)             | 22        |
| 5.3.2. Fast Fourier Transform (FFT)                 | 23        |
| 5.4. FMCW-RADAR Digital Signal Processing           | 24        |
| <b>6. RADAR: PRAKTIJK</b>                           | <b>28</b> |
| 6.1. RSP1-evaluatie bordje                          | 28        |
| 6.2. Custom amplifier                               | 29        |
| 6.3. InnoSent IVQ-3005                              | 30        |
| <b>7. HARDWARE</b>                                  | <b>32</b> |
| 7.1. Electrical data                                | 32        |
| 7.2. Blokschema                                     | 33        |
| 7.3. Schema's                                       | 33        |
| 7.3.1. Blokschema                                   | 33        |
| 7.3.2. Power  | 34        |
| 7.3.2.1. <i>Regelbare uitgangsspanning</i>          | 34        |
| 7.3.2.2. <i>Output Inductor selection</i>           | 35        |
| 7.3.2.3. <i>Output capacitor selection</i>          | 35        |
| 7.3.2.4. <i>PGOOD</i>                               | 35        |
| 7.3.2.5. <i>Automatische power-up sequence</i>      | 36        |
| 7.3.3. FPGA Power                                   | 37        |

|  |    |
|--|----|
| 7.3.3.1. Recommended Operating Conditions                                | 37 |
| 7.3.3.2. Power-On/Off Power Supply Sequencing                            | 37 |
| 7.3.3.3. Required PCB Capacitor Quantities per Device: Spartan-7 Devices | 38 |
| 7.3.4. FTDI  | 39 |
| 7.3.4.1. JTAG  | 39 |
| 7.3.4.2. UART  | 39 |
| 7.3.5. FPGA Configuratie   | 40 |
| 7.3.5.1. Bootmode  | 40 |
| 7.3.5.2. JTAG  | 40 |
| 7.3.5.3. Flash Memory  | 41 |
| 7.3.5.4. Configuration Banks Voltage Select                              | 41 |
| 7.3.5.5. PROGRAM   | 41 |
| 7.3.6. FPGA BANK 14  | 42 |
| 7.3.6.1. 100Mhz Clock  | 42 |
| 7.3.6.2. PUDC  | 42 |
| 7.3.7. FPGA BANK 15  | 43 |
| 7.3.7.1. ADC   | 43 |
| 7.3.7.2. ADC channels  | 43 |
| 7.3.7.3. Analoge Inputs  | 43 |
| 7.3.7.4. Reference spanning  | 43 |
| 7.3.7.5. Differential sampling   | 44 |
| 7.3.7.6. SPI   | 44 |
| 7.3.8. FPGA BANK 34  | 45 |
| 7.3.8.1. Dip Switches  | 45 |
| 7.3.8.2. LEDS  | 45 |
| 7.3.8.3. Testpoints  | 45 |
| 7.3.9. Radar Module  | 46 |
| 7.3.9.1. Power Supply IVQ-3005   | 46 |
| 7.3.9.2. InnoSent IVQ-3005 Connector                                     | 46 |
| 7.3.9.3. Analoge uitgangen IVQ-3005                                      | 47 |
| 7.3.9.4. Analoge Ingangen IVQ-3005                                       | 47 |
| 7.3.9.5. Digitale uitgangen IVQ-3005                                     | 48 |
| 7.3.9.6. Digitale Ingangen IVQ-3005                                      | 48 |
| 7.3.9.7. Digitale I/O IVQ-3005   | 48 |
| 7.4. PCB   | 49 |
| 7.4.1. Layer Stack-up:   | 49 |
| 7.4.2. Design Rules  | 50 |
| 7.4.3. Layout: Voeding   | 51 |
| 7.4.4. Layout: FTDI  | 51 |
| 7.4.5. Layout: FPGA  | 52 |
| 7.4.5.1. Signal Layers FPGA  | 52 |
| 7.4.5.2. Routing FPGA  | 53 |
| 7.4.5.3. Decoupling  | 56 |
| 7.4.6. Via stitching   | 57 |
| 7.4.7. Design Rule Check   | 58 |
| 7.4.8. Mechanical  | 59 |
| 7.4.9. PCB Panel   | 59 |
| 7.4.10. Bill Of Materials  | 59 |

|   |           |
|---|-----------|
| 7.4.11. Production Variant                | 59        |
| <b>8. EMS PRODUCTIE</b>                   | <b>60</b> |
| 8.1.1. OAC ERP-codes                      | 60        |
| 8.1.2. Componenten kast                   | 61        |
| 8.1.3. Feeders                            | 62        |
| 8.1.4. Stencil-machine                    | 63        |
| 8.1.5. Pick & Place                       | 64        |
| 8.1.6. Automatic Optical Inspection (AOI) | 65        |
| 8.1.7. Oven & Vapor Phase                 | 66        |
| 8.1.8. Eindproduct                        | 67        |
| 8.1.9. Testing                            | 68        |
| 8.2. Behuizing                            | 69        |
| 8.2.1. Materiaal                          | 69        |
| 8.2.2. Mechanisch                         | 69        |
| 8.2.3. Storing                            | 69        |
| <b>9. SOFTWARE</b>                        | <b>70</b> |
| 9.1. Xilinx                               | 70        |
| 9.2. Vivado                               | 70        |
| 9.2.1. XDC-file                           | 70        |
| 9.2.2. FTDI to JTAG programmer            | 70        |
| 9.2.3. IP-catalog                         | 71        |
| 9.2.4. Softcore (Microblaze)              | 72        |
| 9.2.5. GPIO                               | 73        |
| 9.2.6. UART                               | 73        |
| 9.2.7. FLASH                              | 74        |
| 9.2.8. Analooq Digitaal Converter (XADC)  | 75        |
| 9.2.9. Fast Fourier Transform (XFFT)      | 78        |
| 9.2.10. Direct Memory Access (DMA)        | 78        |
| 9.3. Vitis                                | 79        |
| 9.3.1. ADC testcode                       | 79        |
| 9.3.2. FFT+DMA testcode                   | 80        |
| 9.3.3. CW-RADAR software                  | 82        |
| <b>10. BESLUIT</b>                        | <b>83</b> |
| <b>11. LIJST MET AFBEELDINGEN</b>         | <b>84</b> |
| <b>12. LITERATUURLIJST</b>                | <b>87</b> |
| <b>13. BIJLAGEN:</b>                      | <b>88</b> |
| 13.1. Bijlagen Custom Amplifier K-LC6:    | 88        |
| 13.2. BIJLAGEN RADAR BOARD:               | 96        |
| 13.3. BIJLAGEN VIVADO:                    | 116       |
| 13.4. BIJLAGEN VITIS:                     | 117       |

# Opstartverslag

| Gegevens student(en)   |  |
|--|--|
| <b>Student 1:</b> Louis Van Lommel<br><b>Opleiding:</b> Elektronica-ICT<br><b>GSM:</b> 0471375863                                      | <b>Student 2:</b> /<br><b>Opleiding:</b> /<br><b>GSM:</b> /  |
| Gegevens bedrijf / instelling  |  |
| <b>Naam:</b> Oac Electronics<br><b>Adres:</b> Dikberd 34unit 13b<br><b>Tel:</b> +32 14 55 83 83  |  |
| Gegevens bedrijfspromoter  |  |
| <b>Naam:</b> Paul Dierckx<br><b>Functie:</b> Klik of tik om tekst in te voeren.<br><b>Afdeling:</b> Klik of tik om tekst in te voeren. | <b>Tel/GSM:</b> +32 475 23 05 12<br><b>Email:</b> <a href="mailto:paul@oacelectronics.be">paul@oacelectronics.be</a> |

## **Gegevens bachelorproef**

### **Titel bachelorproef**

Verkeersteller met snelheidsregistratie en type herkenning (b.v. fiets, personenwagen, lichte vracht, zware vracht...) van voertuigen.

### **Omschrijving van het probleem**

Ontwerp van een prototype verkeersteller voor federale/lokale politiediensten met volgende specificaties:

- telling van voertuigen in beide rijrichtingen
- registratie van snelheid
- herkenning type voertuig

### **Doelstellingen van de bachelorproef (kwantitatief en meetbaar)**

- Onderzoek combinatie 24Ghz I/Q radar
- Onderzoek combinatie DSP functies radar
- Onderzoek optimalisatie FPGA
- Ontwerp van PCB's voor de prototypes

### **Gewenst resultaat: wat moet er (minimaal) opgeleverd worden?**

Het aantonen van de meetprincipes en werking aan de hand van te ontwikkelen prototype PCB's.



# 1. Inleiding

Dit document omvat alle informatie rond het radar verkeersmonitor project. Eerst wordt er uitleg gegeven over de stageplek, de doelstelling van het project en hoe deze tot stand zijn gekomen. Gevolgd door een theoretische basis radar en signal processing om het project beter te begrijpen. Daarna wordt de gebruikte hardware tot in detail besproken met nadruk op enkele mechanische eigenschappen en een kijkje binnen het productieproces. Tot slot wordt de toegepaste software besproken.

## 2. OAC-Electronics

OAC-Electronics is al ruim 30 jaar gespecialiseerd in het design en ontwikkelen van op maat gemaakte elektronica.

Het bedrijf is gelegen in Herentals en heeft een eigen EMS-productielijn en pre-compliance testing mogelijkheden.



*Figuur 1 OAC Electronics te Herentals*

## 3. Opdracht

De opdracht omvat het monitoren van verkeer via radar technologie op basis van FPGA.

- Type voertuig herkennen
- Snelheid registreren
- Telling van voertuigen in beide rijrichtingen

### 3.1. Doelstelling

Ontwerp van een prototype verkeersteller voor federale/lokale politiediensten met volgende specificaties:

- Telling van voertuigen in beide rijrichting
- Registratie van snelheid
- Herkenning van het type voertuig

### 3.2. Onderzoek

- Combinatie 24Ghz I/Q radar
- Combinatie DPS functies radar
- Optimalisatie FPGA
- Ontwerp van PCB's voor de prototypes

### 3.3. Uitdagingen

- FPGA
- Digital Signal Processing
- FMCW-radar...

# 4. Radar: theorie

## 4.1. Introductie

Radar (**RA**dio **D**etection **A**nd **R**anging) is een manier om met behulp van elektromagnetische golven aanwezigheid, afstand, positie of snelheid van objecten te bepalen op het land, in de lucht of op het water.

De verkeersteller maakt gebruik van radar om voertuigen in beide rijrichtingen te gaan monitoren.

In deze introductie worden de fysieke details achter radar uitgelegd om de werking van het project beter te kunnen begrijpen.

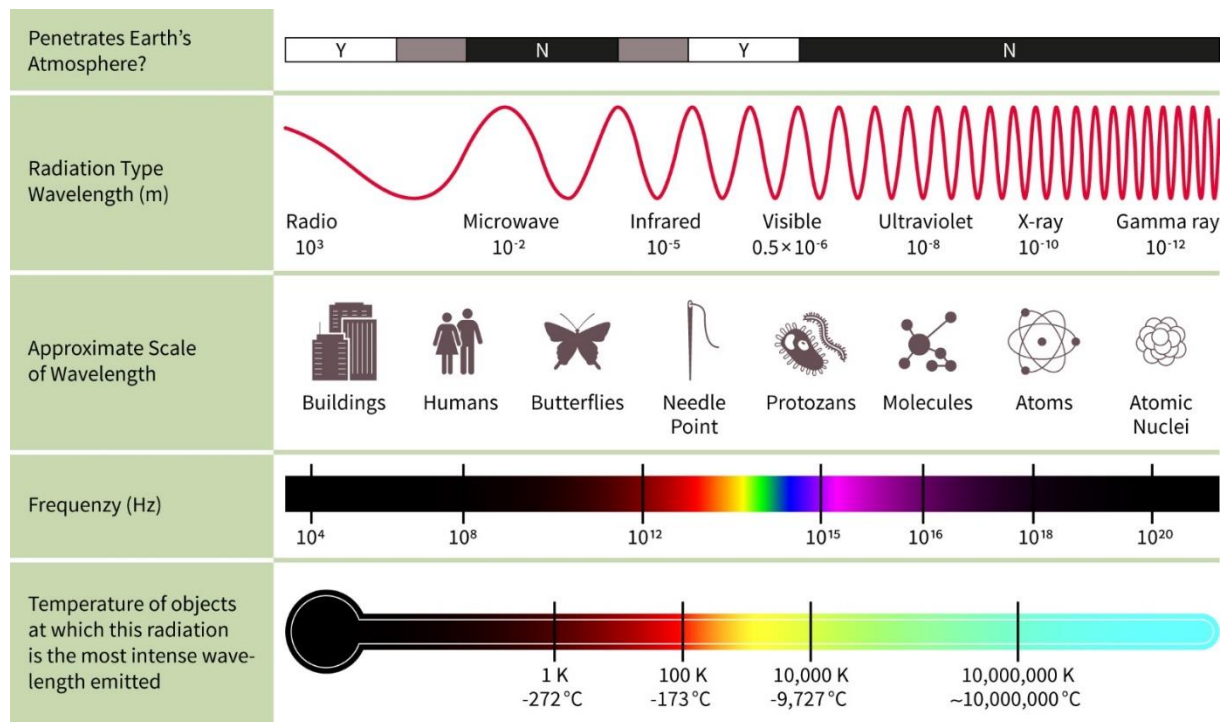
### 4.1.1. Elektromagnetische golven

Het elektromagnetische spectrum bestaat uit verschillende soorten elektromagnetische straling. Deze omvatten radiogolven, microgolven, infrarood, (zichtbaar) licht, ultraviolet, röntgenstraling en gammastraling, elk met zijn eigen toepassing.

Elektromagnetische golven verspreiden zich ongeveer met de snelheid van het licht, oftewel  $3 * 10^8$  (m/s).

De verkeersteller werkt op 24Ghz (K-Band), dit bevindt zich in het microgolf-spectrum. We kunnen de golflengte als volgt berekenen:

$$f = \frac{c}{\lambda} = \frac{\text{speed of light}}{\text{wavelength}} \quad \lambda = \frac{c}{f} = \frac{3 * 10^8 \text{ (m/s)}}{24\text{Ghz}} = 12.5\text{mm}$$



Figuur 2 Elektromagnetisch spectrum

Een andere eigenschap van elektromagnetische straling die wordt toegepast in dit project is het feit dat elektromagnetische straling niet alle materialen penetreert.

Hieronder een tabel van materialen en hun penetreerbaarheid.

| <b>Materiaal:</b>       | <b>Penetreerbaarheid:</b>  |
|-------------------------|--|
| <b>Metaal</b>           | Volledige reflectie, niet penetreerbaar                                  |
| <b>Water</b>            | Volledige absorbatie, niet penetreerbaar                                 |
| <b>Chemische dampen</b> | Goede penetratie, Kleine verliezen                                       |
| <b>Kleren</b>           | Droog – goede penetratie<br>Nat – verlies tot 20dB                       |
| <b>Regen</b>            | Voldoende penetratie, verlies tot 6dB                                    |
| <b>Plastic</b>          | Goede penetratie, 0.5dB tot 3dB verlies afhankelijk van dikte en afstand |
| <b>Personen</b>         | Slechte Penetratie, gedeeltelijke absorptie en reflectie                 |
| <b>Hout</b>             | Tot 10dB verlies   |

Hoe meer een materiaal reflecteert hoe beter detecteerbaar het is in een radar toepassing.

Uit bovenstaande materialen kan je bepalen dat metalen objecten en personen (koolstof) reflecties gaan veroorzaken en dus gedetecteerd kunnen worden door een radarsysteem.

Hieruit kan je ook beslissen uit welk materiaal je best een behuizing voorziet, de duidelijk winnaar is plastic, later hier meer over.

## 4.2. Radar

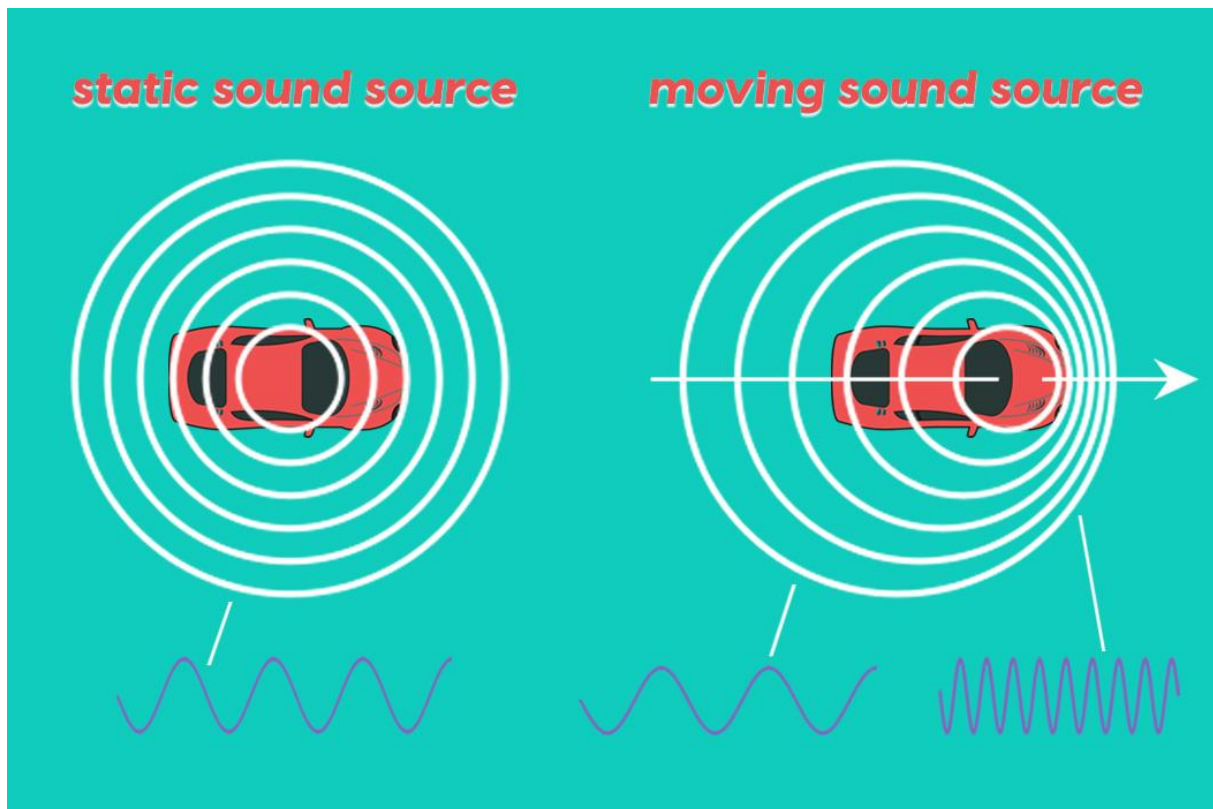
In een radarsysteem worden de elektromagnetische golven uitgezonden door de zendantenne (Tx) deze bewegen zich vervolgens voort in de ruimte (lucht) tot ze een elektrisch geleidend lichaam raken, dit kan bijvoorbeeld een persoon of wagen zijn. Daarna wordt de uitgezonden Elektromagnetische golf gereflecteerd en keert deze terug om ontvangen te worden door de ontvangstantenne (Rx).

Door de periode tussen zenden en ontvangen te meten kan afstand bepaald worden, wetende dat het signaal met de snelheid van het licht ( $3 * 10^8$  (m/s)) beweegt. Later wordt hier dieper op ingegaan.

Door het Doppler-effect zal het ontvangen signaal vervormd zijn tegenover het zendsignaal. Hierdoor kan de snelheid van een object bepaald worden.

### 4.2.1. Doppler (snelheid)

Normaal heb je op een of andere manier het Doppler-effect al ervaren, het meest voorkomende voorbeeld is een verandering in toon als een ambulance je passeert, de sirenes klinken sneller wanneer de ambulance nadert en trager wanneer hij wegrijdt. Dit wordt veroorzaakt door het Doppler-effect.



Figuur 3 Doppler-Effect

Het verschil tussen de zendfrequentie en ontvangen frequentie dat veroorzaakt wordt door het Dopplereffect kan gebruikt worden om snelheid van een object te bepalen, hoe hoger de snelheid hoe hoger de frequentie.

Dit verschil in frequentie noemen we de Doppler frequentie  $fd$ .

Deze kan als volgt bepaald worden:

- $v$  = snelheid (c)
- $fd$  = doppler frequentie
- $\lambda$  = golflengte

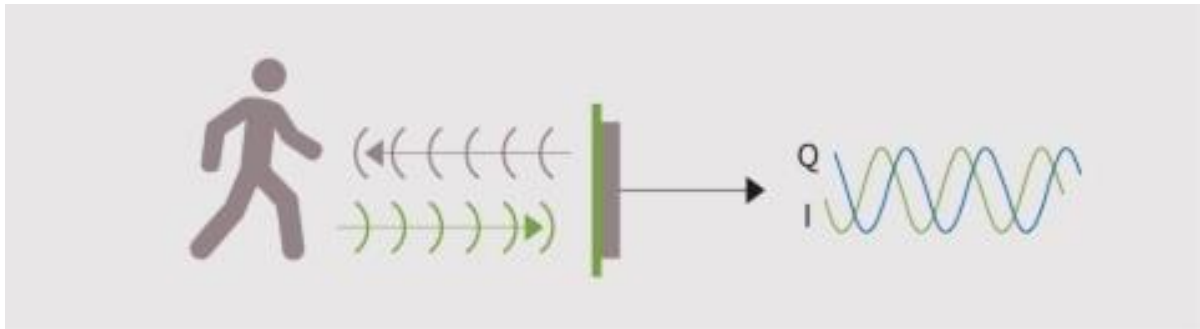
$$fd = \frac{(2*v)}{\lambda}$$

## 4.2.2. IQ-sigitaal (Richting)

Uit de gebruikte radar module komen 2 signalen, I en Q, beiden zijn exact hetzelfde enkel is het Q signaal  $90^\circ$  verschoven tegenover het I signaal. Deze fase verschuiving wordt bepaald door het gereflecteerde object, als deze nadert zal het Q signaal  $+90^\circ$  verschoven zijn tegenover het I signaal als het object weggrijdt zal het Q signaal  $-90^\circ$  verschoven zijn tegenover het I signaal. Hierdoor kan rijrichting bepaald worden.

De frequentie van dit IQ-sigitaal wordt bepaald door het Doppler-effect, hoe hoger de frequentie hoe sneller het gemeten object beweegt.

De amplitude wordt bepaald door het aantal ontvangen reflecties zo gaat de amplitude maximaal zijn als je bijvoorbeeld voor de radar module staat en minimaal als er geen objecten zijn. Op basis hiervan kan afstand gemeten worden maar dit is geen betrouwbare manier, later hier meer over.



Figuur 4 IQ-sigitaal



Figuur 5 Gemeten IQ-sigitaal in de praktijk

### 4.2.3. Modulatie (Afstand)

Afstand kan bepaald worden door verschillende modulatie soorten toe te passen.  
De voornaamste zijn:

- **CW**, Continuous **W**ave
- **Chirping**
- **FMCW**, Frequency **M**odulated Continuous **W**ave

Modulatie wil zeggen een verandering in zendsignaal. We kunnen dus verschillende zendtechnieken gebruiken om informatie te verzamelen, elk met hun voordelen/nadelen en complexiteit.

#### 4.2.3.1. Continuous Wave

Continuous wave radar (CW-radar) is de eenvoudigste vorm van modulatie, er wordt simpelweg constant verzonden en ontvangen.

Het nadeel hiervan is dat er geen tijdsreferentie aanwezig is bij het constant zenden (geen  $\Delta t$ ). Je gaat dus enkel afstand kunnen weergeven op basis van amplitude.

Dit heeft als nadeel bijvoorbeeld dat een vrachtwagen op 10m afstand evenveel golven kan reflecteren als een personenwagen op 2m afstand waardoor voertuig herkenning onmogelijk wordt.

Deze modulatie techniek wordt vooral toegepast in het detecteren van aanwezigheid en snelheid.

#### 4.2.3.2. Chirping

Bij chirping ga je een puls zenden om vervolgens de tijd te meten dat je signaal onderweg was tot je deze terug ontvangt, door de gemeten tijd kan je afstand gaan bepalen.

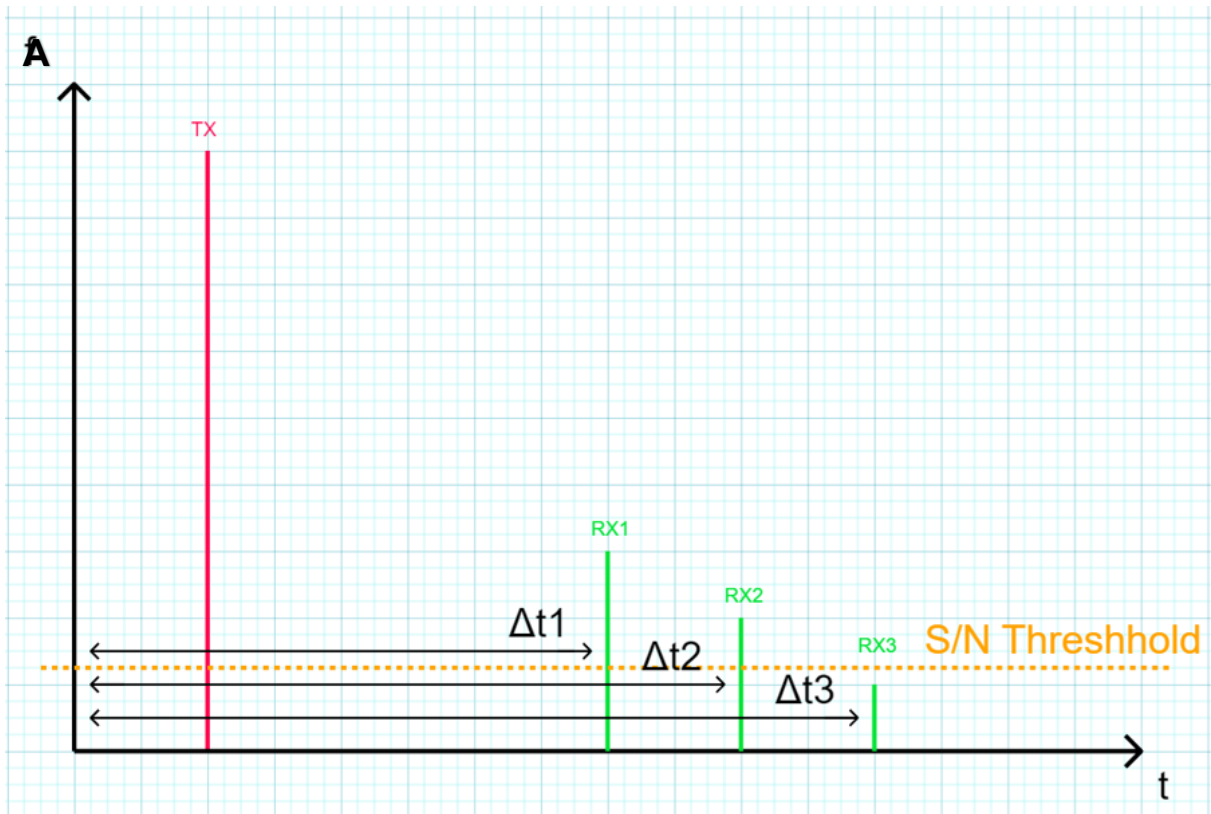
- $c$  = Snelheid van het licht ( $3 * 10^8$  (m/s)).
- $\Delta t$  = De tijd tussen zenden en ontvangen (s)

$$Afstand(m) = \frac{c * \Delta t}{2}$$

Je kan dus ook met 1 zendpuls meerdere reflecties gaan ontvangen waardoor meerdere objecten gemeten kunnen worden, bijvoorbeeld er rijdt 1 auto op 10m van de radar module een ander op 100m en de laatste op 1000m afstand. Als je lang genoeg meet ga je alle 3 reflecties ontvangen en kan je 3 afstanden gaan bepalen zolang de ontvangen reflecties onderscheiden kunnen worden van standaard aanwezige ruis of wel de signal to noise threshold (S/N threshold).

De signal to noise threshold(S/N) gaat bepalen wanneer een ontvangen signaal gezien kan worden als een informatie signaal of gewoon ruis. Alles boven de ingestelde S/N zal dus gezien worden als een informatie signaal, alles eronder zal gezien worden als ruis.





*Figuur 6 Voorbeeld Chirp met 3 reflecties op verschillende afstanden*

Snelheid kan je bepalen als je deze verzonden puls lang genoeg maakt zodat het Dopplereffect zich kan manifesteren op je gereflecteerd signaal. Dit noemt men Pulsed Doppler.

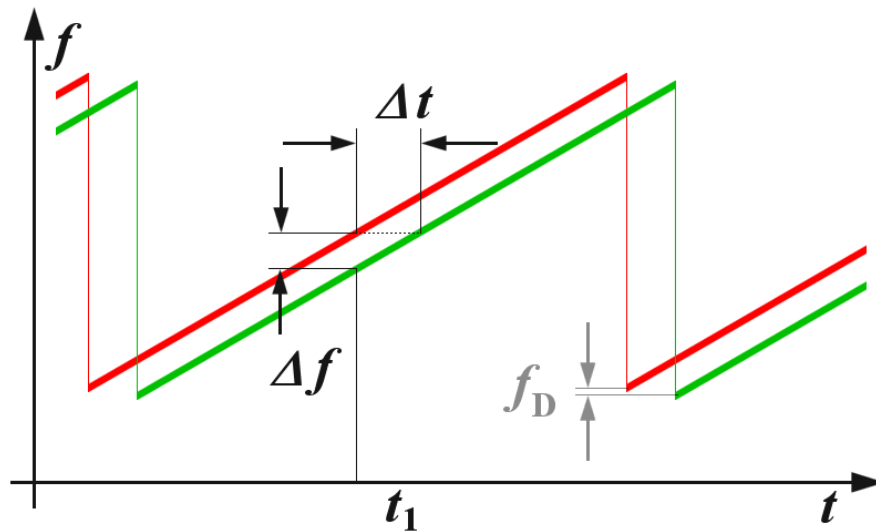
Dit is een goede en eenvoudige manier om voertuigen te gaan monitoren en wordt in de praktijk toegepast. Het enige nadeel hiervan is dat je korte periodes hebt waarop je niets meet, voor onze toepassing is dit geen groot probleem.

### 4.2.3.3. Frequency Modulated Continuous Wave

Frequency Modulated Continuous Wave radar (FMCW-radar) is een modulatie soort waarbij een continu zendsignaal uitgezonden wordt zoals bij een CW-radar.

In tegenstelling tot de CW-radar kan een FMCW-radar zijn zendfrequentie tijdens het zenden veranderen. Deze frequentieveranderingen maken extra meetmogelijkheden mogelijk.

Het is dus mogelijk om op deze manier een constante snelheid en afstand te meten, daarnaast is deze methode nauwkeuriger en minder gevoelig voor externe storing als zijn concurrenten.



Figuur 7 Afstands/snelheids bepaling zaagtand FMCW-radar

Afstand kan als volgt bepaald worden op 2 manieren, tijdsbasis en frequentiebasis:

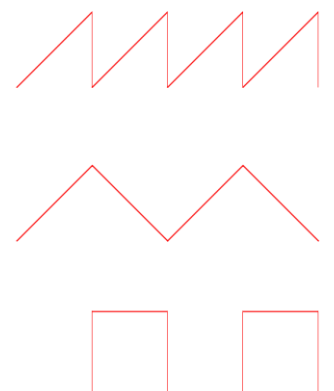
$\Delta f$  = gemeten frequentieverschil (Hz)

$B/Tc$  = frequentieafwijking per tijdseenheid

$$Afstand(m) = \frac{c * \Delta t}{2} = \frac{c * \Delta f}{2 * (\frac{B}{Tc})}$$

Binnen FMCW-radar zijn er ook verschillende modulatiepatronen met elk hun toepassing:

- **Zaagtand modulatie:**
  - Deze modulatie wordt gebruikt voor grote afstanden met verwaarloosbare invloed van het dopplereffect, bijvoorbeeld voor maritieme navigatieradar.
- **Driehoek modulatie:**
  - Deze modulatie maakt een eenvoudige scheiding mogelijk tussen  $\Delta f$  en de dopplerfrequentie, perfect voor de toepassing in dit project.
- **Blokgolf modulatie: (FSK)**
  - Deze modulatie wordt gebruikt voor zeer nauwkeurige afstandsmetingen van 1 object.



Figuur 8 Modulatie patronen voor een FMCW-radar

### Driehoeks frequentiemodulatie:

Met een driehoekige frequentieverandering kan een afstandsmeting worden verricht op zowel de opgaande als neergaande flank.

Het ontvangst signaal (groen) wordt naar rechts verschoven afhankelijk van de afstand van het te meten object, hierdoor wordt  $\Delta t$  bepaald.

Zonder dopplerfrequentie zal de grootte van het frequentieverschil  $\Delta f$  op de opgaande flank gelijk zijn aan de  $\Delta f$  op de neergaande flank.

$$(\Delta f_1 = \Delta f_2)$$

Met dopplerfrequentie verschuift het ontvangst signaal (groen) in hoogte.

De som van het frequentieverschil  $\Delta f$  en de dopplerfrequentie  $f_d$  verschijnt op de opgaande flank. ( $\Delta f_{op} + f_d = \Delta f_1$ )

Het verschil van beide frequenties verschijnt op de neergaande flank. ( $\Delta f_{neer} - f_d = \Delta f_2$ )

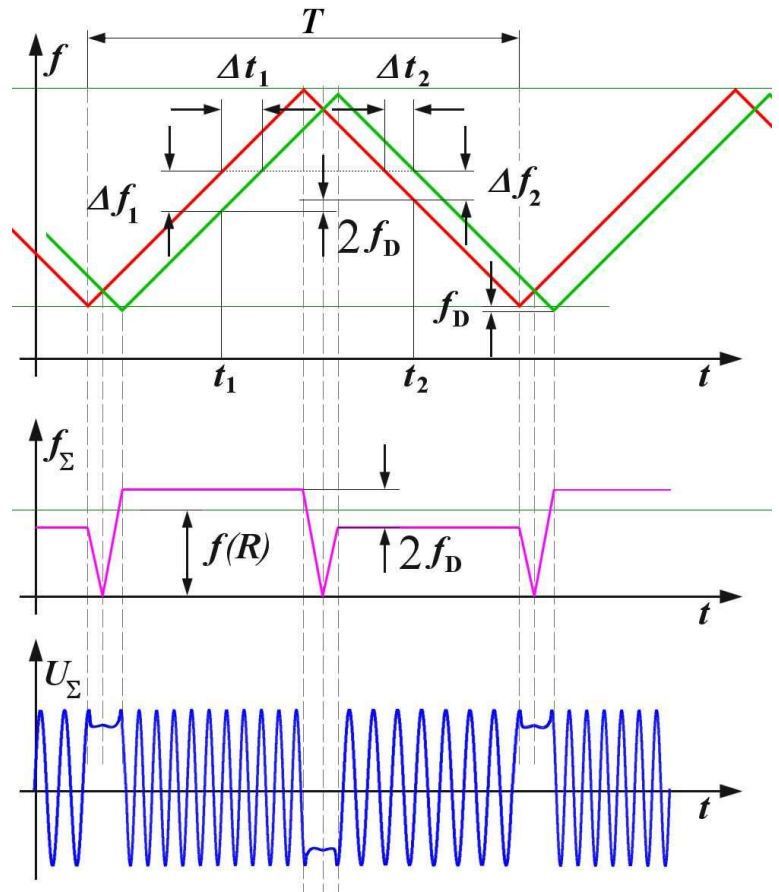
Dit geeft de mogelijkheid om een exacte afstandsbepaling te maken desondanks de frequentieverandering die veroorzaakt is door het dopplereffect.

$f(R)$  = Frequentie afstandsmeting

$f(D)$  = Doppler-frequentie snelheidsmeting

$\Delta f_1$  = Frequentie verschil opgaande flank

$\Delta f_2$  = Frequentie verschil neergaande flank



Figuur 9 Verband afstand & snelheid binnen driehoeksmodulatie

$$f(R) = \frac{\Delta f_1 + \Delta f_2}{2}$$

$$f(D) = \frac{|\Delta f_1 - \Delta f_2|}{2}$$

FMCW-radar is een heel mooie manier om snelheid en afstand te gaan bepalen, het nadeel hiervan is zijn complexiteit daarom wordt deze modulatie techniek gezien als het absolute einddoel binnen dit project.

## 5. Processing: theorie

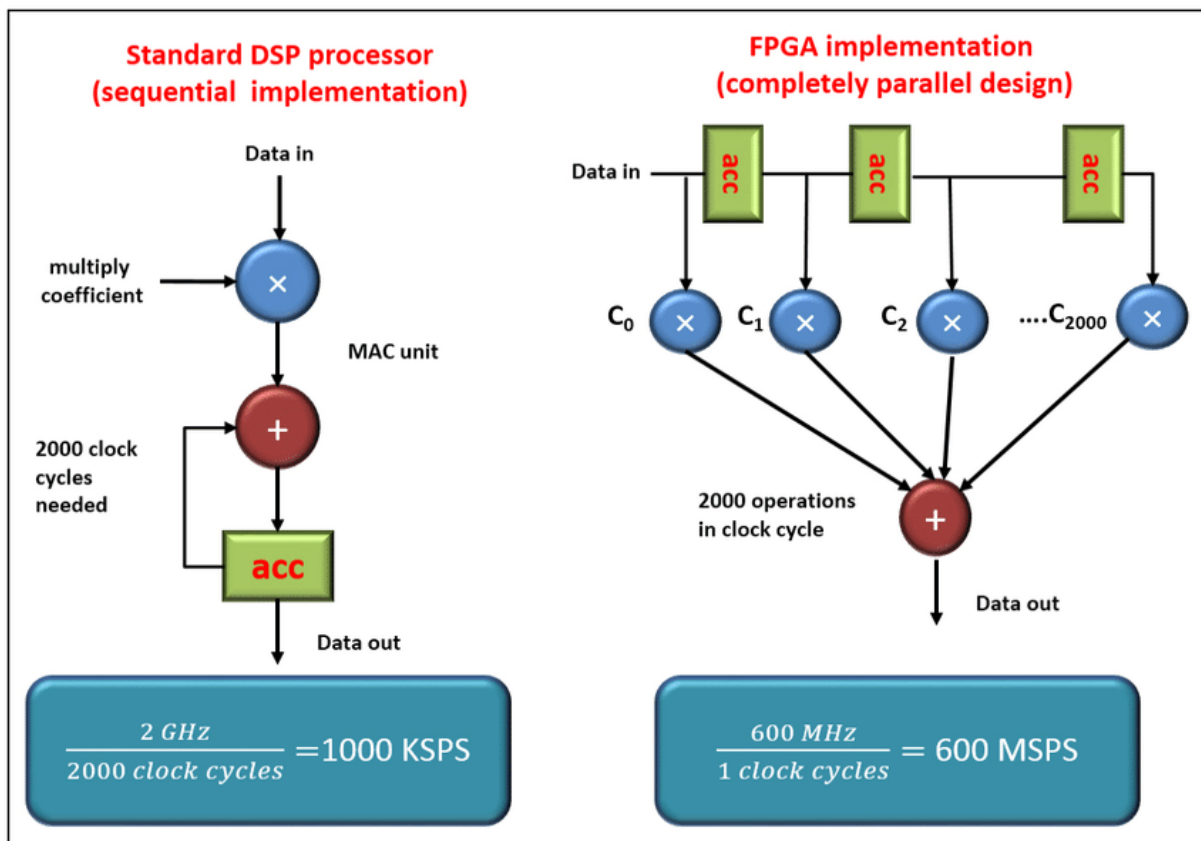
De processing theorie omvat het theoretische aspect achter de software. Om de werking van de software te kunnen begrijpen is het belangrijk om eerste deze theoretische basis aan te kaarten.

### 5.1. Digital Signal Processing

In tegenstelling tot een traditionele microcontroller heeft een FPGA de mogelijkheid om parallel processing te benutten. Dit is handig als je grote delen data snel en tegelijkertijd moet verwerken.

In dit project wordt het beste van 2 werelden benut zo is er gekozen om op de FPGA een softcore te implementeren. Deze biedt de traditionele werking en programmatie van een microcontroller met de mogelijkheid om bepaalde functies en algoritmes op FPGA-niveau oftewel parallel basis uit te voeren.

Een eigenschap van dit project is dus ook om te kijken of FPGA's voordeliger en of beter zijn voor radar toepassingen.



Figuur 10 Microcontroller VS FPGA DSP

## 5.2. Sampling ADC

Het omzetten van een analogoog signaal naar een digitaal signaal kunnen we gaan doen doormiddel van een Analogoog Digitaal Converter (ADC) te gebruiken. Deze heeft 2 voornaamste parameters waar rekening moet mee gehouden worden bij het kiezen van een ADC, Sampling Rate & Resolutie.

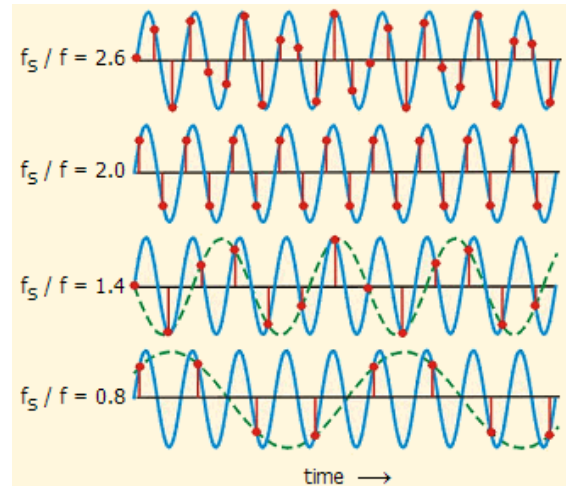
### Sampling Rate:

De sampling rate, ook wel sampling frequentie genoemd, bepaald hoeveel samples er per seconden worden gemeten.

Een sample zijnde een meting van een analogoog signaal op een tijdstip.

Het is belangrijk om rekening te houden met het "Nyquist-Shannon sampling theorem", die zegt dat de sampling frequentie op zijn minst het dubbele moet zijn van de te meten frequentie om aliasing te voorkomen.

Aliasing is wanneer de sampling frequentie te laag ligt en zo een incorrecte meting gaat weergeven zoals hiernaast afgebeeld. ( $\frac{f_s}{f} > 2$ )



Figuur 11 Sampling voorbeeld Shannon-Nyquist

### Resolutie:

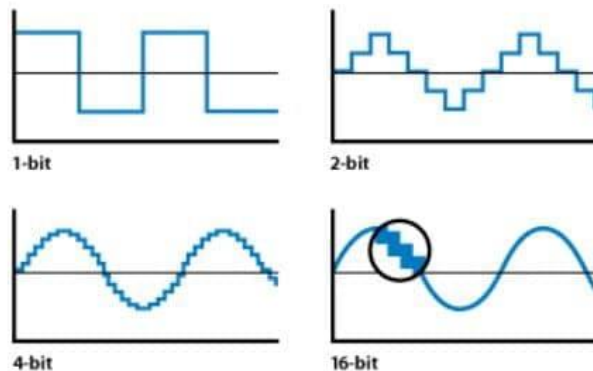
De resolutie gaat de precisie van het gemeten analoge signaal bepalen, ook dus hoeveel een meting kan verschillen van de effectieve analoge waarde.

Bijvoorbeeld, er is gekozen voor een resolutie van 12bit, het te meten signaal bevindt zich tussen 0V en +5V (5Vpp). De nauwkeurigheid kan als volgt bepaald worden:

$$\frac{V_{inmax} - V_{inmin}}{Resolution} = \frac{5V - 0V}{12bit(= 4096)} = 1.22mV$$

Er kunnen dus verandering van minimaal 1.22mV digitaal gezien worden.

Figuur 12 Resolutie Analogoog signaal voorbeeld



De uiteindelijke digitale uitgang zal een binaire waarde zijn die bepaald wordt door de resolutie, uitgaande van bovenstaand voorbeeld geldt het volgende:

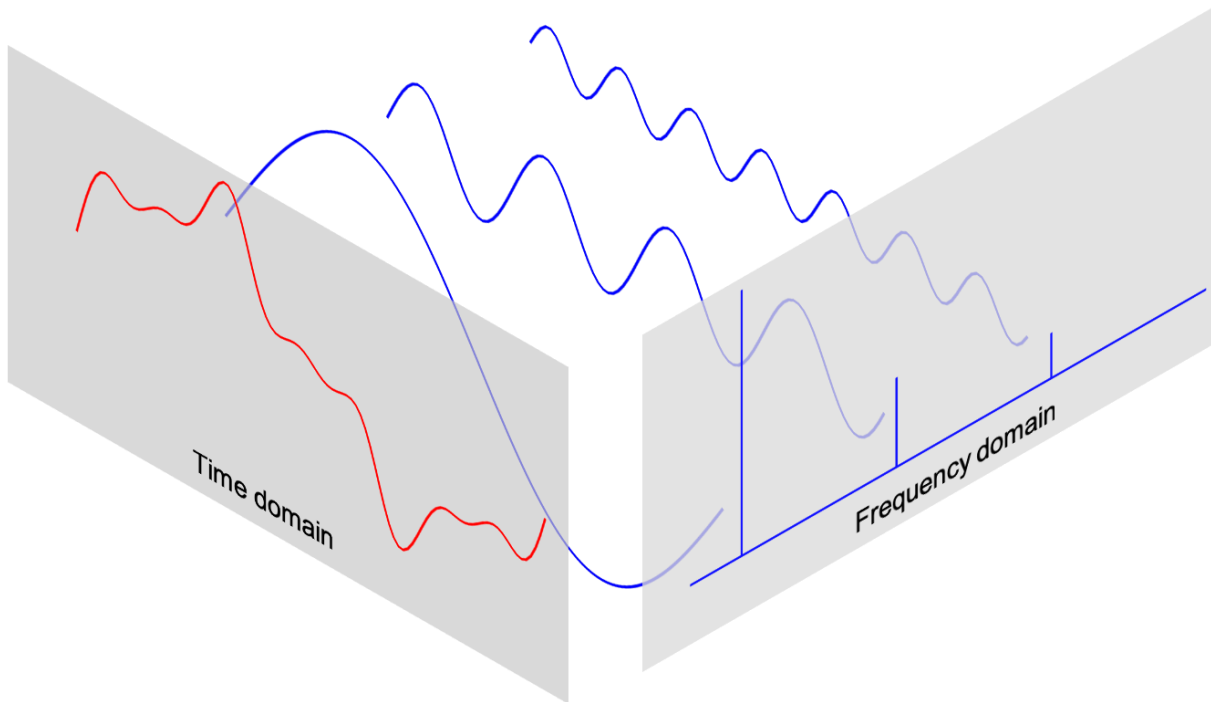
- 000000000001 = 1.22mV
- 000000000010 = 2.44mV
- 000000000011 = 3.66mV
- ...
- 111111111110 = 4998.78mV
- 111111111111 = 5V

In dit project is een FPGA gekozen met interne ADC, deze beschikt over een resolutie van 12bit en een instelbare sampling rate tot 1Msample/sec. Meer als voldoende voor deze toepassing. Later hier meer over.

## 5.3. Fourier transformation

### 5.3.1. Discrete Fourier Transform (DFT)

De DFT is een wiskundig algoritme dat gebruikt wordt om een signaal in het tijdsdomein om te zetten naar het frequentiedomein. Dit heeft als voordeel dat frequentie softwarematig kan bepaald worden.



Figuur 13 Discrete Fourier Transform Time domain to frequency domain

De volgende formule geldt:

$X(k)$  = De output

$X(n)$  = De input (sample n)

$N$  = Het aantal samples

$e^{jx}$  = Getal van Euler,  $\cos(x) - j * \sin(x)$

$k$  = Frequentie

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-jnk2\pi/N} \quad k = 0, \dots, N - 1$$

Dit legt de basis van de Fast Fourier Transform (FFT).

### 5.3.2. Fast Fourier Transform (FFT)

De FFT is een geoptimaliseerd versie van de DFT voor signal processing toepassingen.

Eerst en vooral is het belangrijk om volgende parameters te herkennen:

- Sampling rate of sampling frequency, dit is het gemiddelde aantal samples gemeten per seconden. ( $f_s$ )
- Input Block Length, de input van de FFT meestal bestaande uit een complex signaal. Dit is altijd een macht van 2 ( $BL$ ).

Met deze 2 parameters kunnen we verdere parameters gaan bepalen.

**Bandbreedte  $f_n$**  (Nyquist frequentie), dit bepaalt de theoretische maximum frequentie die gemeten kan worden door de FFT.

$$f_n(\text{Hz}) = \frac{f_s}{2}$$

Bijvoorbeeld, bij een sampling rate van 100kHz kan er een maximum frequentie correct gemeten worden van 50kHz.

**Frequentie resolutie  $df$**  (frequentie bins), dit geeft de nauwkeurigheid aan per frequentie stap (bin).

$$df = \frac{f_s}{BL}$$

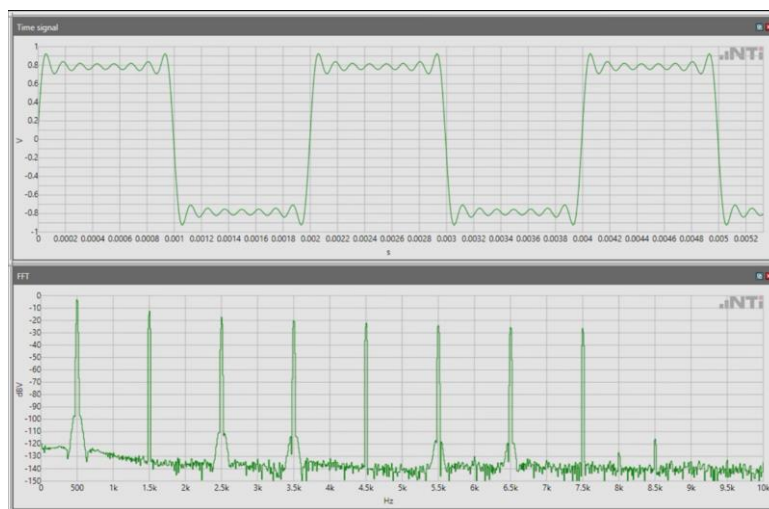
Bijvoorbeeld, bij een sampling rate van 100kHz en block length van 2048 is 1 frequentie bin gelijk aan 48.8Hz. Zo vertegenwoordigt frequentie bin1=48.8Hz, bin2=97.6Hz ... bin2048=100kHz.

Aangezien de sampling rate in meeste toepassingen niet regelbaar is wordt er meestal gekozen om de FFT te regelen via zijn block length.

Zo resulteert een kleine block length in een snelle minder precieze meting die vaak ook minder processing power vereist, en een grote block length in een tragere precieze meting die wat meer processing power vereist.

Als input signaal van de FFT wordt het IQ-signaal gebruikt met volgende complexe structuur:  $I + iQ$ .

Waarbij  $I$  de reële waarde voorstelt en  $Q$  de imaginaire waarde.



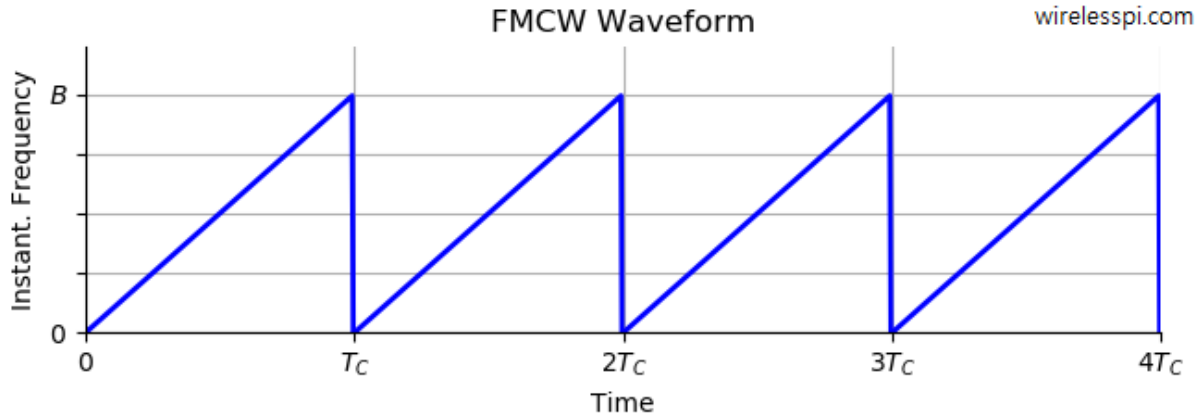
Figuur 14 [Voorbeeld FFT met Fourier Blokgolf](#)

## 5.4. FMCW-RADAR Digital Signal Processing

Dit hoofdstuk probeert een beter beeld te scheppen van de digital signal processing achter een FMCW-radar systeem.

Bij een Zaagtand modulatie kan de chirp rate (frequentie afwijking per tijdseenheid) als volgt bepaald worden.

$$\text{Chirp rate} = \mu = \frac{B}{T_c} \text{ (Hz/s of MHz}/\mu\text{s)}$$



Figuur 15 Chirp rate example zaagtand

Om afstand te bepalen is het mogelijk om de delay ( $\tau_0$ ) tussen zenden en ontvangen te meten, echter is dit geen goede manier aangezien het meten van korte delays vaak ingewikkeld en inaccuraat is.

Een betere manier is om de beat frequentie  $fb$  te meten, deze frequentie wordt geïntroduceerd bij het ontvangen van het gereflecteerd signaal.

De beat frequentie kan als volgt berekend wetende dat de chirp rate parameters door de gebruiker zijn ingesteld:

$$fb = \mu * \tau_0$$

$\tau_0$  Kan herleid worden naar 2 keer de afstand (heen en terug) gedeeld door de snelheid van het licht:

$$\tau_0 = \frac{2 * R}{c}$$

$\mu$  Kan zoals hierboven vermeld herleid worden naar:

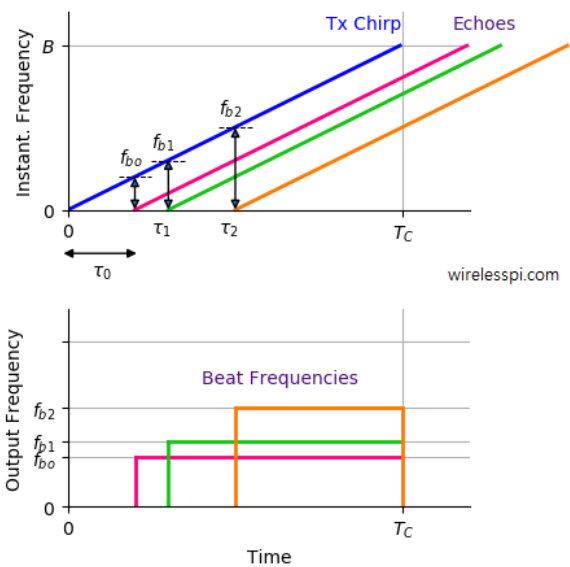
$$\mu = \frac{B}{T_c}$$

$fb = \mu * \tau_0$  Kan nu herleid worden naar:

$$fb = \frac{B}{T_c} * \frac{2 * R}{c}$$

Door deze formule wordt het mogelijk om de afstand  $R$  te bepalen aan de hand van de gemeten beat frequentie  $fb$ :

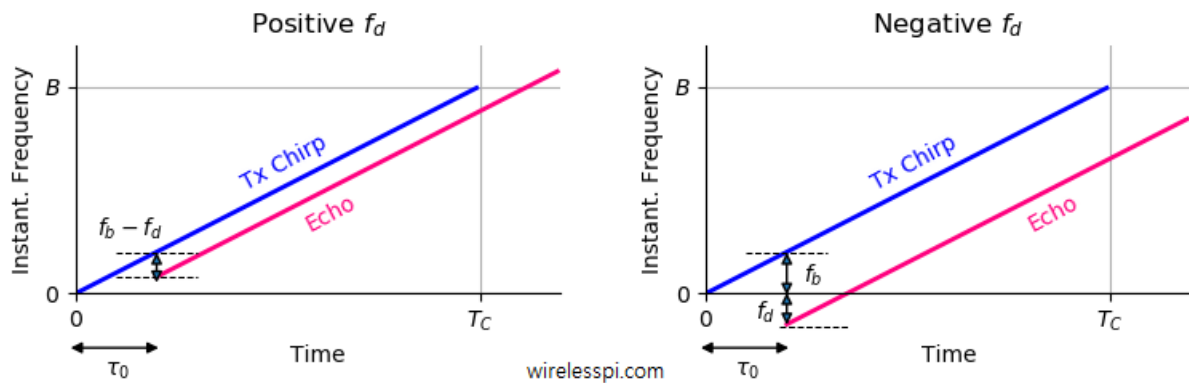
$$R = \frac{c * T_c * fb}{2B}$$



Figuur 16 Beat Frequencies Example



Door het Dopplereffect zal de frequentie van het gereflecteerd signaal hoger of lager liggen afhankelijk van de rijrichting. Dit biedt de mogelijkheid om snelheid te bepalen.

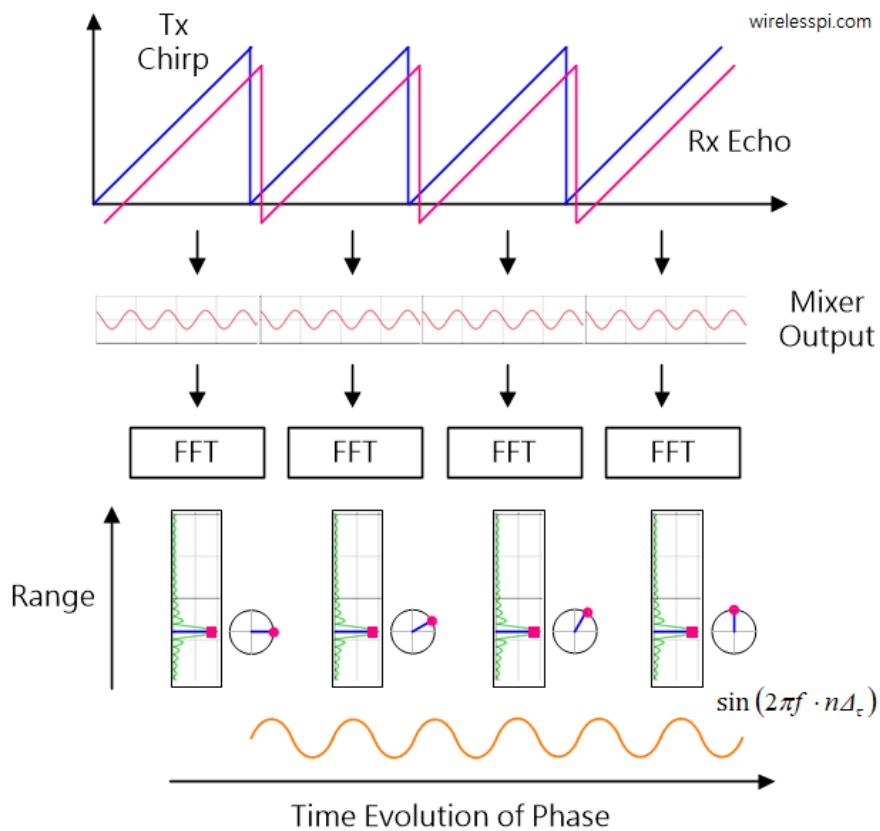


Figuur 17 FMCW-RADAR doppler meting

Om snelheid te bepalen wordt er van elke chirp een range FFT bepaald (de beat frequentie  $f_b$ ) voor N aantal keren. Door de korte periode van deze metingen verandert de afstand bijna niet, wat wel verandert is de fase van elke FFT piek (beat frequentie  $f_b$ ) die de afstand representeert.

Deze fase verschuiving wordt veroorzaakt door het Dopplereffect. Als er dan een doppler FFT wordt toegepast op de N aantal pieken van de range FFT's kan de doppler frequentie bepaald worden. Hierdoor wordt het mogelijk om snelheid te bepalen op basis van fase verschuiving.

Volgende afbeelding toont een voorbeeld van 1 object met een bepaalde snelheid dat gemeten wordt. Merk op hoe de fase van elke piek verandert.



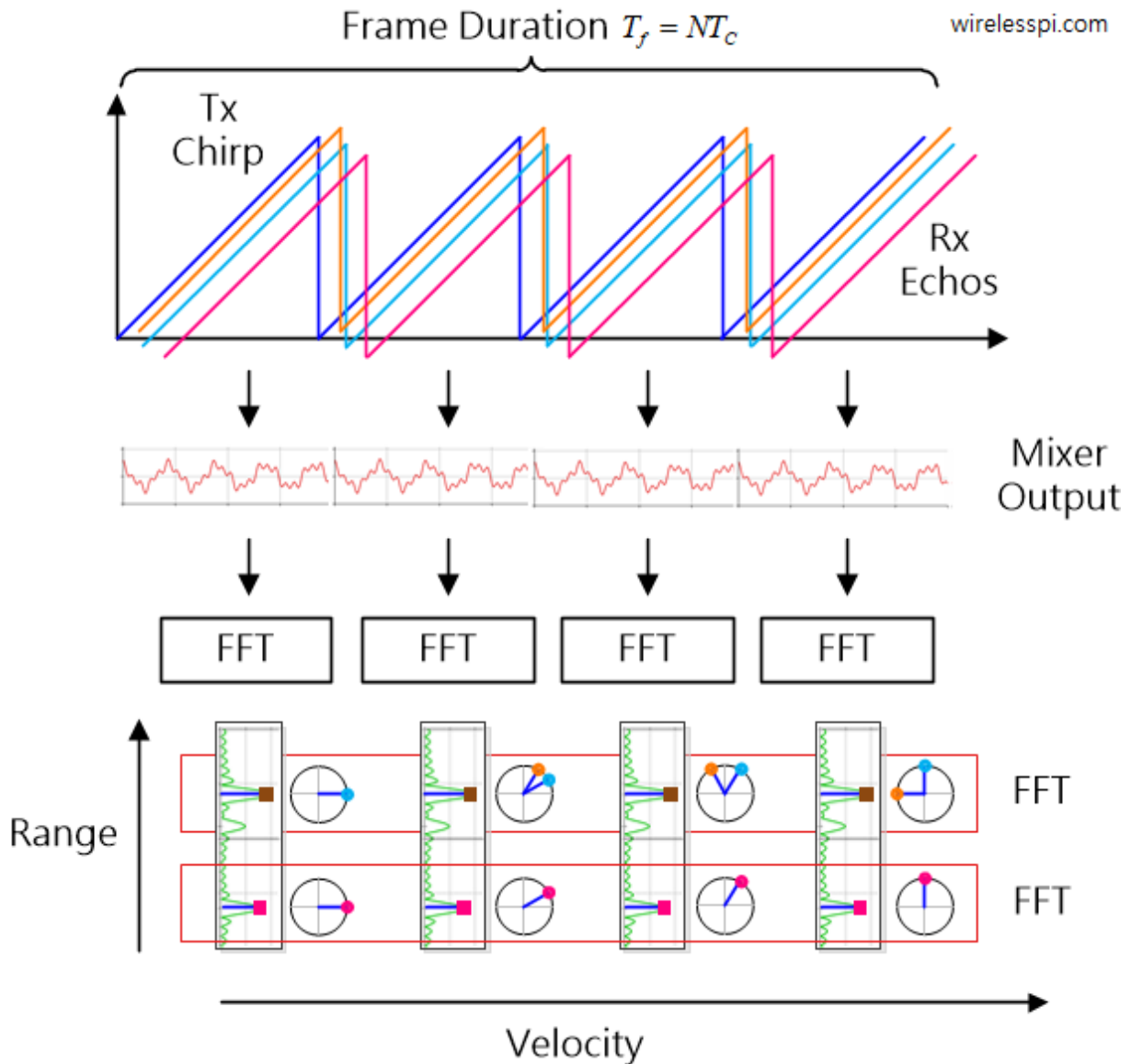
Figuur 18 FMCW-RADAR doppler bepalen a.d.h.v. fase verschuiving

Snelheid kan met de volgende formule bepaald worden op basis van de gemeten doppler frequentie:

$$v = \frac{\lambda * fd}{2}$$

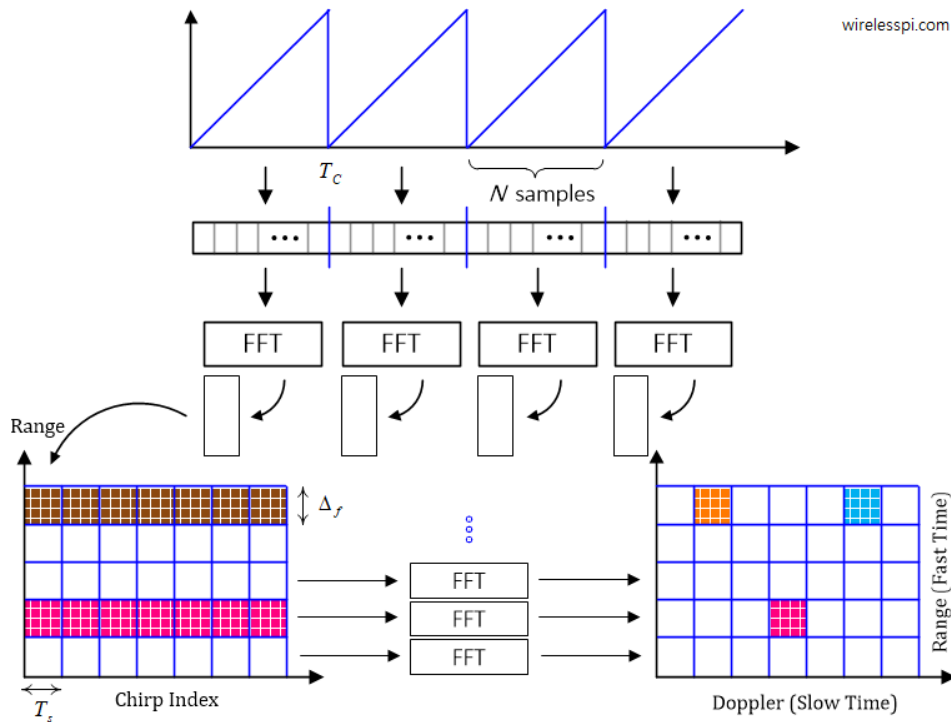
Door deze techniek wordt het dus ook mogelijk om verschillende objecten op dezelfde afstand met een verschillende snelheid te gaan detecteren.

Onderstaand voorbeeld toont 3 targets waarvan 2 targets op dezelfde afstand met een verschillende snelheid en rijrichting.



Figuur 19 FMCW-RADAR 3 targets, 2 afstanden, 3 snelheden

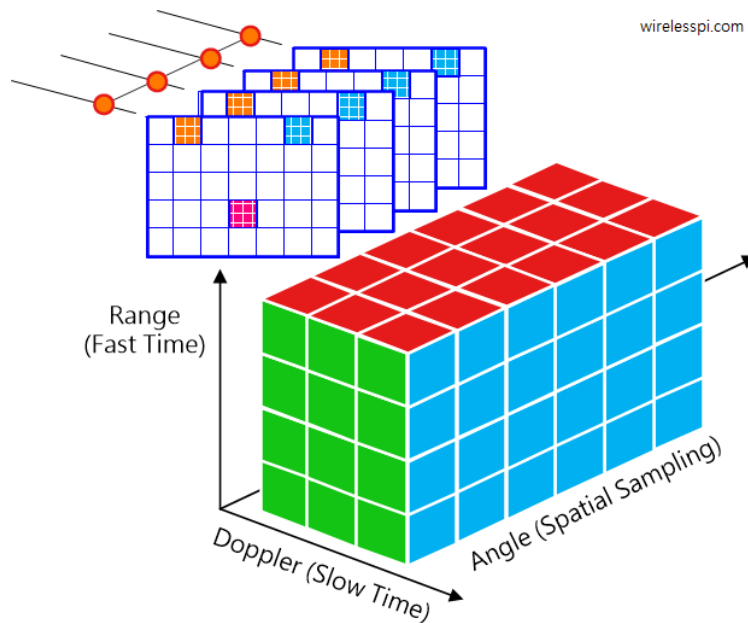
Volgend blokschema omvat het bovenstaande scenario in een ander perspectief waarbij de data in een 2D grafiek wordt weergegeven.



Figuur 20 FMCW-RADAR 2D-DATA

Vaak hebben FMCW-radars meerdere ontvangstantennes, hierdoor wordt het mogelijk om de invalshoek van een signaal te bepalen.

Voor de invalshoek te bepalen wordt bovenstaand proces doorlopen voor elke ontvangstantenne, hierdoor kan een derde dimensie toegevoegd worden aan de 2D data door deze in een Radar Data Cube te representeren zoals onderaan weergegeven.



Figuur 21 FMCW-RADAR 3D RADAR DATA CUBE

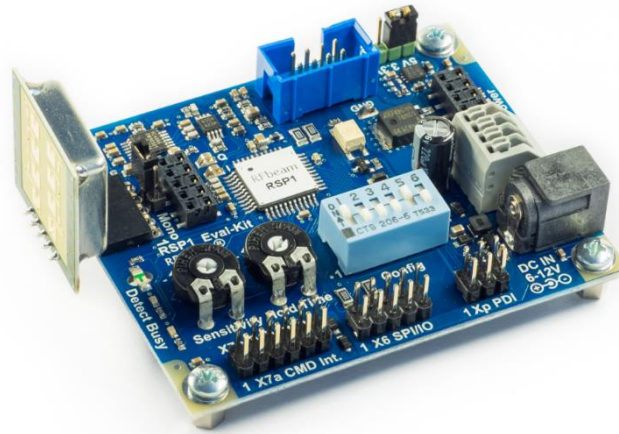
## 6. Radar: praktijk

### 6.1. RSP1-evaluatie bordje

De eerste kennismaking met Radar is gemaakt met het RF-beam RSP1-evaluatie bordje, hierbij was een development bordje aanwezig met enkele verschillende radarmodules. In de documentatie was heel veel informatie terug te vinden over DSP en parameters van de radar module.

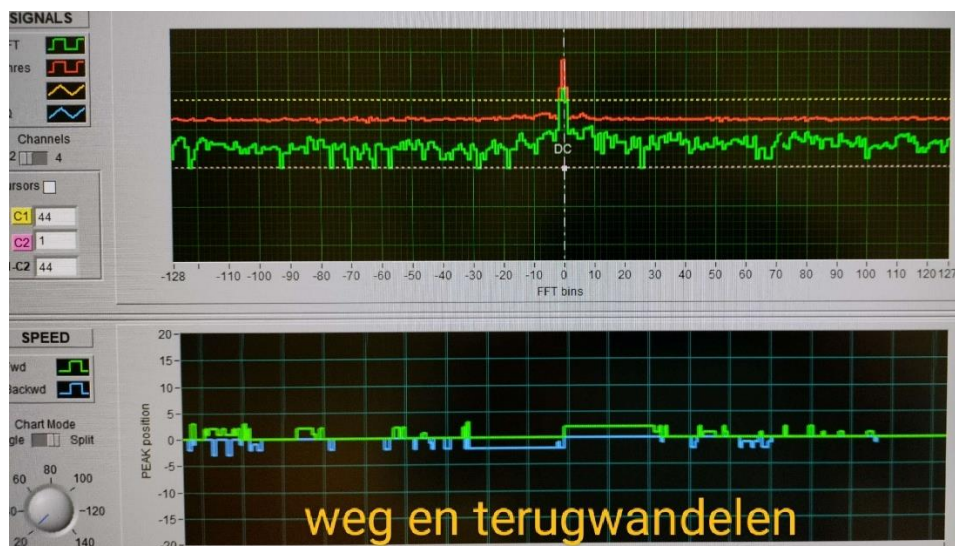
Op basis van die documentatie zijn veel parameters bepaald voor dit project.

Deze radarmodules waren vooral low profile, bedoeld voor eenvoudige applicaties. Hierdoor had ik al snel door dat voor onze toepassing een uitgebreidere radarmodule nodig was.



Figuur 22 RSP1 evaluation board

Plug & Play software was ook voorzien, hierdoor kon ik op een spelende wijze bijleren en had ik al een eerste kennismaking met Doppler en FFT.



Figuur 23 RSP1 Test software

## 6.2. Custom amplifier

Het custom amplifier bordje is gemaakt met als nut de uitgangsspanning, van de bijgeleverde radar modules van het RSP1-evaluatie bordje, om te zetten naar een spanning tussen 0V en 1V (1Vpp) om testen te kunnen uitvoeren met het basys3 FPGA development board.

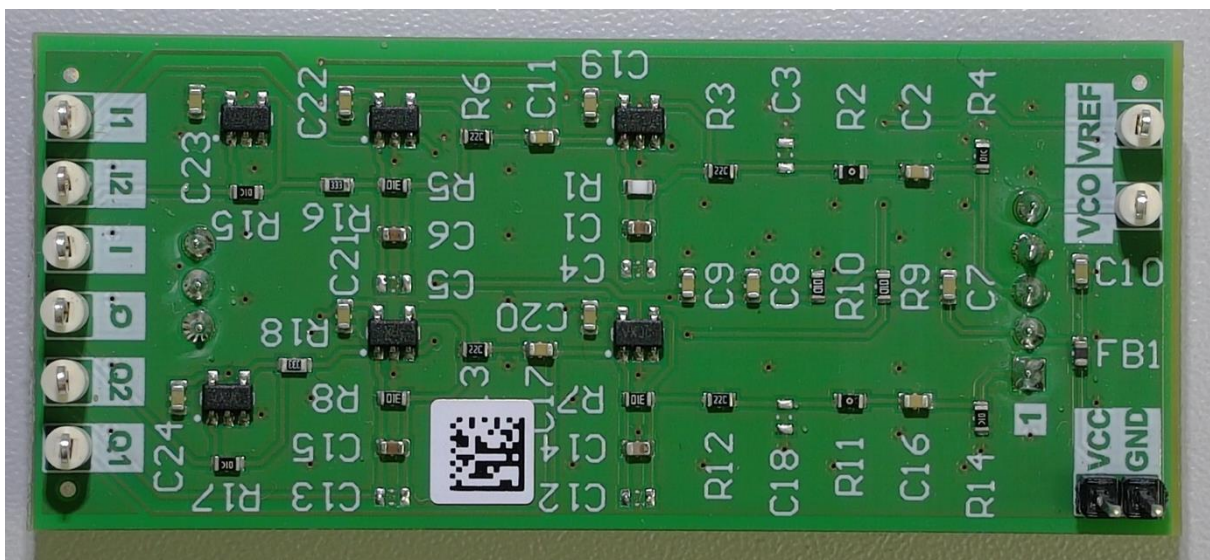
Dit was een goede manier om de altium skills nog eens op te warmen en een eerste keer kennis te maken met het OAC productie proces.

De eerste 2 versterkingstrappen werken zoals verwacht, met een bandbreedte tussen de 50Hz en 30kHz, de laatste trap niet, die diende om de spanning omlaag te halen om het informatie signaal tussen 0V en 1V te krijgen.

Dit komt omdat er een inverterende attenuator is gebruikt die aan zijn ingang altijd een positieve spanning heeft, deze zal dus altijd een negatieve uitgangsspanning hebben die niet gemaakt kan worden aangezien de attenuator niet voorzien is van een negatieve voeding (minimaal 0V).

Daarom is er gekozen om met een simpele spanningsdeler te werken na de eerste 2 versterkingstrappen, hierdoor is het bovenstaand probleem opgelost.

Alle informatie zoals schema's, PCB, 3D-model en BOM zijn terug te vinden onder 13.1 Bijlagen.



Figuur 24 Prototype Custom Amplifier PCB

### 6.3. InnoSent IVQ-3005

De gebruikte radar module in dit project is de InnoSent IVQ-3005.

Deze state of the art radar module is voorzien van 2 ontvangst antennes waardoor het mogelijk wordt om de invalshoek van een gedetecteerd object te gaan bepalen.

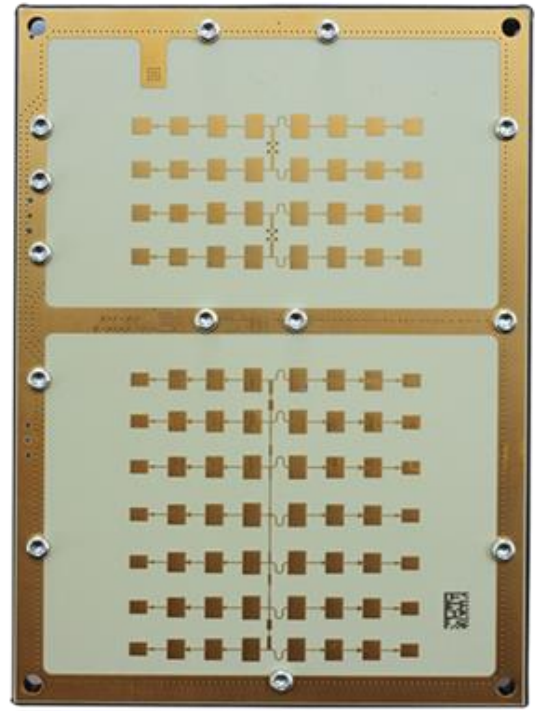
De interne Voltage Controlled Oscillator (VCO) ondersteund modulatiesoorten zoals **CW**, **FSK** en **FMCW**.

In FMCW-mode kan deze afstand, snelheid, richting en hoek bepalen van een bewegend of stilstaand object binnen de 300m.

De modulatie soort kan ingesteld worden via de onboard ADF4158, dit is een Direct modulation/waveform generating frequency synthesizer. Communicatie hiermee gebeurt via de SPI, TX\_Data en MUX\_DATA pinnen.

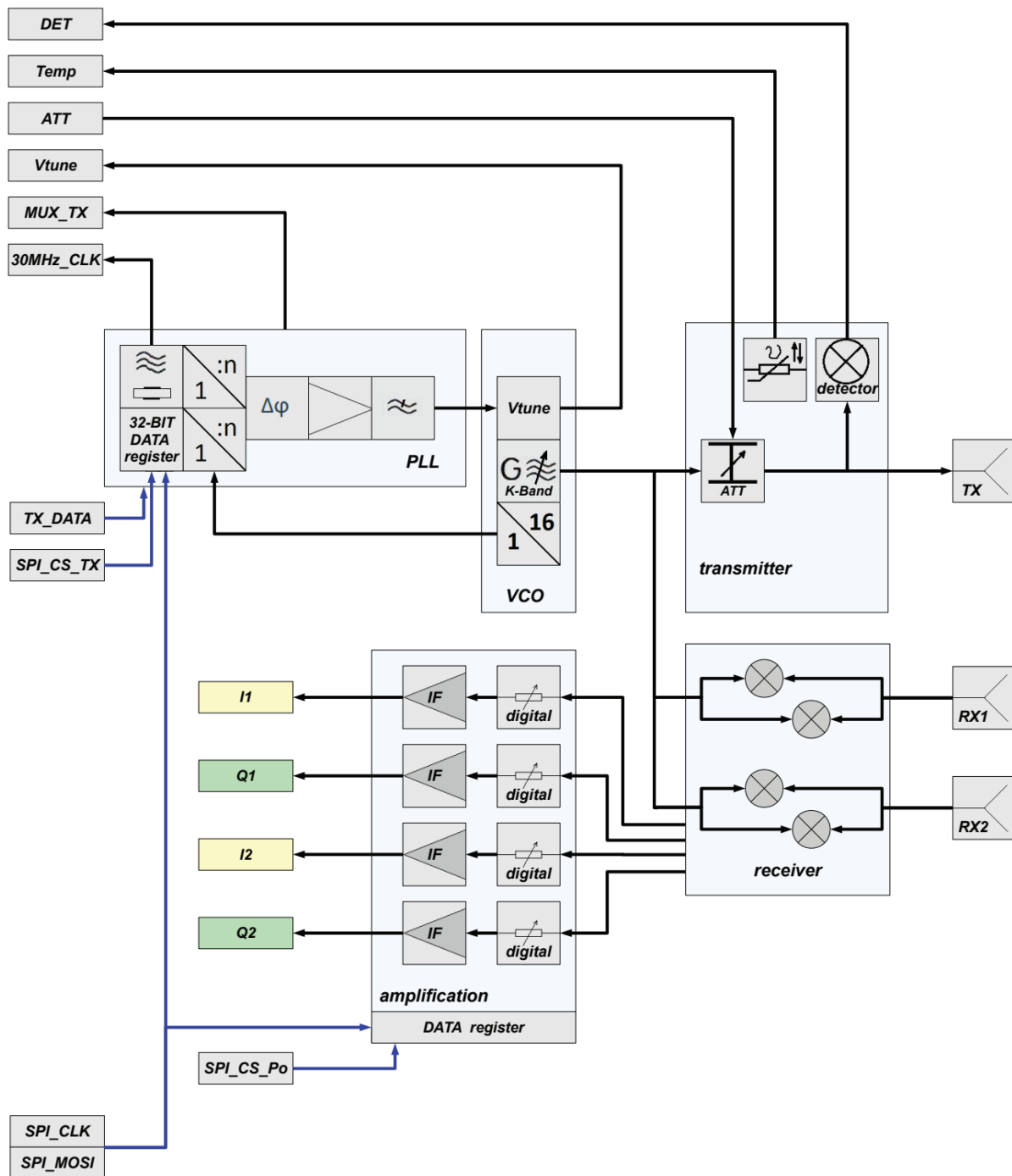
De radar module heeft ook nog enkele andere coole features zoals:

- **Detection pin**
  - Geeft het momentele zendvermogen weer in de vorm van een analogoog signaal tussen de 0 en 3V3.
- **Temperatuur regeling**
  - Temperatuur kan invloed hebben op de werking van de radar module, met deze pin kan temperatuur gemeten worden (LM20CIM).
- **Attenuator**
  - Het maximaal zendvermogen is 30dB dit is hoger als het maximaal toegelaten zendvermogen (20dB), met deze attenuator kan het zendvermogen geregeld worden door een signaal tussen 0 en 3.3V aan te leggen.
- **SPI-communicatie**
  - De amplifier en PLL zijn volledig instelbaar via SPI, later hier meer over.
- **On board regelbare amplifier**
  - Deze onboard amplifier geeft een I/Q signaal dat tussen de 0V en 3V ligt met een DC-offset van 1.5V.



*Figuur 25 InnoSent IVQ-3005 Radar-Module*

In het blokschema wordt de interne opbouw mooi weergegeven.



Figuur 26 Blokschema IVQ-3005

Deze radar module heeft 1 groot nadeel en dat is zijn Chinese connector, de IRISO IMSA-9855B-22Y914 online is hier niks over terug te vinden, na enkele weken mailen met zowel IRISO als InnoSent zijn er 4 samples bemachtigd kunnen worden met datasheet en 3D-model.

De InnoSent IVQ-3005 is een beetje overkill voor een toepassing zoals deze, echter perfect om alles mee uit te werken in een prototype. Later bij een volgende iteratie van dit project kan een goedkopere radar module gebruikt worden om kostprijs omlaag te halen.

## 7. Hardware

De hardware omvat alle informatie die nodig is om het ontwikkelde prototype PCB te kunnen begrijpen. Alle informatie zoals schema's, PCB, 3D-model en BOM zijn terug te vinden onder 13.2 Bijlagen.

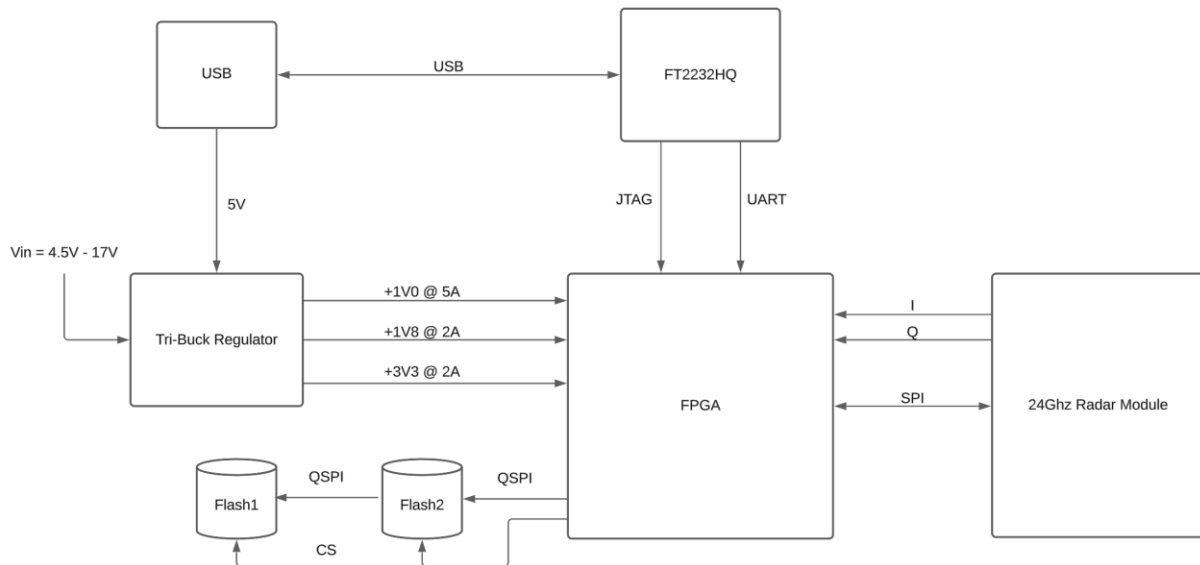
### 7.1. Electrical data

| Parameter             | Conditions | Symbol           | Min  | Typ. | Max   | Units |
|-----------------------|------------|------------------|------|------|-------|-------|
| <b>Power Supply:</b>  |            |                  |      |      |       |       |
| Supply Voltage        |            | V <sub>CC</sub>  | 4.8  | 5    | 18    | V     |
| Supply Current        |            | I <sub>CC</sub>  | 120  | 650  | 750   | mA    |
| <b>I/O voltage:</b>   |            |                  |      |      |       |       |
| Input high            | LVC MOS33  | V <sub>iH</sub>  | 2    | 3.3  | 3.45  | V     |
| Input low             | LVC MOS33  | V <sub>iL</sub>  | -0.3 | 0    | 0.8   | V     |
| Output high           | LVC MOS33  | V <sub>oH</sub>  | 2.9  | 3.3  |       | V     |
| Output low            | LVC MOS33  | V <sub>oL</sub>  |      | 0    | 0.4   | V     |
| <b>Transmitter:</b>   |            |                  |      |      |       |       |
| Transmit frequency    |            | f                | 24   |      | 24.25 | GHz   |
| Output power (EIRP)   | adjustable | P <sub>adj</sub> | 8    |      | 30    | dBm   |
| <b>Omgeving:</b>      |            |                  |      |      |       |       |
| Operating temperature |            | T <sub>dp</sub>  | -25  |      | +60   | °C    |



## 7.2. Blokschema

Dit blokschema omvat de algemene werking en verbinding tussen componenten op de custom PCB. Alle componenten worden in dit hoofdstuk tot in detail besproken.



Figuur 27 Simpel blokschema radar

## 7.3. Schema's

Alle schema's zijn te raadplegen onder "13.2 Bijlagen" achteraan in dit document.

De schema's zijn zo leesbaar mogelijk getekend, met zo veel mogelijk referenties naar informatie. Dit heeft als voordeel dat externen heel snel de schema's kunnen begrijpen.

### 7.3.1. Blokschema

Dit blokschema (zie 13.2 Bijlagen) omvat de algemene werking en layout van het project. Hierin is een mooi overzicht te zien van wat er zich allemaal in de FPGA bevindt en wat er zich rond de FPGA bevindt.

### 7.3.2. Power

Op het power schematic is de algemene voeding en voedingsingang te zien.

De ingangsvoeding bestaat uit een 5V DC USB C connector of een externe voeding die minimaal 5V tot maximaal 17V mag bedragen, bijvoorbeeld een 12V batterij.

Er zijn 2 polariteits beveiligingen (D101, D102) voorzien. Hiervoor is een Schottky diode gekozen om het spanningsverlies zo klein mogelijk te houden (0.5V).

De USB C connector is ook voorzien van een ESD protection device.

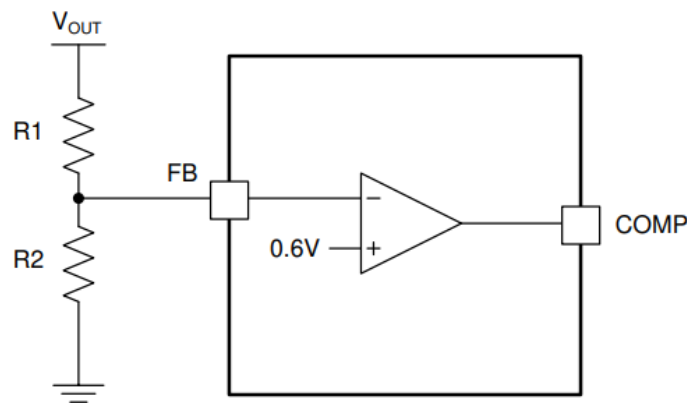
De voeding gebruikt in dit project is een TPS65265RHBT. Dit is een triple synchronous step-down (buck) converter met eeningangsspanning tussen de 4.5V-17V.

#### 7.3.2.1. Regelbare uitgangsspanning

Triple step-down wil dus zeggen dat deze buck converter 3 uitgangen heeft, deze zijn instelbaar tussen 0V en de ingangsspanning.

De volgende uitgangsspanningen zijn gekozen: +1.0V, +1.8V, +3.3V later wordt dit verder uitgelegd.

Deze zijn instelbaar door een spanningsdeler met een interne referentiespanning van 0.6V (Vref).



Figuur 28 Spanningsdeler (TPS65265 Datasheet)

De volgende formule geldt:

$$R2 = R1 * \frac{V_{ref}}{V_{out} - V_{ref}}$$

| Uitgangsspanning (V) | R1 (kΩ) | R2(kΩ) |
|----------------------|---------|--------|
| 1.0                  | 10      | 15     |
| 1.8                  | 20      | 10     |
| 3.3                  | 10      | 2.2    |

### 7.3.2.2. Output Inductor selection

Spoelen zijn berekend volgens de volgende formule:

- L = Inductie (H)
- Vinmax = Maximum ingangsspanning (17V)
- Vout = Uitgangsspanning (V)
- Io = Maximale uitgangsstroom (A)
- LIR = 30% van maximale uitgangsstroom (A)
- Fsw = Switching frequency (612 kHz)

$$L = \frac{V_{inmax} - V_{out}}{I_o * LIR} * \frac{V_{out}}{V_{inmax} * f_{sw}}$$

$$0.2\mu H = L1 = \frac{17V - 1.0V}{5A * 1.5A} * \frac{1.0V}{17V * 612kHz}$$

$$2.2\mu H = L2 = \frac{17V - 1.8V}{2A * 0.6A} * \frac{1.8V}{17V * 612kHz}$$

$$3.6\mu H = L1 = \frac{17V - 3.3V}{2A * 0.6A} * \frac{3.3V}{17V * 612kHz}$$

Vanwege BOM consideration werd alles 4.7uH genomen, dit is ook de aanbeveling uit de datasheet.

### 7.3.2.3. Output capacitor selection

Condensators zijn berekend volgens de volgende formule:

- Cout = Output capacitance (C)
- ΔIout = De verandering in uitgangsstroom (I)
- Fsw = Switching frequency (kHz)
- ΔVout = De verandering in uitgangsspanning(V)

$$C_{OUT} = \frac{2 * \Delta I_{out}}{f_{sw} * \Delta V_{out}}$$

$$22\mu F = C1 = \frac{2 * 5A}{612kHz * 1.0V}$$

$$12\mu F = C2 = \frac{2 * 2A}{612kHz * 1.8V}$$

$$22\mu F = C3 = \frac{2 * 2A}{612kHz * 3.3V}$$

Vanwege BOM consideration werd alles 22uF genomen, ook de recommendation uit de datasheet.

### 7.3.2.4. PGOOD

De PGOOD-pin is een open drain output die laag wordt getrokken zolang de uitgangsspanningen niet binnen hun respectabele uitgangswaardes liggen. Zodra alle uitgangsspanningen binnen hun respectabele uitgangswaardes liggen zal de interne pull-down weerstand losgekoppeld worden.

In dit project is een N-channel mosfet voorzien om een led uit te schakelen als er een fout optreedt.

### 7.3.2.5. Automatische power-up sequence

Een heel handig feature van de TPS65265 is dat deze beschikt over een automatische power-up sequence, dit is bijzonder handig bij schakelingen die werken met een FPGA, later meer uitleg hierover.

|   | MODE                     | EN1                          | EN2                          | EN3                                      | START SEQUENCING      | SHUTDOWN SEQUENCING   |
|---|--------------------------|------------------------------|------------------------------|--|-----------------------|-----------------------|
| <b>Automatic Power Sequencing</b>       | Connect to GND           | High                         | High                         | Used to start and stop bucks in sequence | buck1 → buck2 → buck3 | buck3 → buck2 → buck1 |
|   | Connect to GND           | Low                          | High                         |  | buck2 → buck1 → buck3 | buck3 → buck1 → buck2 |
|   | Connect to GND           | High                         | Low                          |  | buck2 → buck3 → buck1 | buck1 → buck3 → buck2 |
|   | Connect to high or float | High                         | High                         |  | buck1 → buck3 → buck2 | buck2 → buck3 → buck1 |
|   | Connect to high or float | Low                          | High                         |  | buck3 → buck1 → buck2 | buck2 → buck1 → buck3 |
|   | Connect to high or float | High                         | Low                          |  | buck3 → buck2 → buck1 | buck1 → buck2 → buck3 |
|   | Connect to GND           | Low                          | Low                          | Reserved                                 | Reserved              | Reserved              |
|   | Connect to high or float | Low                          | Low                          | Reserved                                 | Reserved              | Reserved              |
| <b>Externally Controlled Sequencing</b> | Floating                 | Used to start and stop buck1 | Used to start and stop buck2 | Used to start and stop buck3             | x                     | x                     |

Figuur 29 Power-up sequence (TPS65265 Datasheet)

De volgende aansluitingen werden gemaakt:

- *MODE* = GND
- *EN1* = *Vin*
- *EN2* = *Vin*
- *EN3* = *Vin*

Dit zorgt ervoor dat zodra de ingangsspanning wordt aangesloten de volgende uitgangsspanningen in volgorde tot stand komen:

*Start Sequencing:* Buck1 (+1.0V) => Buck2 (+1.8V) => Buck3 (+3.3V)

*Shutdown Sequencing:* Buck3 (+3.3V) => Buck2 (+1.8V) => Buck1 (+1.0V)

Moest dit feature nu niet aanwezig zijn, wat het geval is bij de meeste triple buck converters, dan was een externe sequence nodig. Dit zou direct de kostprijs en de BOM verhogen aangezien de eerste spanning +1.0V dan aangesloten zou moeten worden op de EN2 pin. Hiervoor is een extra schakeling nodig om de spanning te verhogen tot 1.15V (=  $V_{ENXH}$ ), aangezien +1.0V niet gezien wordt als een hoog signaal.

### 7.3.3. FPGA Power

#### 7.3.3.1. Recommended Operating Conditions

De datasheet “DC and AC Switching Characteristics (DS189)” van Xilinx werd gevolgd voor de Operating Conditions te bepalen:

| Symbol                             | Description   | Min   | Typ  | Max                     | Units |
|------------------------------------|---|-------|------|-------------------------|-------|
| <b>FPGA Logic</b>                  |   |       |      |                         |       |
| V <sub>CCINT</sub> <sup>(3)</sup>  | For -2 and -1 (1.0V) devices: internal supply voltage.  | 0.95  | 1.00 | 1.05                    | V     |
|                                    | For -1L (0.95V) devices: internal supply voltage.   | 0.92  | 0.95 | 0.98                    | V     |
| V <sub>CCAUX</sub>                 | Auxiliary supply voltage.   | 1.71  | 1.80 | 1.89                    | V     |
| V <sub>CCBRAM</sub> <sup>(3)</sup> | For -2 and -1 (1.0V) devices: block RAM supply voltage.   | 0.95  | 1.00 | 1.05                    | V     |
|                                    | For -1L (0.95V) devices: block RAM supply voltage.  | 0.92  | 0.95 | 0.98                    | V     |
| V <sub>CCO</sub> <sup>(4)(5)</sup> | Supply voltage for HR I/O banks.  | 1.14  | –    | 3.465                   | V     |
| V <sub>IN</sub> <sup>(6)</sup>     | I/O input voltage.  | –0.20 | –    | V <sub>CCO</sub> + 0.20 | V     |
|                                    | I/O input voltage (when V <sub>CCO</sub> = 3.3V) for V <sub>REF</sub> and differential I/O standards except TMDS_33. <sup>(7)</sup> | –0.20 | –    | 2.625                   | V     |
| I <sub>IN</sub> <sup>(8)</sup>     | Maximum current through any pin in a powered or unpowered bank when forward biasing the clamp diode.                                | –     | –    | 10                      | mA    |
| V <sub>CCBATT</sub> <sup>(9)</sup> | Battery voltage.  | 1.0   | –    | 1.89                    | V     |
| <b>XADC</b>                        |   |       |      |                         |       |
| V <sub>CCADC</sub>                 | XADC supply relative to GNDADC.   | 1.71  | 1.80 | 1.89                    | V     |
| V <sub>REFP</sub>                  | Externally supplied reference voltage.  | 1.20  | 1.25 | 1.30                    | V     |
| <b>Temperature</b>                 |   |       |      |                         |       |
| T <sub>j</sub>                     | Junction temperature operating range for commercial (C) temperature devices.  | 0     | –    | 85                      | °C    |
|                                    | Junction temperature operating range for industrial (I) temperature devices.  | –40   | –    | 100                     | °C    |

Figuur 30 Xilinx 7 Series: Recommended Operating Conditions (DS189)

Hieruit kan je bepalen dat de volgende 3 voedingsspanningen nodig zijn: 1.0V, 1.8V, 3.3V.

#### 7.3.3.2. Power-On/Off Power Supply Sequencing

De aanbevolen power-on sequence is V<sub>CCINT</sub>, V<sub>CCBRAM</sub>, V<sub>CCAUX</sub> en V<sub>CCO</sub> om een minimaal stroomverbruik te bereiken en ervoor te zorgen dat de I/O's bij het inschakelen 3-state zijn. De aanbevolen uitschakelvolgorde is de omgekeerde volgorde van het inschakelen.

### 7.3.3.3. Required PCB Capacitor Quantities per Device: Spartan-7 Devices

| Package | Device | V <sub>CCINT</sub> |             |             |              | V <sub>CCBRAM</sub> |            |             |              | V <sub>CCAUX</sub> |             |              | V <sub>CCO</sub> (per Bank)                 |             |              |
|---------|--------|--------------------|-------------|-------------|--------------|---------------------|------------|-------------|--------------|--------------------|-------------|--------------|---|-------------|--------------|
|         |        | 330 $\mu$ F        | 100 $\mu$ F | 4.7 $\mu$ F | 0.47 $\mu$ F | 100 $\mu$ F         | 47 $\mu$ F | 4.7 $\mu$ F | 0.47 $\mu$ F | 47 $\mu$ F         | 4.7 $\mu$ F | 0.47 $\mu$ F | 47 $\mu$ F or<br>100 $\mu$ F <sup>(3)</sup> | 4.7 $\mu$ F | 0.47 $\mu$ F |
| CPGA196 | XC7S6  | 0                  | 1           | 1           | 1            | 0                   | 1          | 1           | 1            | 1                  | 1           | 2            | 1   | 2           | 4            |
| CSGA225 | XC7S6  | 0                  | 1           | 1           | 1            | 0                   | 1          | 1           | 1            | 1                  | 1           | 2            | 1   | 2           | 4            |
| FTGB196 | XC7S6  | 0                  | 1           | 1           | 1            | 0                   | 1          | 1           | 1            | 1                  | 1           | 2            | 1   | 2           | 4            |
| CPGA196 | XC7S15 | 0                  | 1           | 2           | 2            | 0                   | 1          | 1           | 1            | 1                  | 1           | 2            | 1   | 2           | 4            |
| CSGA225 | XC7S15 | 0                  | 1           | 2           | 2            | 0                   | 1          | 1           | 1            | 1                  | 1           | 2            | 1   | 2           | 4            |
| FTGB196 | XC7S15 | 0                  | 1           | 2           | 2            | 0                   | 1          | 1           | 1            | 1                  | 1           | 2            | 1   | 2           | 4            |
| CSGA225 | XC7S25 | 0                  | 1           | 2           | 3            | 0                   | 1          | 1           | 1            | 1                  | 1           | 2            | 1   | 2           | 4            |
| CSGA324 | XC7S25 | 0                  | 1           | 2           | 3            | 0                   | 1          | 1           | 1            | 1                  | 1           | 2            | 1   | 2           | 4            |
| FTGB196 | XC7S25 | 0                  | 1           | 2           | 3            | 0                   | 1          | 1           | 1            | 1                  | 1           | 2            | 1   | 2           | 4            |

Figuur 31 Required PCB Capacitor Quantities per Device: Spartan-7 Devices (UG483)

Voor decoupling is de datasheet recommendation gevolgd.

De packaging van de condensators is 0402 gekozen waar mogelijk.

### 7.3.4. FTDI

Om een FTDI-apparaat te laten herkennen als een USB to JTAG programmer in Xilinx softwaretools zoals Vivado of Vitis moet de EEPROM op het FTDI-apparaat geprogrammeerd zijn met aangepaste firmware geleverd door Xilinx.

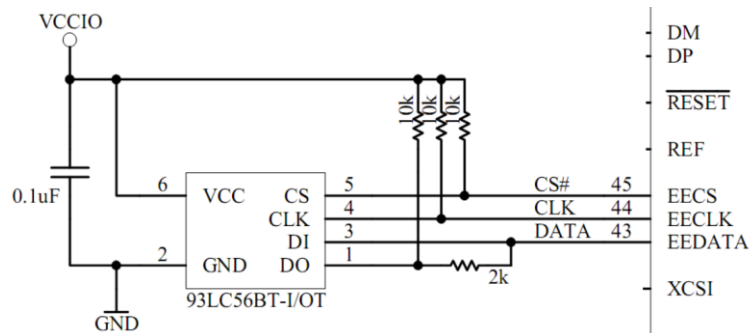
Het programmeren van het FTDI-apparaat wordt gedaan door het commando “program\_ftdi” te gebruiken in de Vivado Tcl-console. Eenmaal geprogrammeerd wordt het FTDI-apparaat in Vivado en Vitis herkend als een geldige Xilinx programmeerkabel.

De volgende FTDI-apparaten worden ondersteund: (zie UG908 p. 355)

- FT232H
- FT2232H
- FT4232H

De FT2232HQ is voor dit project gekozen, deze zal zich dus gaan gedragen als JTAG programmer en zorgt ook voor de Serial to USB-communicatie.

De Xilinx firmware configuratie moet worden opgeslagen op een externe EEPROM, volgens de datasheet van de FT2232H wordt een 93LC46/56/66 EEPROM aangeraden.



Figuur 32 EEPROM Interface, FT2232H Datasheet

Er moet een extern 12Mhz crystal voorzien worden, hiervoor zijn de volgende crystal load capacitors berekent:

- Cl = Crystal load capacitance
- Cs = Stray board capacitance

$$CL1 = CL2 = 2 * (Cl - Cs)$$
$$CL1 = CL2 = 2 * (10 - 2) = 16pF$$

Vanwege BOM consideration is er gekozen voor 22pF condensatoren.

#### 7.3.4.1. JTAG

JTAG-communicatie gebeurt over poort A van de FTDI-chip. Deze gaat ervoor zorgen dat we de FPGA kunnen programmeren.

#### 7.3.4.2. UART

UART-communicatie gebeurt over poort B van de FTDI chip deze wordt gebruikt om bijvoorbeeld data te zenden naar een computer zijn serial monitor, bijvoorbeeld Putty.

Er zijn 2 leds voorzien om USB-communicatie te visualiseren, Tx led en Rx led.

### 7.3.5. FPGA Configuratie

Op dit schema wordt Bank 0 van de FPGA weergegeven met bijhorende opslag, deze bepaald de programmeer modus en configuratie.

#### 7.3.5.1. Bootmode

De pinnen M [2:0] dienen om de configuratie mode te selecteren, deze worden als volgt bepaald:

Table 2-1: 7 Series FPGA Configuration Modes

| Configuration Mode          | M[2:0] | Bus Width                   | CCLK Direction |
|-----------------------------|--------|-----------------------------|----------------|
| Master Serial               | 000    | x1                          | Output         |
| Master SPI                  | 001    | x1, x2, x4                  | Output         |
| Master BPI                  | 010    | x8, x16                     | Output         |
| Master SelectMAP            | 100    | x8, x16                     | Output         |
| JTAG                        | 101    | x1                          | Not Applicable |
| Slave SelectMAP             | 110    | x8, x16, x32 <sup>(1)</sup> | Input          |
| Slave Serial <sup>(2)</sup> | 111    | x1                          | Input          |

Figuur 33 Bootmode configuration, (UG470)

In het schema is een jumper voorzien om te wisselen tussen JTAG BOOT en MASTER SPI.

JTAG Boot zal de opgeladen firmware opslagen in het vluchtig geheugen van de FPGA, bij stroomverlies gaat de opgeslagen firmware dus verloren.

MASTER SPI zal de firmware opslagen in de externe Flash Memory, bij stroom verlies blijft deze opgeslagen. Bij de volgende power-on wordt de firmware automatisch ingeladen in de FPGA.

#### 7.3.5.2. JTAG

Via JTAG kunnen we onze firmware in de FPGA inladen, deze bestaat uit volgende pinnen:

- TCK: Clock voor alle devices op een JTAG-chain
- TMS: Mode select voor alle apparaten op een JTAG-chain
- TDI: JTAG-chain serial data input
- TDO: JTAG-chain serial data output

Er zijn 50Ω weerstanden voorzien tegen ringing.



### **7.3.5.3. Flash Memory**

Om onze firmware niet te verliezen bij het uitschakelen van de stroom moet deze extern ergens opgeslagen worden, hier is gekozen voor een 16Mbit (2M \* 8bit) Serial NOR flash memory. Deze is gekozen uit een lijst met ondersteunde flash memory, je kan deze terugvinden in UG908 pagina 375.

Gekozen Flash Memory: MACRONIX MX25V1635FM2I

Deze communiceert over QSPI, als de boot modus MASTER SPI is geselecteerd gaat de FPGA bij zijn power up automatisch de opgeslagen firmware uit de flash memory halen. Bij een geslaagde firmware upload gaat de led "DONE" branden.

Een tweede flash memory is nog voorzien om andere data in op te slaan bijvoorbeeld de informatie van geregistreerde voertuigen.

### **7.3.5.4. Configuration Banks Voltage Select**

**CFGBVS** bepaald de I/O werkspanning en tolerantie voor bank 0 en de multi-function configuration pins in bank 14 en 15, bijvoorbeeld de automatische communicatie met flash memory gebruikt deze spanning.

Stel we moeten communiceren naar een flash memory met een werkspanning van 1.8V dan wordt CFGBVS best aan 1.8V gehangen, in dit geval 3.3V.

### **7.3.5.5. PROGRAM**

De laag actieve program pin dient om de geconfigureerde logica te resetten. Ik heb ervoor gekozen om deze hoog te hangen aangezien deze softwarematig kan gereset worden bij het inladen van nieuwe firmware of door een power reset in JTAG BOOT modus.

### 7.3.6. FPGA BANK 14

De FPGA bestaat uit verschillende banken die verschillende features, clocken en spanningen kunnen hebben. Voor mijn toepassing werken alle banken op 3.3V met een klokfrequentie van 100Mhz.

Alle I/O pinnen in een bank zijn geroute als differential pair, er is dus altijd een \_P en een \_N in een pair. Er zijn ook 4 verschillende timers in een bank, T0 tot en met T3, deze kunnen dienen als bijvoorbeeld interrupt op timer-basis T2.

Een bank kan zoals eerder vermeld op een andere spanning werken als andere banken, een voorbeeld hiervan is als je extra RAM wilt voorzien in je project kan je gebruik maken van een externe DDR3 RAM ic, deze werkt op 1.8V.

Hiervoor wordt meestal 1 bank gekozen om de volledige communicatie te voorzien met de DDR3 RAM, als je deze bank zijn voeding dan aansluit op 1.8V zal deze ook 1.8V signalen sturen.

Xilinx heeft een tool voorzien in de IP-catalog die toelaat je specificaties in te geven voor jou toepassing waarna deze vervolgens een lijst geeft met ondersteunde DDR3 RAM ic's.

Als je kiest voor 1 van de ondersteunde ic's krijg je een voorgestelde pinout en wordt alle communicatie automatisch door de FPGA geregeld en wordt dit gezien als extra BRAM/DRAM in de FPGA, softwarematig moet je dus geen code toevoegen om de DDR3 RAM te benutten.

Op Bank 14 vindt de communicatie: QSPI (flash), UART en serial plaats.

#### 7.3.6.1. 100Mhz Clock

Ook is de algemene clock van 100Mhz hier aangesloten op een MRCC pin, MRCC staat voor Multi-Region-Capable-Clocking pin deze voorziet alle timers en de FPGA van een clock source. Intern is een PLL voorzien om eventueel frequentie te regelen.

Het is Belangrijk dat er gekozen wordt voor een precisie crystal, ik heb hier de keuze gemaakt om een 100Mhz oscillator te gebruiken.

Zorg ervoor dat je zeker de MRCC\_P aansluit, aangezien MRCC\_N enkel werkt bij een differentieel signaal en niet bij een singel ended signaal zoals bij deze oscillator.

#### 7.3.6.2. PUDC

De laag actieve PUDC-pin staat voor Pull-Up During Configuration, deze bepaald of intern alle I/O omhoog of omlaag worden getrokken bij power-up.

Ik heb gekozen voor interne pull-up, laat deze pin zeker niet floaten voor en tijdens de configuratie!

### 7.3.7. FPGA BANK 15

Op bank 15 zijn alle ADC-features aanwezig, hier worden dus alle analoge signalen aangesloten. Alle SPI-communicatie met de radar module bevindt zich ook hier.

#### 7.3.7.1. ADC

Zoals eerder vermeld is een Analooq-Digitaal Converter nodig om van een analoog signaal naar een digitale waarde te gaan.

In de FPGA is een interne ADC aanwezig, de aansluitingen bevinden zich op bank 15, deze heeft de volgende specificaties:

#### 7.3.7.2. ADC channels

Er zijn 11 ADC channels aanwezig op de FPGA elk voorzien als differential pair. Alle signalen die worden gemeten in dit project zijn single-ended enkel de `_P` pin van het pair wordt aangesloten, `_N` is altijd GND.

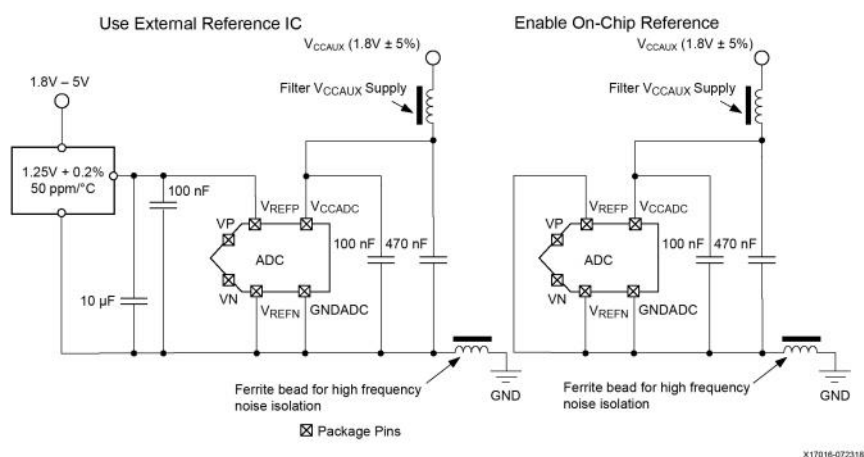
Daarnaast zijn er nog enkele on-chip sensors die temperatuur en interne spanningen monitoren.

#### 7.3.7.3. Analoge Inputs

De analoge inputs kunnen een signaal tussen 0V en 1V meten, later wordt er nog uitgelegd hoe we tot 1.5V kunnen meten en welke verschillende configuraties er mogelijk zijn voor de ADC. Voor onze toepassingen worden alle signalen naar een spanningsniveau tussen 0V en 1V gebracht via een spanningsdeler. Intern zijn clamping diodes voorzien om te beschermen tegen overspanning.

#### 7.3.7.4. Reference spanning

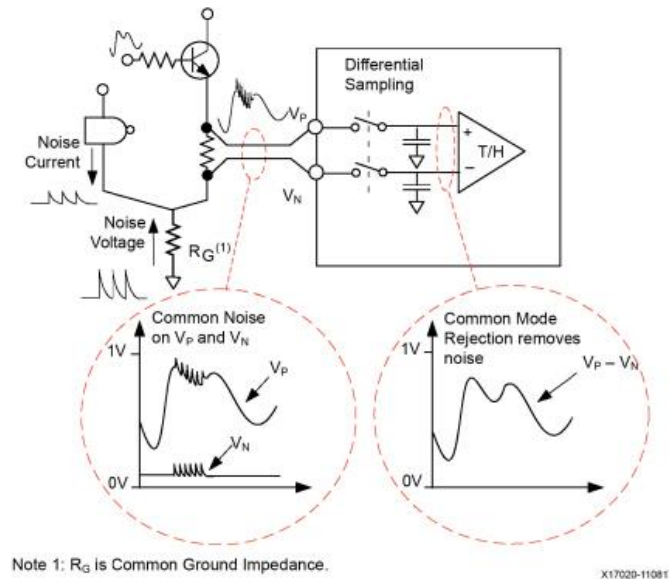
Er zijn 2 configuraties om de ADC te voorzien van een reference spanning, de eerste (links) maakt gebruik van een externe spanning, dit biedt betere performance en thermische weerbaarheid. Of de interne reference spanning (rechts) wordt gebruikt, voor de doeleinden van dit project is de interne reference spanning voldoende accuraat.



Figuur 34 Reference Voltage XADC, (UG480)

### 7.3.7.5. Differential sampling

Alle analoge inputs zijn voorzien als differential pair, dit zorgt ervoor dat het mogelijk is om common-mode noise signalen te gaan wegfilteren zoals onderaan weergegeven.



Figuur 35 Common Mode Noise Rejection, (UG480)

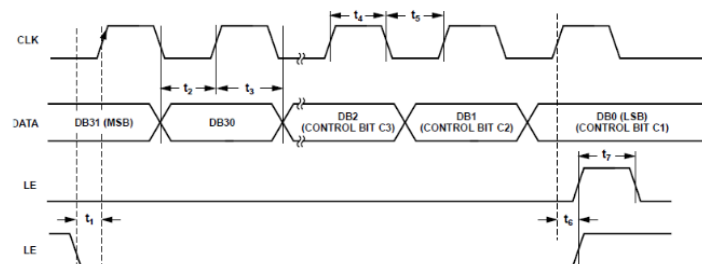
### 7.3.7.6. SPI

Alle SPI-communicatie met de IVQ-3005 gebeurt via bank 15. Deze dient om de on board PLL te programmeren namelijk de ADF4158.

De ADF4158 heeft 8 verschillende registers, de configuratie van deze registers bepaald het uitgezonden FMCW signaal. Enkele instelbare parameters: zendfrequentie, ramp up speed, delay interval... deze parameters zijn belangrijk bij het bepalen van afstand en snelheid zoals uitgelegd in het digital signal processing hoofdstuk.

De maximum SPI clock rate is 2Mhz. Hieronder is de minimaal te respecteren timing te zien.

|     |           |                        |
|-----|-----------|------------------------|
| t1: | 40 ns min | LE setup time          |
| t2: | 20 ns min | Data to CLK setup time |
| t3: | 20 ns min | Data to CLK hold time  |
| t4: | 50 ns min | CLK high duration      |
| t5: | 50 ns min | CLK low duration       |
| t6: | 20 ns min | CLK to LE setup time   |
| t7: | 40 ns min | LE pulse width         |



Figuur 36 SPI write timing IVQ-3005

### 7.3.8. FPGA BANK 34

Op bank 34 is er gekozen om alle General Purpose Inputs & Outputs (GPIO's) aan te sluiten. Zoals eerder vermeld is de interne pull-up (PUDC) enabled.

#### 7.3.8.1. Dip Switches

Er zijn 8 dip switches voorzien, deze kunnen dienen om een bepaalde configuratie in te stellen, bijvoorbeeld: radar maximum range instellen, enkel voertuigen die in 1 rijrichting rijden te meten...

#### 7.3.8.2. LEDS

Light Emitting Diodes (LED's) kunnen gebruikt worden als indicator en zijn handig tijdens het testen. De stroom door de Leds kan als volgt berekend worden wetende dat de spanning die minimaal over de LED moet staan 2.5V is:

- $I_{led}$  = stroom door de LED.
- $V_{cc}$  = output spanning FPGA.
- $V_{led}$  = de spanning die over de LED komt te staan (zie datasheet LED)
- $R$  = voorschakel weerstand.

$$I_{led} = \frac{V_{cc} - V_{led}}{R} = \frac{3.3V - 2.5V}{470\Omega} = 1.7mA$$

#### 7.3.8.3. Testpoints

Enkele niet gebruikte pinnen zijn nog naar buiten gebracht als testpoints deze kunnen later gebruikt worden als testpinnen of deze kunnen nog dienen voor eventuele uitbreiding, bijvoorbeeld om draadloze communicatie toe te voegen.

De test pinnen zijn voorzien van ESD protection en een 50Ω serie weerstand tegen ringing/overstroom.

### 7.3.9. Radar Module

De radar module die gebruikt wordt in dit project is de IVQ-3005 van InnoSent, op dit schema kan je alle aansluitingen tussen de PCB en de module terugvinden.

#### 7.3.9.1. Power Supply IVQ-3005

De IVQ-3005 radar module specificeert een voedingsspanning van minimaal 5.5V tot maximaal 6.1V met een maximaal toelaatbare ripple spanning van 10mVpp.

Er is gekozen om eerst de ingangsspanning omlaag te halen via een low-dropout linear voltage regulator (LDO, LDL1117S50R) deze zal een ingangsspanning tussen 5V en 18V omlaaghalen naar een 5V DC, als er een ingangsspanning van 5V is aangesloten zal er een dropout voltage van 300mV zijn op de uitgang dus 4.7V Vout bij 5V Vin.

Hierna gaat deze spanning door een boost converter (ST8R00WPU) met een typical output ripple spanning van 5mVpp. Deze specificeert een ingangsspanning van minimaal 4V tot maximaal 6V. De uitgang van deze boost converter is instelbaar tussen de 6V en 12V, voor de radar module kiezen we de minimale spanning van 6V.

Dit wordt bepaald met volgende spanningsdeler:

- Vout = uitgangsspanning
- Vfb = feedbackspanning
- R1 = spanningsdeler weerstand 1
- R2 = spanningsdeler weerstand 2

$$V_{out} = V_{fb} * \left(1 + \left(\frac{R1}{R2}\right)\right) \quad 6V = 1.22V * \left(1 + \left(\frac{39K\Omega}{10K\Omega}\right)\right)$$

#### 7.3.9.2. InnoSent IVQ-3005 Connector

De Radar module heeft 2 connectoren, 1 voor digitale communicatie en 1 voor analoge communicatie.

De analoge connector heeft volgende aansluitingen:

- VCC: Power Supply, 6.1V DC max. 10mVpp ripple.
- Q1: IF-output Q1 (Rx0 antenne).
- I1: IF-output I1 (Rx0 antenne).
- Q2: IF-output Q2 (Rx1 antenne).
- I2: IF-output I2 (Rx1 antenne).
- ATT: Controle-spanning voor output power attenuator.
- TEMP: Output temperature sensor (LM20CIM).
- DET: Output power detector.
- Vtune: VCO tuning-voltage output.

De digitale connector heeft volgende aansluitingen:

- VCO\_EN: VCO enable >2V=ON, <0.4V=OFF.
- MUX\_TX: MUX-output PLL ADF4158.
- SPI\_CS\_TX: SPI chip select input voor PLL.
- SPI\_CS\_POT: SPI chip select input voor programmeerbare gain potentiometer.
- SPI\_MOSI: SPI input MOSI
- SPI\_CLK: SPI input CLK
- TX\_DATA: PLL-modulatie trigger input
- 30Mhz\_clk: 30Mhz reference clk output

### 7.3.9.3. Analoge uitgangen IVQ-3005

De Radar module heeft verschillende analoge uitgangen die gelezen moeten worden door de ADC van de FPGA, zoals eerder vermeld kan de ADC op de FPGA spanningen tussen 0V en 1V lezen echter liggen de waarden van de radar module hoger, daarom zijn er verschillende spanningsdeler voorzien.

Q1, I1, Q2, I2, DET liggen tussen 0V en 3V, volgende spanningsdeler is voorzien:

- $V_{out}$  = uitgangsspanning
- $V_{in}$  = ingangsspanning
- $R1$  = spanningsdeler weerstandswaarde 1
- $R2$  = spanningsdeler weerstandswaarde 2

$$V_{out} = \frac{V_{in} * R2}{(R1 + R2)} = 1.0V = \frac{3.0 * 10K\Omega}{(20K\Omega + 10K\Omega)}$$

De TEMP output komt van een LM20 temperatuur sensor en heeft een spanningswaarde tussen 0.3V en 2.5V, volgende spanningsdeler is voorzien:

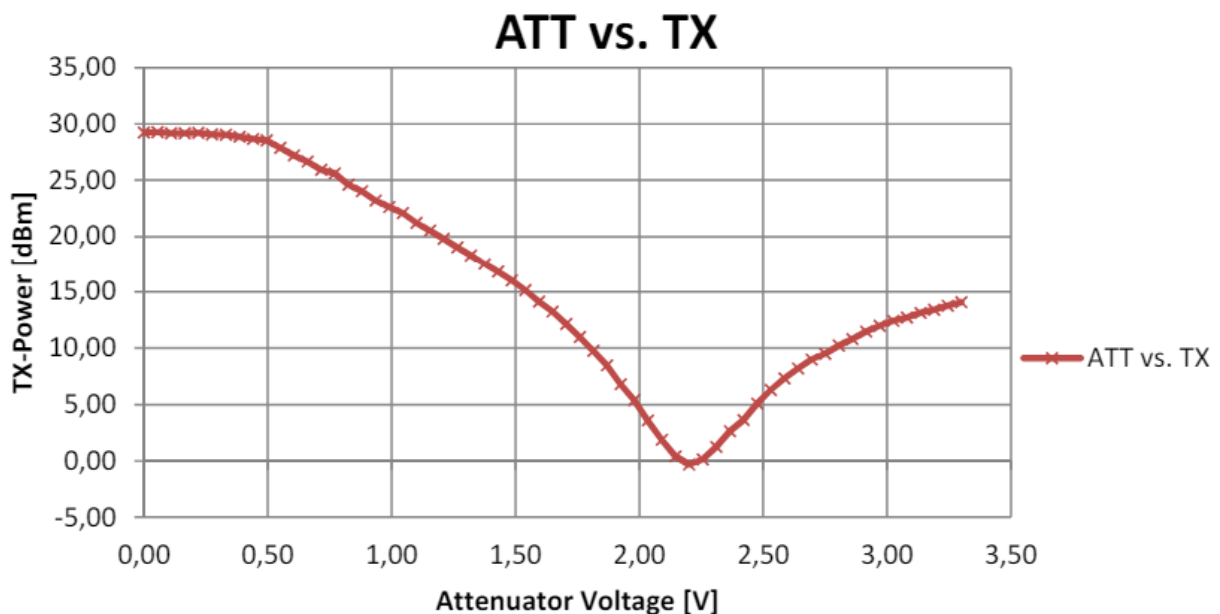
$$V_{out} = \frac{V_{in} * R2}{(R1 + R2)} = 1.0V = \frac{2.5 * 10K\Omega}{(15K\Omega + 10K\Omega)}$$

De Vtune output is een gebufferde output spanning van de PLL tuning loop, deze heeft een spanning tussen 0V en 5.4V. Volgende spanningsdeler is voorzien:

$$V_{out} = \frac{V_{in} * R2}{(R1 + R2)} = 1.0V = \frac{5.0 * 10K\Omega}{(40K\Omega + 10K\Omega)}$$

### 7.3.9.4. Analoge Ingangen IVQ-3005

De radar module heeft 1 analoge ingang ATT, deze ingang dient om het zendvermogen te gaan regelen. Het analoge signaal wordt gecreëerd door een 12-bit DAC die serieel wordt aangestuurd. Dit analogo signaal zit tussen 0V en 3.3V. Let op 0V = 29.22dBm



Figuur 37 Attenuator voltage vs TX-Power Example

### **7.3.9.5. Digitale uitgangen IVQ-3005**

De radar module heeft 2 digitale uitgangen, MUX\_TX en 30Mhz\_CLK.

MUX\_TX is de output van de interne PLL (ADF4158, een frequency synthesizer). Met deze pin kan de configuratie van de PLL gelezen worden.

30Mhz\_CLK is een single ended clock frequentie die naar buiten is gebracht. Hierop kan DSP gesynchroniseerd worden met de PLL.

### **7.3.9.6. Digitale Ingangen IVQ-3005**

De radar module heeft 2 digitale ingangen, VCO\_EN en TX\_DATA.

De TX\_DATA input kan gebruikt worden om een read commando van de PLL te activeren (ADF4158). De PLL zal bij een hoge puls op de TX\_DATA lijn zijn configuratie over de MUX\_TX pin uitzenden.

Het VCO-ENABLE signaal wordt gebruikt om de power supply te enablen van de Tx-VCO. Deze moet hoog zijn om te enablen, laag om te disablen

### **7.3.9.7. Digitale I/O IVQ-3005**

Communicatie met de radar module gebeurt op basis van SPI.

Er zijn 2 chip selects voorzien, 1 om de interne gain aan te sturen de ander om de PLL aan te sturen.



## 7.4. PCB

De Printed Circuit Board (PCB) is getekend in Altium Designer, dit is een 4-layer board bestaande uit FR4 materiaal met bottom en top assembly, bij productie is ENIG finishing aangeraden.

De standaard grootte voor een passieve component op deze print is 0603 gekozen waar mogelijk. Alle pads van componenten zijn voorzien van thermal relief om een goede soldering te garanderen en tombstoning te voorkomen.

Alle Altium PCB-documenten van layers tot 3D-model zijn terug te vinden onder 13.2 Bijlagen.

### 7.4.1. Layer Stack-up:

De volgende stack up is voorzien:

- 1) *Signal+GROUND*
- 2) *GROUND*
- 3) *POWER*
- 4) *Signal+GROUND*

| # | Name               | Material | Type           | Weight | Thickness | Dk  | Df   |
|---|--------------------|----------|----------------|--------|-----------|-----|------|
|   | Top Overlay        |          | Overlay        |        |           |     |      |
|   | Top Solder         | SM-001   | Solder Mask    |        | 0.0254mm  | 4   | 0.03 |
|   | Top Surface Finish | PbSn     | Surface Finish |        | 0.02mm    |     |      |
| 1 | L1 (Sig+GND)       | CF-004   | Signal         | 1oz    | 0.035mm   |     |      |
|   | Dielectric 1       | PP-017   | Prepreg        |        | 0.12954mm | 4.3 | 0.02 |
|   | Dielectric 2       | PP-017   | Prepreg        |        | 0.12954mm | 4.3 | 0.02 |
| 2 | L2 (GND)           |          | Signal         | 1oz    | 0.035mm   |     |      |
|   | Dielectric 3       | Core-039 | Core           |        | 0.7112mm  | 4.8 | 0.02 |
| 3 | L3 (PWR)           | CF-004   | Signal         | 1oz    | 0.035mm   |     |      |
|   | Dielectric 4       | PP-017   | Prepreg        |        | 0.12954mm | 4.3 | 0.02 |
|   | Dielectric 5       | PP-017   | Prepreg        |        | 0.12954mm | 4.3 | 0.02 |
| 4 | L4 (Sig+GND)       | CF-004   | Signal         | 1oz    | 0.035mm   |     |      |
|   | Bottom Surface...  | PbSn     | Surface Finish |        | 0.02mm    |     |      |
|   | Bottom Solder      | SM-001   | Solder Mask    |        | 0.0254mm  | 4   | 0.03 |
|   | Bottom Overlay     |          | Overlay        |        |           |     |      |

Figuur 38 PCB Layer-Stackup

## 7.4.2. Design Rules

Vanuit onze PCB manufacturer (in dit geval Multi-circuit-boards) krijgen we enkele parameters mee waaraan we ons moeten houden binnen het design. Het is niet verstandig om constant enkel de minimum parameters te gebruiken, beter is om deze enkel te gebruiken waar nodig, Daar is in dit project rekening mee gehouden.

Enkele minimale parameters van multi-cb (PCB-fabrikant):

**Via:**

- Diameter = 200  $\mu\text{m}$
- Via pad = 400  $\mu\text{m}$

**Conductor:**

- Width = 100  $\mu\text{m}$
- Spacing = 100  $\mu\text{m}$

De minimale parameters in dit design zijn:

**Via:**

- Diameter = 300  $\mu\text{m}$
- Via pad = 500  $\mu\text{m}$

**Conductor:**

- Width = 200  $\mu\text{m}$
- Spacing = 200  $\mu\text{m}$

Met enkele uitzonderingen waar nodig, later hier meer over.

### 7.4.3. Layout: Voeding

De footprint van de TPS65265 is een QFN-32 met een pitch van 0.5mm.

Voor deze kleine pitch is een room voorzien genaamd QFN, deze room heeft aangepaste design rules zoals een kleinere clearance van 100  $\mu\text{m}$  om deze zonder errors correct te kunnen aansluiten. Deze rules respecteren nog steeds de minimum constraints van de PCB manufacturer.

De layout van de voeding is overgenomen uit het layout example van de TPS65265.

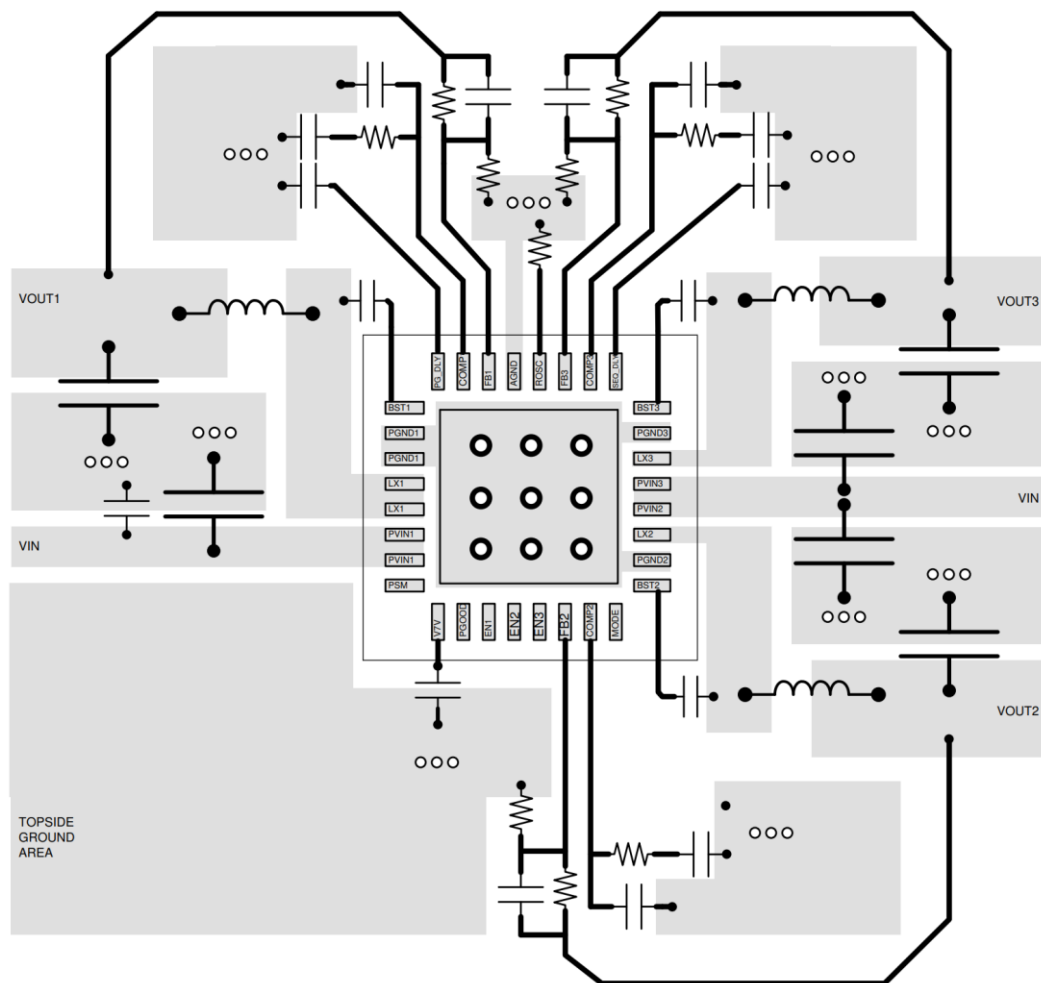
Deze houdt in gedachte om een zo kort mogelijke return path naar de ground te hebben.

TPS65265

SLVSD86B – DECEMBER 2015 – REVISED MAY 2023

INSTRUMENTS  
www.ti.com

#### 9.4.2 Layout Example



Figuur 39 Layout Example TPS65265

### 7.4.4. Layout: FTDI

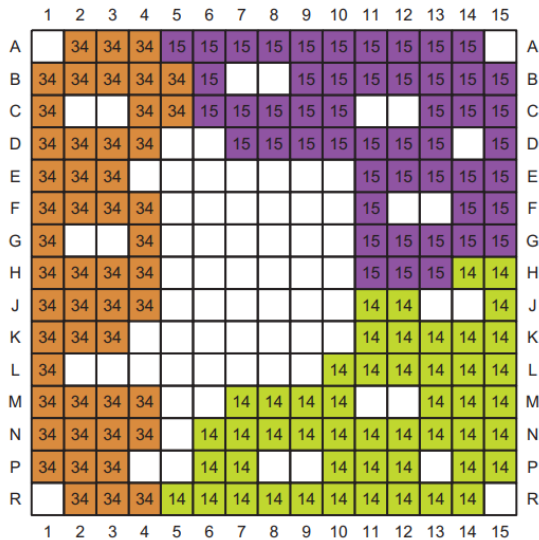
De FT2232HQ is een 64-QFN met 0.5mm pitch.

De USB-signalen die gebruikt worden zijn USB2.0 signalen hiervoor is geen impedance matching nodig wel zijn deze geroute als differential pair, moest er later naar USB3.0 overgegaan worden is impedance matching (90 $\Omega$  voor USB) aangeraden.

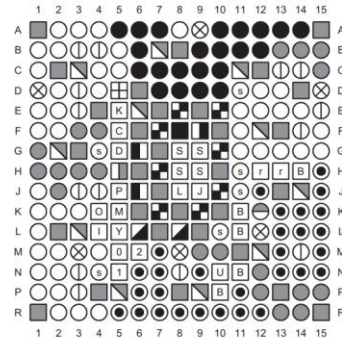
### 7.4.5. Layout: FPGA

De FPGA die gebruikt is in dit project is de XC7S25. De package, CSGA225, is een 15x15 0.8mm pitch BGA. (Pitch is de afstand, center-to-center, tussen 2 pads)

De layout kan gemakkelijk bepaald worden met onderstaande fotos. Zo is er gekozen om alle GPIO (leds, switches...) links van de FPGA te plaatsen aangezien deze op bank 34 zijn aangesloten die zich links bevindt op de FPGA.



Figuur 40 CSGA225 Package - Pinout diagram



| User I/O Pins   | Dedicated Pins   | Other Pins   |
|---|--|--|
| <ul style="list-style-type: none"> <li>IO_LXXY_#</li> <li>IO_XX_#</li> </ul>  | <ul style="list-style-type: none"> <li>CCLK_0</li> <li>CFGBVS_0</li> <li>DONE_0</li> <li>DXP_0</li> <li>DXN_0</li> <li>GNDADC_0</li> <li>INIT_B_0</li> <li>M0_0</li> <li>M1_0</li> <li>M2_0</li> <li>PROGRAM_B_0</li> <li>TCK_0</li> <li>TDI_0</li> <li>TDO_0</li> <li>TMS_0</li> <li>VCCADC_0</li> <li>VCCBATT_0</li> </ul> | <ul style="list-style-type: none"> <li>GND</li> <li>VCCAUX_IO_G#</li> <li>VCCAUX</li> <li>VCCINT</li> <li>VCCO_#</li> <li>VCCBRAM</li> <li>NC</li> </ul> |
| Multi-Function Pins   |  |  |
| <ul style="list-style-type: none"> <li>ADV_B</li> <li>FCS_B</li> <li>FOE_B</li> <li>MOSI</li> <li>FWE_B</li> <li>DOUT_CSO_B</li> <li>CSL_B</li> <li>PUDC_B</li> <li>RDWR_B</li> <li>RS0-RS1</li> <li>AD0P/AD0N-AD15P/AD15N</li> <li>EMCCLK</li> </ul> | <ul style="list-style-type: none"> <li>VRN</li> <li>VRP</li> <li>VREF</li> <li>D00-D31</li> <li>A00-A28</li> <li>DQS</li> <li>MRCC</li> <li>SRCC</li> </ul>  |  |

Figuur 41 CSGA225 Package - I/O Banks

De voornaamste factor die de complexiteit van BGA routing bepaald is de pitch, factoren zoals BGA array quantity, type solder mask en het aantal signal layers spelen ook een grote rol.

#### 7.4.5.1. Signal Layers FPGA

Met een pitch van 0.8mm wordt routing vrij ingewikkeld, de pitch kan bijvoorbeeld bepalen hoeveel layers je nodig hebt. Je kan de layer count estimation uitrekenen met de volgende formule:

$$\text{Layers} = \frac{\text{Signals}}{\text{Routing Channels} * \text{Routes Per Channel}}$$

Xilinx heeft dit al uitgerekend voor ons en in een tabel geplaatst: **Estimated Signal Layers Required**

| BGA Pins | Ball (mm) | Pitch | Routing Channels | One Trace per routing channel | Two traces per routing channel |
|----------|-----------|-------|------------------|-------------------------------|--------------------------------|
| 225      | 0.8       |       | 56               | 2                             | 1                              |

(UG1099)

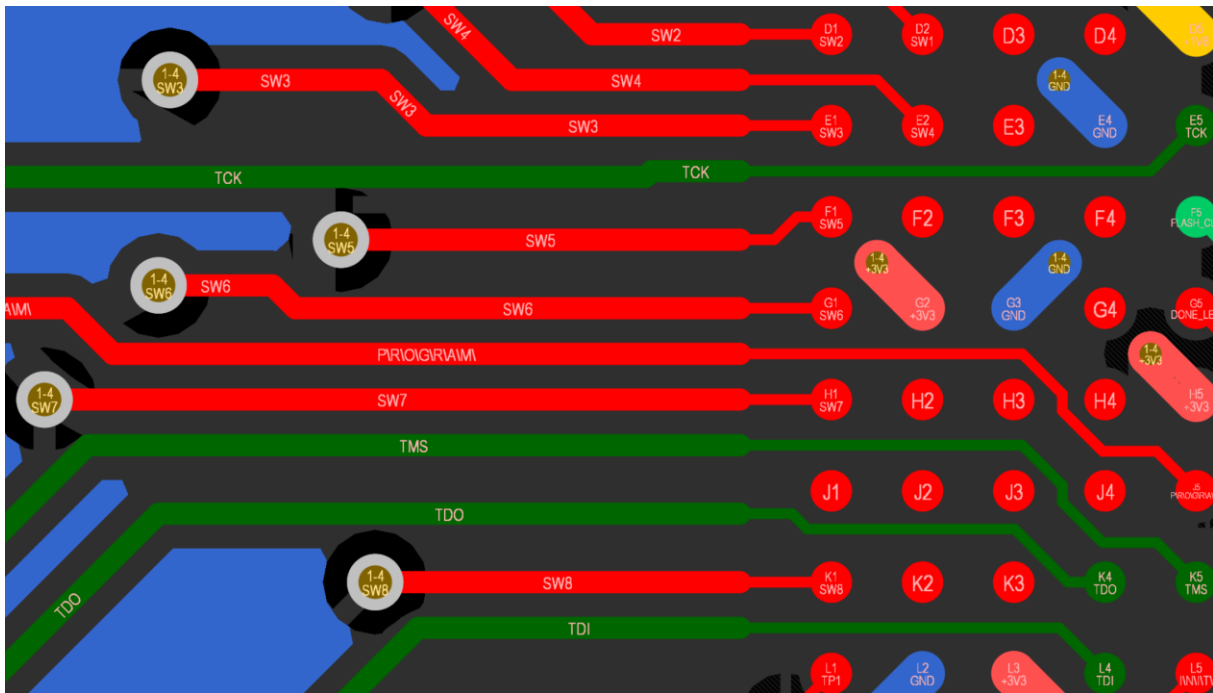
Daarom is er in dit project gekozen voor een 4-layer PCB.

### 7.4.5.2. Routing FPGA

Aangezien er weinig ruimte aanwezig is onder de FPGA-footprint is er gekozen om een room genaamd BGA aan te maken met andere design rules om routing efficiënter te maken.

- **Via's tot diameter:0.4mm, hole size:0.2mm**
- **Width/clearance tot 0.1mm**

Zo kan je zien dat zodra er de BGA room verlaten er wordt overschakelen naar onze algemene design rules waar de width/clearance gelijk is aan 0.2mm.



*Figuur 42 RADAR PCB, BGA ROOM exit rules*

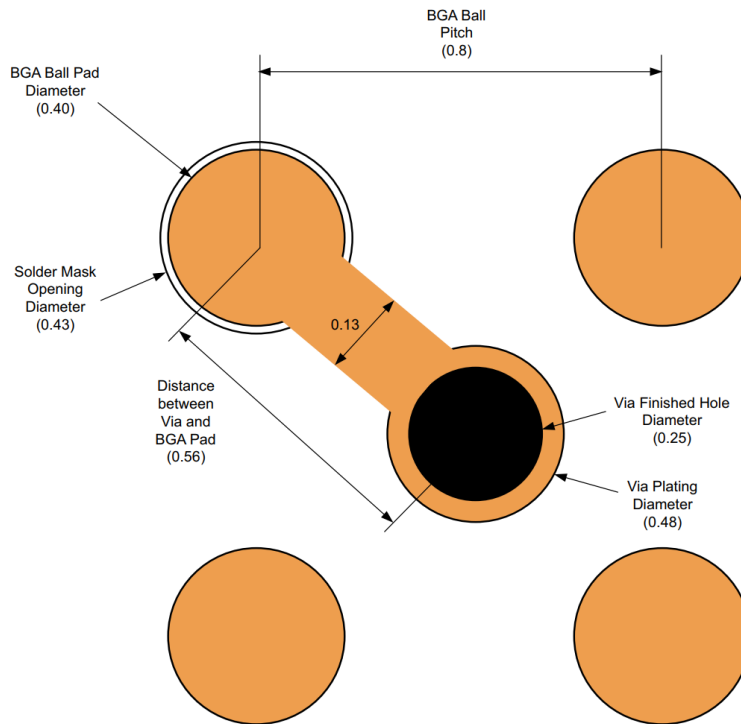
Om de traces naar buiten te routen is er gekozen voor de Dog-bone methode, deze heeft als voordeel dat dit kost efficiënt is maar wel het meest ingewikkelde om te routen. Andere methode zoals Blind Vias, Buried Vias en Via in Pad hebben extra fabricatie kosten maar zijn meestal wel eenvoudiger te routen.

Blind vias zijn vias die enkel van top of bottom naar een interne layers verbinden, deze zijn dus niet through hole (+20% tot +40% fabricatie kosten).

Buried vias zijn vias die enkel interne layers intern met elkaar verbindt, deze zijn dus niet zichtbaar op top of bottom layer (+25% tot +60% fabricatie kosten).

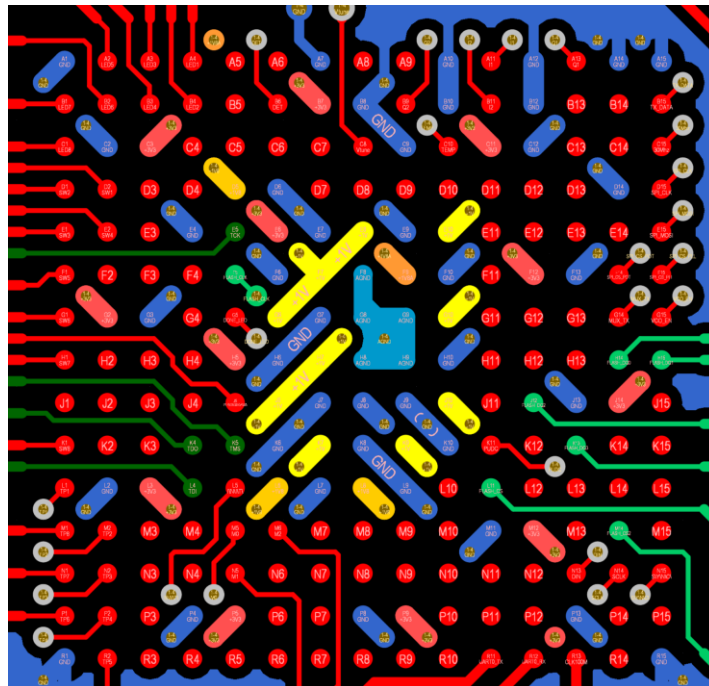
Via-in-Pad, er is even twijfel geweest of er via-in-pad zou gebruikt gaan worden aangezien dit niet heel duur is en er al veel positieve ervaringen zijn geweest hiermee. Bij via-in-pad ga je de pad die voorzien is voor connectie met de FPGA gebruiken als via waardoor je niet hoeft te dog-bonen en ruimte bespaart voor de traces (+10% tot +20% fabricatie kosten).

De Dog-bone methode heeft zijn naam gekregen vanwege zijn vorm, zoals onderaan te zien. Hierbij ga je vanuit je pad een trace trekken naar een via die geplaatst is tussen je pads.



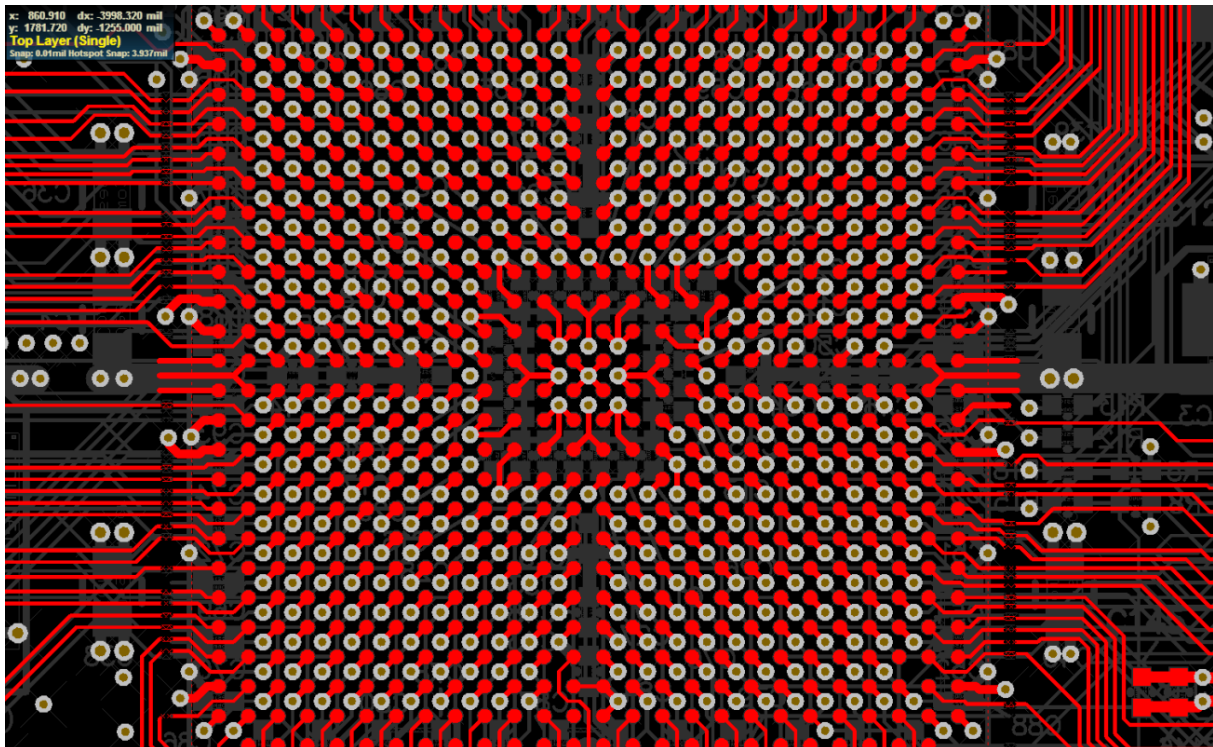
Figuur 43 Recommended Layout Dimensions within BGA Area for 0.8mm Pitch Devices (UG1099)

In dit design is gekozen om vias met een hole diameter van 0.2mm en pad diameter van 0.4mm te gebruiken en traces tot 0.1mm.



Figuur 44 BGA routing PCB

Voor de BGA room zijn ook design rules gedefinieerd om automatisch een fanout te laten maken voor de BGA-footprint. Tools > Fanout > Fanout Component. Hier zijn verschillende opties in, de voornaamste is dat alle gebruikte pads automatisch worden gedog-boned met uitzondering op de 2 uiterste pin arrays, deze worden automatisch naar de rand van de BGA-room geroute.



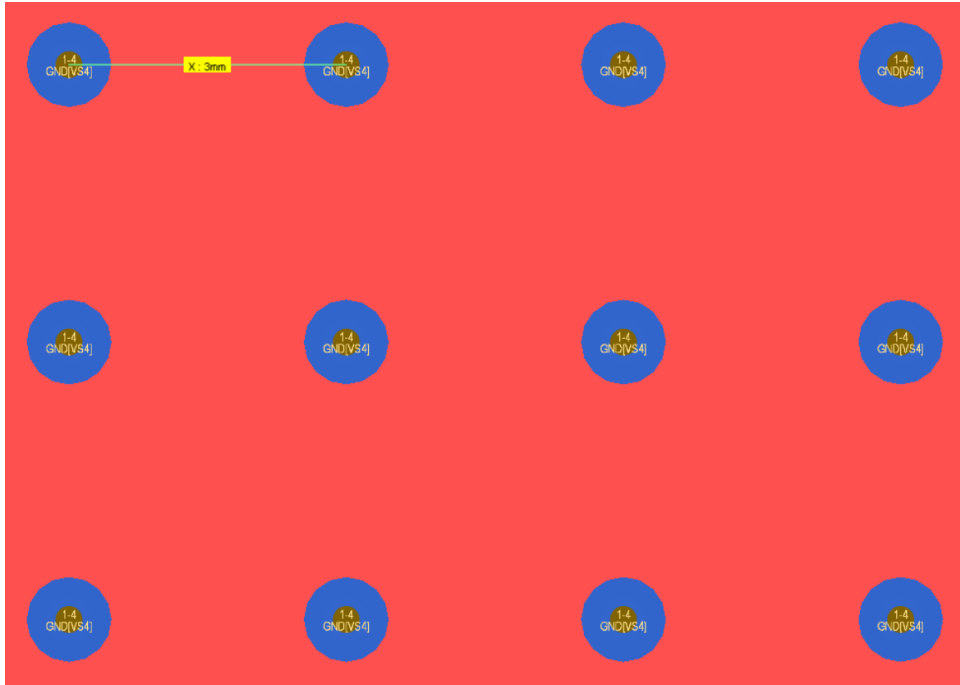
Figuur 45 voorbeeld [BGA Fanout](#).





### 7.4.6. Via stitching

Zoals eerder vermeld zijn er op de top en bottom layers naast de signal routing ook ground poors voorzien. Deze ground poors zijn aan elkaar ge via stitched, dit wil zeggen dat er met behulp van een tool in altium automatisch via's zijn geplaatst met een bepaalde spacing en grid die de ground planes met elkaar verbinden waar mogelijk. Dit zorgt voor een betere ground impedantie, return path en een vermindering in EMC straling.



*Figuur 50 Via stitching, gezien vanuit de power plane*

## 7.4.7. Design Rule Check

De Altium tool “Design Rule Check” geeft ons de mogelijkheid om te checken of alle op voorhand ingestelde regels gerespecteerd worden binnen het project, zo niet zal deze tool aangeven waar er fouten zijn. Daarnaast worden ook nog warnings gegeven over eventueel niet aangesloten pad's of kortsluitingen.

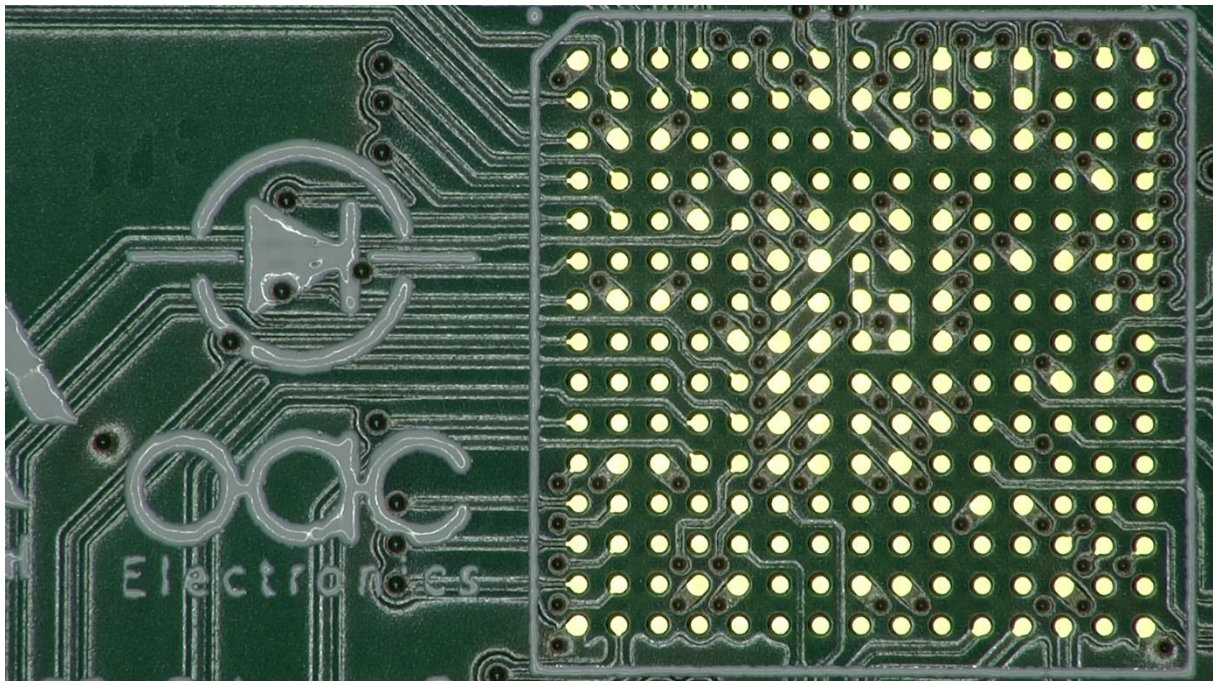
Design Rule Check geeft de volgende output voor dit project:

| Summary  |          |
|--|----------|
| Warnings   |          |
|  | Count    |
|  | Total 0  |
| Rule Violations  |          |
|  | Count    |
| Clearance Constraint (Gap=0.1mm) (WithinRoom('DEN')) (All)   | 0        |
| Clearance Constraint (Gap=0.2mm) (All) (All)   | 0        |
| Clearance Constraint (Gap=0.1mm) (WithinRoom('BGA')) (All)   | 0        |
| Short-Circuit Constraint (Allowed=No) (All) (All)  | 0        |
| Un-Routed Net Constraint (All)   | 0        |
| Modified Polygon (Allow modified: No) (Allow shelved: No)  | 0        |
| Width Constraint (Min=0.2mm) (Max=1mm) (Preferred=0.254mm) (All)   | 0        |
| Width Constraint (Min=0.1mm) (Max=0.5mm) (Preferred=0.1mm) (WithinRoom('BGA'))   | 0        |
| Power Plane Connect Rule(Relief Connect) (Expansion=0.508mm) (Conductor Width=0.254mm) (Air Gap=0.254mm) (Entries=4) (All) | 0        |
| Hole Size Constraint (Min=0.025mm) (Max=5mm) (All)   | 0        |
| Hole To Hole Clearance (Gap=0.254mm) (All) (All)   | 0        |
| Minimum Solder Mask Sliver (Gap=0.1mm) (All) (All)   | 7        |
| Silk To Solder Mask (Clearance=0.1mm) (aPad) (All)   | 0        |
| Silk to Silk (Clearance=0.1mm) (All) (All)   | 40       |
| Net Antennae (Tolerance=0mm) (All)   | 0        |
| Room CPN (Bounding Region = (84.042mm, 41.636mm, 89.573mm, 47.157mm)) (False)  | 0        |
| Room BGA (Bounding Region = (63mm, 67mm, 75.8mm, 79.8mm)) (False)  | 0        |
| Height Constraint (Min=0mm) (Max=25.4mm) (Preferred=12.7mm) (All)  | 0        |
|  | Total 47 |

Figuur 51 Design Rule Check output RADAR

Hieruit kunnen we afleiden dat er 7 openingen zijn in onze soldermask die een violation veroorzaken, deze zijn allemaal nagekeken en vormen geen probleem.

De andere 40 violations zijn veroorzaakt door silkscreen dat te dicht op elkaar zit, dit wordt vaak veroorzaakt door logo's en dergelijke, dit is geen enkel probleem.



Figuur 52 FPGA PCB-Footprint

### 7.4.8. Mechanical

De PCB is 87.6mm x 117.6mm groot, dit is exact even groot als de gebruikte radar module. De bedoeling is dus om deze PCB achter op de radar module te kunnen klikken, hiervoor zijn ook 4 mounting holes voorzien op exact dezelfde locatie als de mounting holes van de radar module.

Aangezien de radar module 1 grote common ground heeft, dus ook de mounting holes zijn geground, is er gekozen om de mounting holes van de PCB te platen en te verbinden met ground om deze common ground te respecteren.

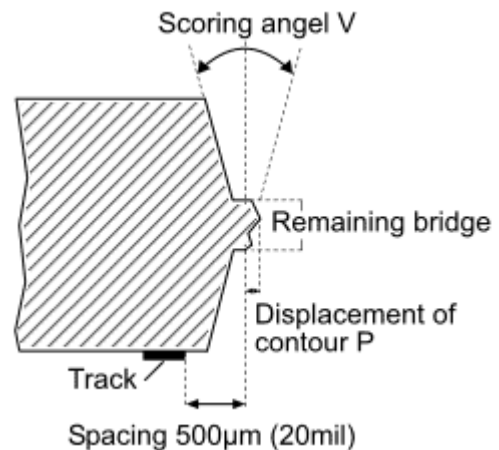
### 7.4.9. PCB Panel

Voor productie redenen is er ook een PCB-panel file voorzien voor eventuele latere massa productie. Op deze panel staan meerdere dezelfde PCB's die zijn gescheiden door een V-cut, fiducials kunnen voorzien worden op de extra border spacing.

Deze extra border space maakt het productieproces eenvoudiger aangezien het plaatsen van de PCB op de productie band gebeurt via een knijp systeem. Dit knijp systeem klampt zich vast aan de randen van de PCB.

Moest die border space niet aanwezig zijn dan was het belangrijk om op de uiterste 5-6mm van de PCB geen componenten te voorzien. Bij kleine high density projecten kan dit dus problemen veroorzaken.

Panelization is altijd aangeraden, wanneer V-Cut's niet mogelijk zijn worden mouse bites aangeraden.



Figuur 53 Voorbeeld V-Scoring (V-Cut)

### 7.4.10. Bill Of Materials

In de Bill Of Materials oftewel BOM staan alle gebruikte componenten met hun eigen ERP code uit de OAC-database als extra parameter. Deze ERP code wordt gebruikt bij het productieproces om een component te identificeren en op te slaan met zijn bijhorende specificaties.

### 7.4.11. Production Variant

Altium heeft als functie om een productie variant toe te voegen aan het project. Zo is het mogelijk om bepaalde componenten te excluderen uit de BOM, dit is vaak handig bij through hole componenten die achteraf geplaatst worden of componenten die überhaupt niet geplaatst moeten worden.

|             |      |           |      |                    |  |
|-------------|------|-----------|------|--------------------|--|
| POWER       | P100 | Header 2  | P100 | POWER.SchDoc       | Miscellaneous Connectors.IntLib Not Fitted |
| FPGA_CONFIG | P400 | Header 3  | P400 | FPGA_CONFIG.SchDoc | Miscellaneous Connectors.IntLib Not Fitted |
| FPGA_BANK34 | P700 | Header 10 | P700 | FPGA_BANK34.SchDoc | Miscellaneous Connectors.IntLib Not Fitted |
| POWER       | Q100 | MOSFET-N  | Q100 | POWER.SchDoc       | Miscellaneous Devices.IntLib/M/            |
| POWER       | R100 | 10K       | R100 | POWER.SchDoc       | Miscellaneous Devices.IntLib/Re            |

Figuur 54 Voorbeeld not fitted through hole componenten altium production variant

## 8. EMS Productie

OAC-Electronics beschikt over een volledig automatische Electronics Manufacturing Service (EMS)-productielijn.

### 8.1.1. OAC ERP-codes

Het BOM-project wordt ingeladen in de scanner, deze gaat vervolgens alle ERP codes uit het BOM-project lezen en deze uit de OAC-database halen.

Zodra dit gebeurd is komt er op het scherm van de scanner een lijst met alle componenten die gebruikt zijn binnen het BOM-Project, in deze lijst staat de naam, locatie en aantal van een component. Als je dan een gebruikte component uit het magazijn neemt en deze zijn barcode scant krijg je een groen of rood lampje.

Rood wil zeggen dat je een component hebt genomen die niet gebruikt wordt in je project.

Groen wil zeggen dat je de juiste component hebt genomen, de gescande component wordt vervolgens uit de lijst op de scanner verwijderd en krijgt de status "OUT".



Figuur 55 ERP code scanner OAC electronics

### 8.1.2. Componenten kast

De componentenkast dient om componenten op te slaan in een ruimte met bepaalde luchtvochtigheid en temperatuur. Het is belangrijk dat er geen vocht aanwezig is in een component anders kan dit vocht zich tijdens het reflow proces (oven) omzetten in stoom, als deze stoom niet kan ontsnappen zal er druk opbouwen met als gevolg een defecte component.

Bij het inladen van een BOM-project worden de ERP codes gelezen, vervolgens gaat er een ledje branden bij de gebruikte component in de kast om de locatie aan te geven, plek A39 in de kast wordt gebruikt om componenten aan te geven die in het andere magazijn liggen.



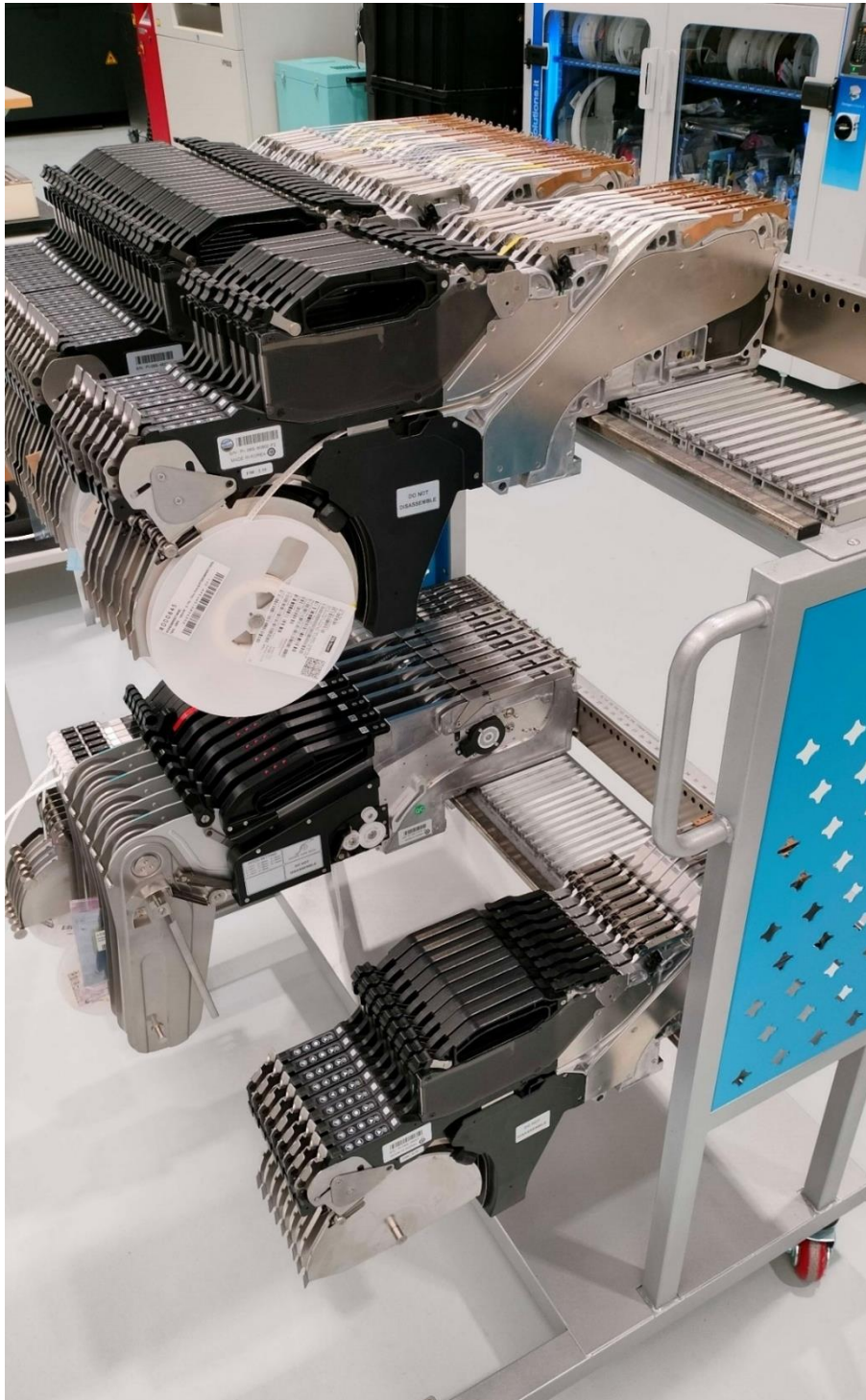
Figuur 56 Componenten kast OAC Electronics

### 8.1.3. Feeders

Feeders dienen om de Pick & Place machines te voorzien van componenten, deze feeders worden met de hand opgemaakt en in de Pick & Place geplaatst op de juiste locatie.

In dit project zijn 44 verschillende feeders gebruikt, de meeste hiervan zijn 8mm feeders, deze worden meestal gebruikt voor 0805 componenten en kleiner. Verder zijn er een paar 12mm, 16mm, 24mm en 32mm die gebruikt worden voor grotere componenten zoals IC's op reel.

Sommige componenten die niet op reel zijn aangekocht worden meestal op een aparte tray geplaatst.



*Figuur 57 Feeders P&P OAC Electronics*

### 8.1.4. Stencil-machine

De stencil machine gaat de soldeerpasta over de stencil smeren voor een optimale laag soldeerpasta.

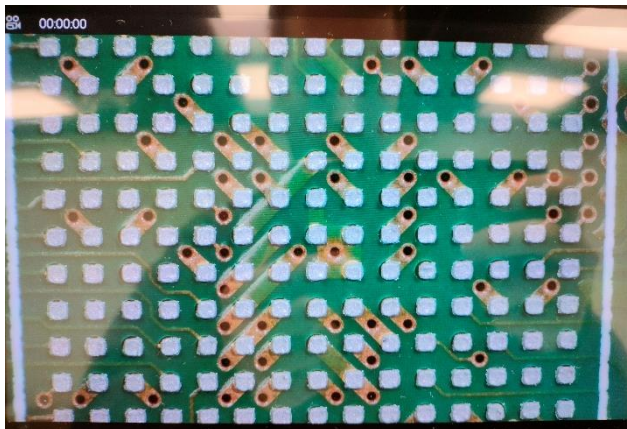
Het gebruikte type soldeerpasta is type 5 soldeerpasta, dit heeft als voordeel dat de pasta bestaat uit fijnere soldeerballetjes wat belangrijk is bij kleinere footprints zoals een BGA of QFN.

Het is belangrijk dat de gebruikte pasta voor gebruik op temperatuur komt, hiervoor wordt een centrifuge gebruikt die de pasta tegen hoge snelheid ronddraait en opwarmt.

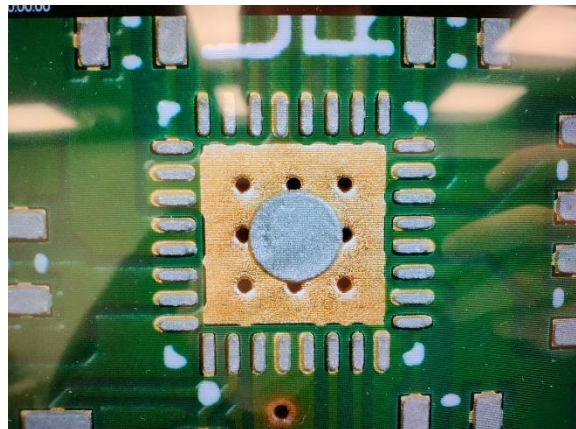
In de machine worden steunpalen (stuts) voorzien om de print te ondersteunen, deze moeten geplaatst worden waar geen componenten staan. Vervolgens wordt de stencil op de print gekalibreerd/gealigneerd en worden parameters zoals druk en snelheid ingesteld.



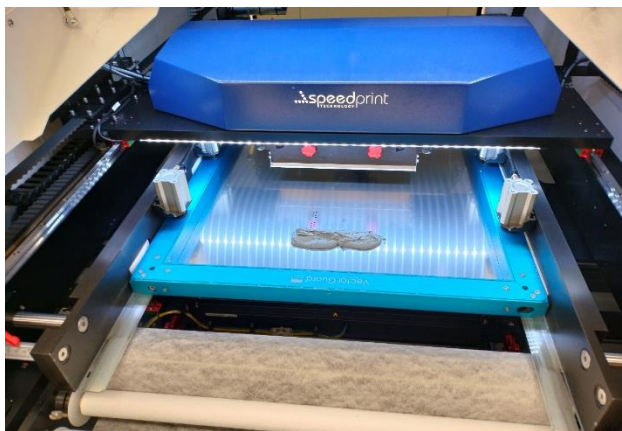
Figuur 58 Stencil Radar board opgespannen kader



Figuur 60 BGA pasta



Figuur 59 QFN pasta



Figuur 61 Stencil machine met pasta



Figuur 62 Stencil machine met PCB

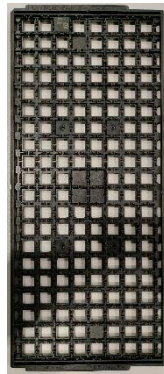
### 8.1.5. Pick & Place

Er worden 2 Pick & Place machines gebruikt in de productielijn, de Hanwha-Decan S2 wordt voornamelijk gebruikt voor het snel plaatsen van veel gebruikte componenten bv. 100nF condensator. De Hanwha-SM482 wordt voornamelijk gebruikt voor het plaatsen van grotere componenten of componenten op tray.

In de Pick & Place machine wordt een .csv file die gegenereerd is door Altium ingeladen, hierin staat welke component op welke x/y coördinaten staat. Verder moet zowel de polarisatie als ligging van elke componenten gecontroleerd worden.

De SM482 is voorzien van een feeder die gevuld is met labels waar een QR-code opstaat die een seriële nummer representeert. Moest er ooit een productie gedraaid worden met bijvoorbeeld een defect kan dit herleid worden via de seriële code.

Componenten die niet beschikbaar zijn op reel (bijvoorbeeld FPGA) worden geplaatst op tray. De trays worden ingeladen in de tray machine deze voorziet de Pick & Place van 22 verschillende beschikbare trays.



Figuur 63 Tray FPGA

Figuur 64 STF-1005 Tray machine



Figuur 65 Smans Hanwha-Decan S2

Figuur 66 Smans Hanwha-SM482

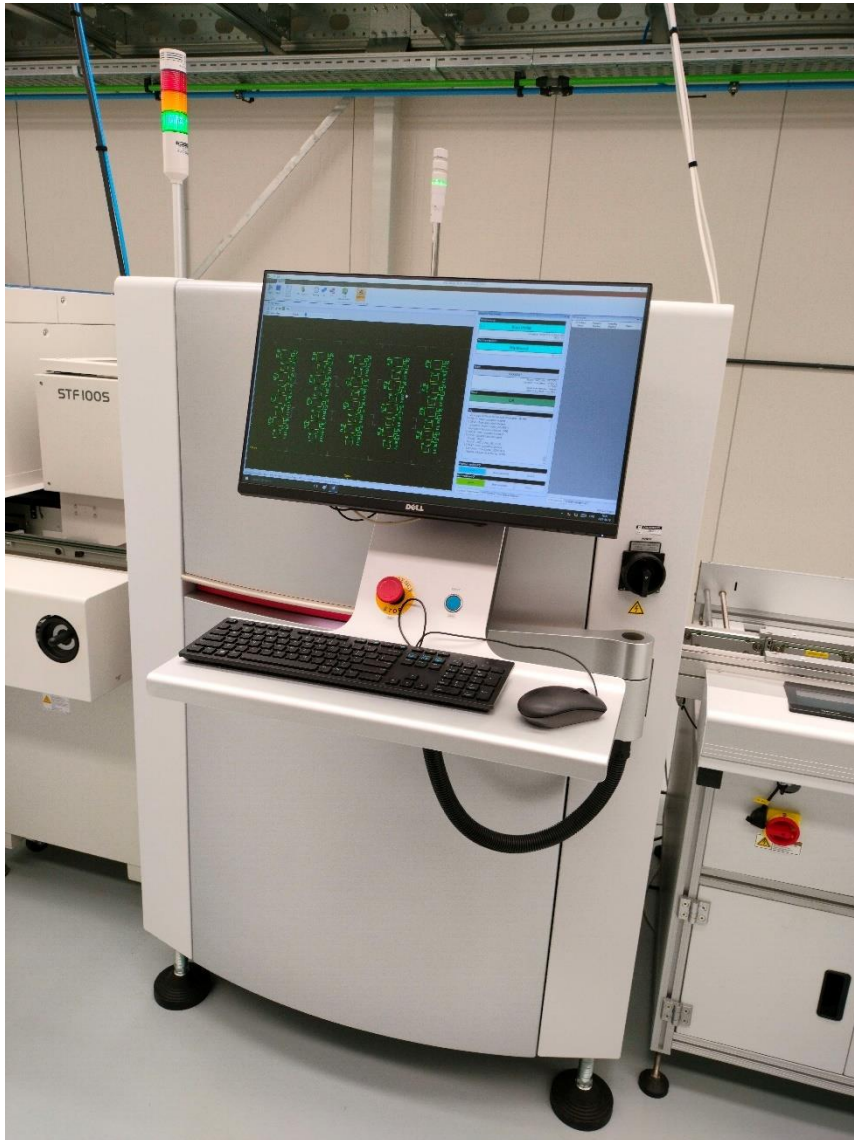
|      |           |          |           |       |        |     |                      |         |
|------|-----------|----------|-----------|-------|--------|-----|----------------------|---------|
| C805 | 100n      | TopLayer | CAP_0603  | 62    | 98     | 360 | Capacitor            | C000137 |
| C804 | 10u       | TopLayer | CAP_0805  | 62    | 100    | 360 | Capacitor            | C000367 |
| U800 | LDL1117S  | TopLayer | FP-SOT-22 | 70.75 | 85.5   | 90  | IC REG LINEAR5V 1.2  |         |
| C801 | 10u       | TopLayer | CAP_0805  | 75    | 88.75  | 180 | Capacitor            | C000367 |
| C800 | 10u       | TopLayer | CAP_0805  | 72    | 79.75  | 360 | Capacitor            | C000367 |
| U801 | ST8R00WF  | TopLayer | DFN(4x4)_ | 76    | 95     | 360 | Micropower step up D |         |
| R802 | 10K       | TopLayer | RES_0603  | 79.5  | 93     | 90  | Resistor             | R000180 |
| R801 | 39K       | TopLayer | RES_0603  | 79.5  | 97     | 90  | Resistor             | R000222 |
| L800 | 4.7u      | TopLayer | IND4x4mm  | 70    | 95     | 90  | Inductor             | L000008 |
| J100 | USB4105-( | TopLayer | FP-USB410 | 44    | 3.7125 | 0   | CONN RCPTUSB2.0 T    |         |

Figuur 67 .csv file in excel



### 8.1.6. Automatic Optical Inspection (AOI)

Automated Optical Inspection (AOI) is een geautomatiseerde visuele inspectie die objecten gaat scannen met camera's om een 3D-model te genereren. Dit 3D-model kan nagekeken worden en vergeleken worden met het 3D-model dat gegenereerd is door de Pick & Place machines. Zo kunnen verkeerd geplaatste componenten gedetecteerd worden, denk bijvoorbeeld aan tombstoning of componenten geplaatst met een verkeerde offset.



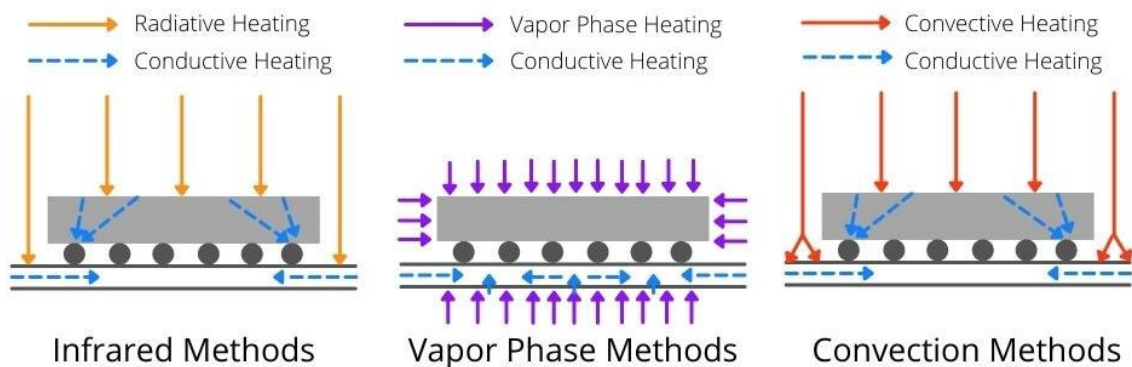
*Figuur 68 AOI productie*

### 8.1.7. Oven & Vapor Phase

De soldeerpaste kan op verschillende manieren gesmolten worden, er is keuze uit een reflow oven of vapor phase.

Voor dit project is gekozen voor de vapor phase deze heeft als voordeel dat het geen warmte straalt zoals een traditionele reflow oven, dit heeft als nadeel dat de warmte niet snel tot bij de middelste array van BGA pins komt waardoor een goede reflow connectie niet verzekerd is.

De vapor phase maakt gebruik van een chemisch materiaal (Galden), als je dit opwarmt ontstaat er een soort damp op heel hoge temperatuur, deze damp kan perfect hoge temperatuur voorzien onder de BGA (FPGA) waardoor een goede reflow connectie verzekerd is.



Figuur 69 [Reflow soldering Vapor phase vs Infrared / Convective heating](#)

De gebruikte FPGA heeft volgend reflow profile waar rekening mee moet gehouden worden.

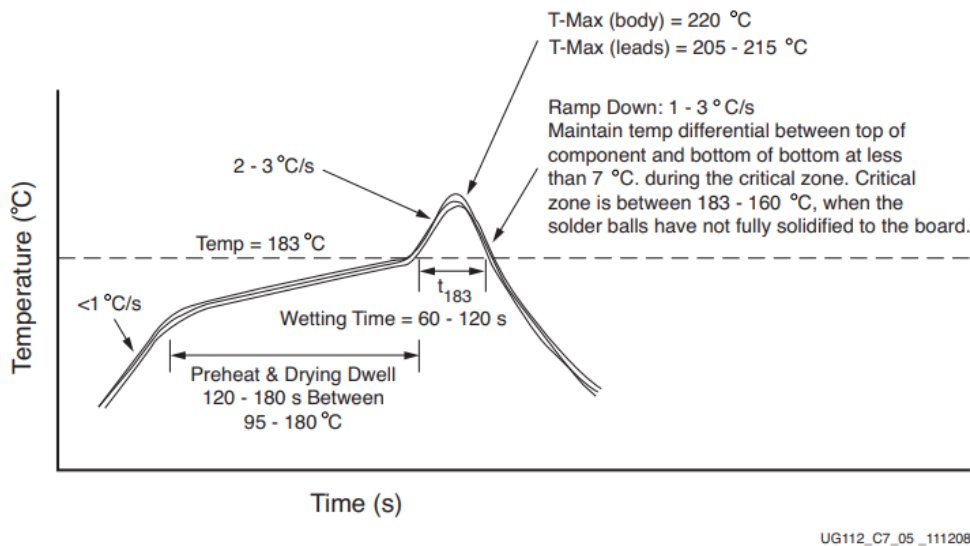
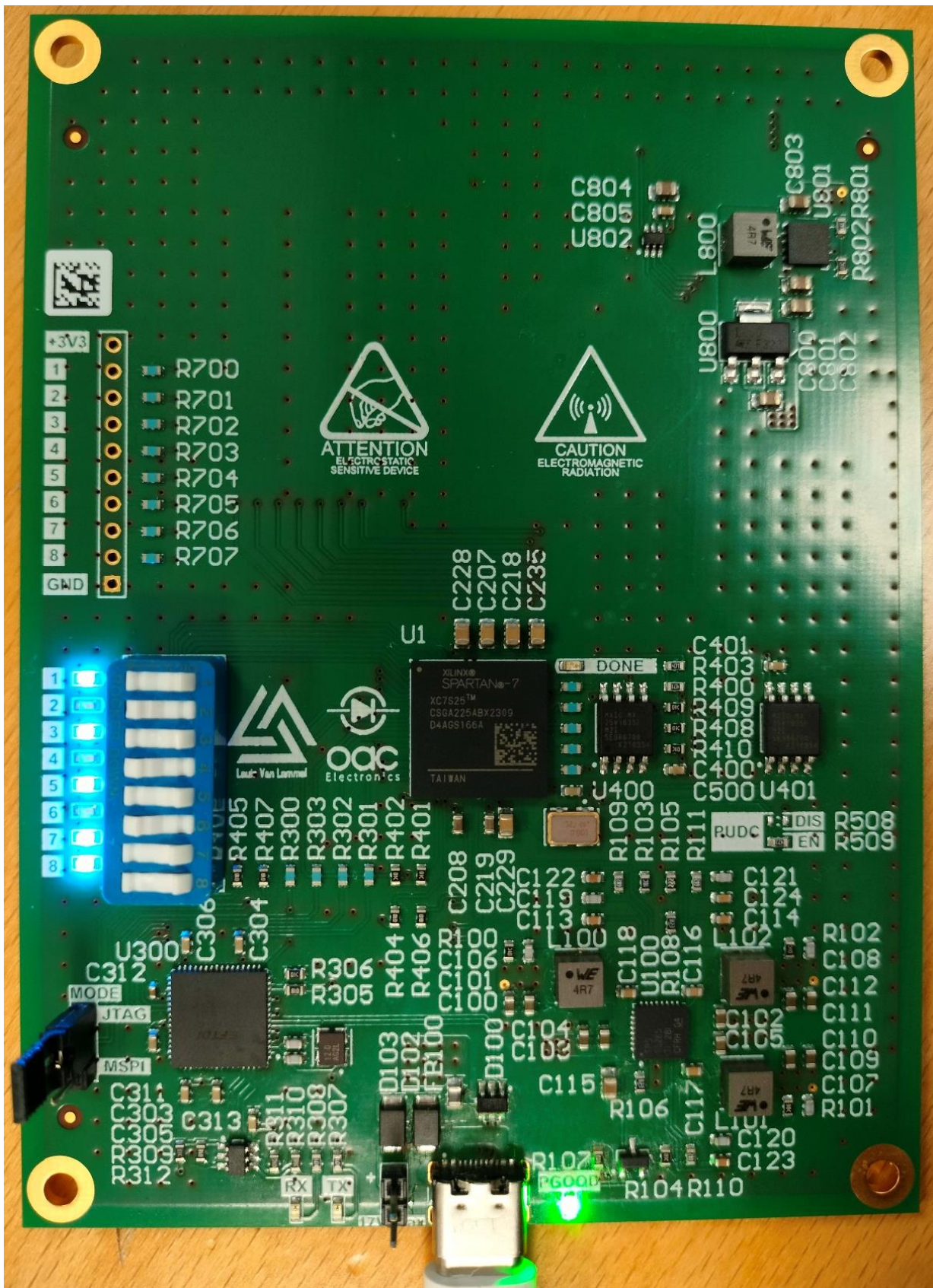


Figure 7-5: **Typical Solder Reflow Profile for BGA**

Figuur 70 Typical Solder Reflow Profile for BGA (CSGA225) (UG112/XAPP427)

### 8.1.8. Eindproduct



Figuur 71 PCB na volledige productie

### 8.1.9. Testing

Op aanvraag en met behulp van stagebegeleider Pedro Wyns is een near field meting gedaan van het project in operating condities met een uitzend vermogen van 14dBm.

Hierbij werd vastgesteld dat de maximumstraling van 3V/m in de near field niet werd overschreven. Extra maatregelen en signalisatie om hiertegen te waarschuwen zijn dus niet nodig.

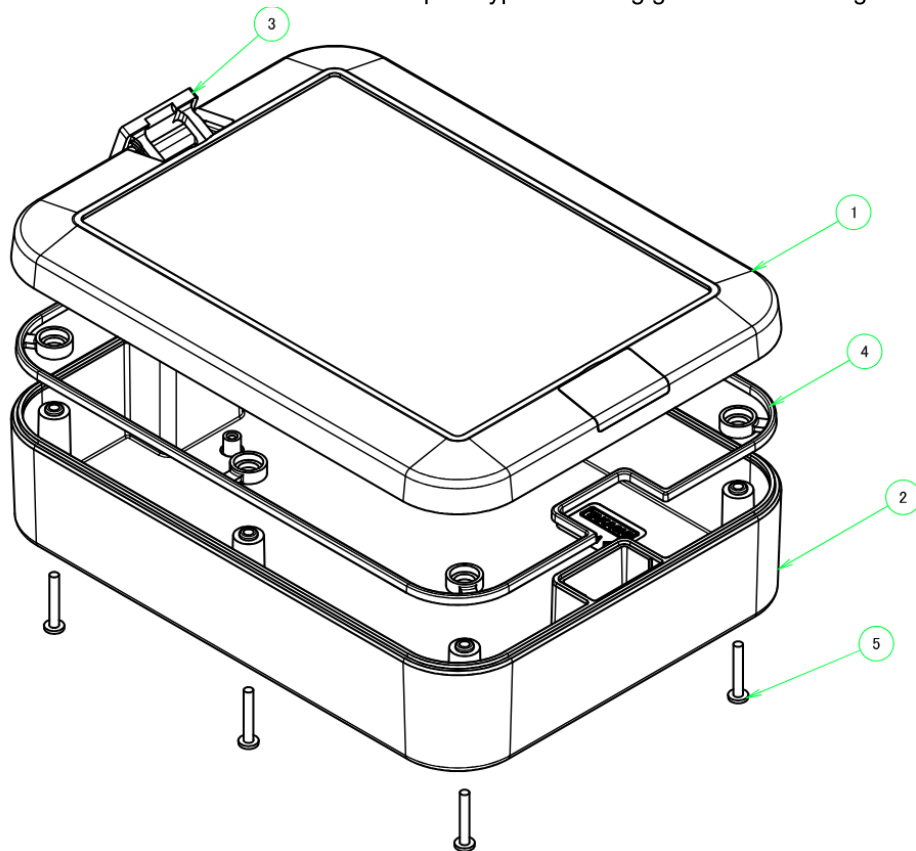
Voor deze meting werd de Narda NBM-550 broadband field meter gebruikt.



*Figuur 72 Near field meting IVQ-3005 op 14dBm, resultaat op ~60cm = 0.4V/m*

## 8.2. Behuizing

De behuizing moet voldoen aan enkele vereisten om een optimale werking van de radar module te kunnen garanderen. De Takachi WP13-18-5C is als eerste prototype behuizing gekozen met rating IP68.



Figuur 73 Takachi WP13-18-5C

### 8.2.1. Materiaal

Het is belangrijk om een behuizing te kiezen die geen elektromagnetische reflecties kan veroorzaken oftewel materiaal dat geen metaal of carbon deeltjes bevat. Enkele materialen die geschikt zijn hiervoor zijn: polycarbonaat of ABS, best met een dikte van 3-4mm. Stickers of verf die later over de case komen kunnen ook storing veroorzaken.

### 8.2.2. Mechanisch

Mechanisch is het belangrijk dat de behuizing en PCB goed aan elkaar vast hangen aangezien vibraties storing kunnen veroorzaken.

Bijvoorbeeld, er komt een vrachtwagen voorbijgereden die trillingen veroorzaakt, dan is het de bedoeling dat de behuizing op exact dezelfde manier trilt als de PCB waardoor er voor de radar module in zijn nabije omgeving niets verandert.

Verder kan er best een afstand van minimaal 1cm gerespecteerd worden tussen behuizing en radarantenne.

### 8.2.3. Storing

Fluorescerend licht kan storing op 50Hz en 100Hz veroorzaken.

# 9. Software

## 9.1. Xilinx

De voornaamste reden waarom er is gekozen voor een Xilinx FPGA is vanwege de sublieme documentatie en software ondersteuning. Het is ongelofelijk hoe goed al hun producten gedocumenteerd zijn, van software tot hardware, dit maakt het begrijpen veel eenvoudiger en versnelt het proces om aan de slag te kunnen met hun producten. Aan dit alles hangt natuurlijk ook een prijskaartje.

De 2 Softwarepakketten die gebruikt zijn binnen dit project zijn Xilinx Vivado en Xilinx Vitis.

## 9.2. Vivado

Vivado wordt gebruikt om FPGA's hun interne transistor to transistor connecties te gaan definiëren, je bent dus als het ware hardware aan het programmeren.

In dit project wordt Vivado gebruikt om de hardware te beschrijven die nodig is voor onze toepassing, hiervoor wordt de IP-catalog gebruikt.

Verder is er ook testcode geschreven in VHDL om tijdens het productieproces de werking van de PCB te testen. Er kunnen ook custom IP-blocks geschreven worden in VHDL deze kunnen dan geïmplementeerd worden in uw IP-catalog-design.

### 9.2.1. XDC-file

De XDC-file bepaald welke pinnen welke naam en operating conditions krijgen. Zo heeft dit project zijn eigen XDC-file, deze is terug te vinden onder VIVADO => Spartan7S25-1CSGA225\_RADAR.xdc.

### 9.2.2. FTDI to JTAG programmer

In Vivado is een tcl-console aanwezig, deze kunnen we gebruiken om de EEPROM van de FTDI-chip te gaan programmeren met licensed Xilinx Firmware (Let op Vivado V2020.2 of later vereist).

Volgend commando wordt gebruikt:

```
program_ftdi -write -ftdi FT2232H -serial OABC01 -vendor "my vendor co" -board "my board" -desc "my product desc"
```

```
program_ftdi -write -ftdi FT2232H -serial OAC123 -vendor "OAC Electronics" -board "RADAR" -desc "Traffic counter"
WARNING: [Common 17-259] Unknown Tcl command 'program_ftdi -write -ftdi FT2232H -serial OAC123 -vendor OAC Electrc

***** program_ftdi v2023.1
**** Build date : May 7 2023-15:26:58
** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.
** Copyright 2022-2023 Advanced Micro Devices, Inc. All Rights Reserved.

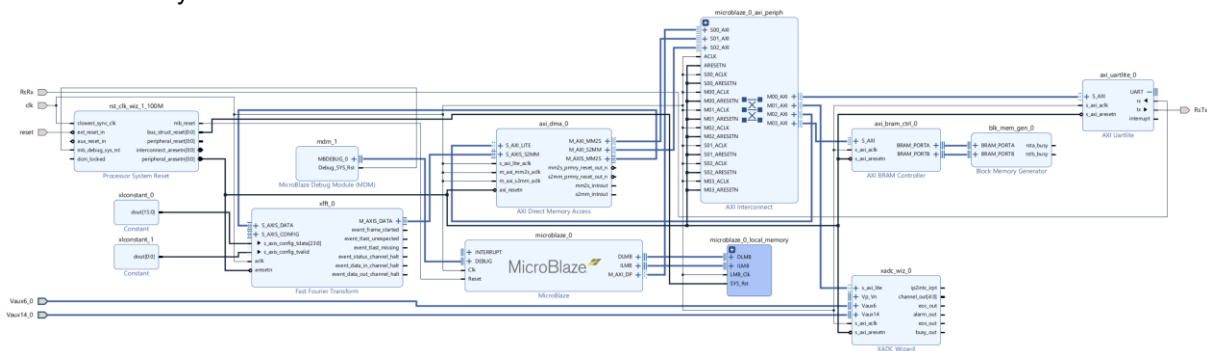
INFO: ftdi part = FT2232H
INFO: Serial = OAC123
INFO: Detected 1 devices
INFO: Device location = 801
INFO: fwid=0x584a0003
INFO: FTDI Programming Passed
```

Figuur 74 Program FTDI commando TCL console

### 9.2.3. IP-catalog

In de IP-catalog kunnen we de bouwstenen gaan definiëren voor de inwendige hardware, hieronder vallen:

- Softcore: Microblaze
- ADC: XADC
- FFT: XFFT
- DMA: Axi DMA
- UART: UARTvLite
- GPIO: XGPIO
- Memory: FLASH/BRAM



Figuur 75 Vivado IP-catalog

Alle verbindingen tussen de bouwstenen gebeuren doormiddel van de AXI-bus (**A**dvanced **e**Xtensible **I**nterface), dit is een parallelle high-performance, synchrone, high-frequency, multi-master, multi-slave, communicatie interface protocol op basis van de ARM AMBA (**A**dvanced **M**icrocontroller **B**us **A**rchitecture) standard voor on-chip communicatie.

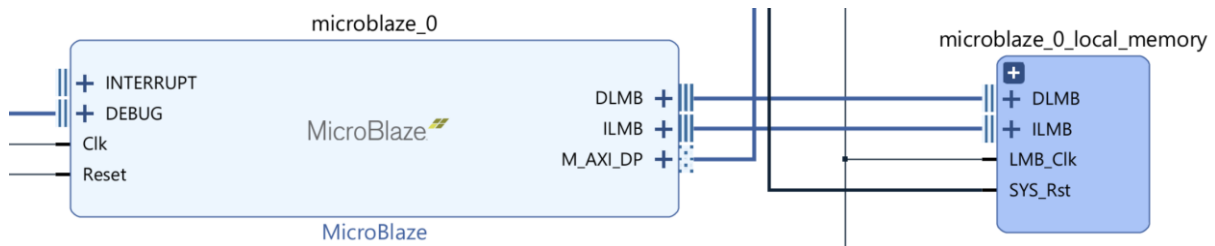
Naast functionele bouwstenen zijn er ook debugging bouwstenen zo kan een Integrated Logic Analyzer (ILA) toegevoegd worden, dit biedt de mogelijkheid om live debugging toe te passen in vivado. Een testbench maken is altijd een goede manier om de werking van een project op voorhand te verifiëren.

In de IP-catalog kunnen we dus bouwstenen toevoegen, configureren en verbinden om zo de gewenste hardware te bekomen. Zodra deze configuratie klaar is kan deze geëxporteerd worden, deze export zal nodig zijn om te includen bij de programmatie in C. (.xsa file in Vitis)

## 9.2.4. Softcore (Microblaze)

Een softcore (Microblaze in dit geval) is eigenlijk een volledige microcontroller die beschreven gaat worden in de FPGA, er zijn verschillende settings zoals Microcontroller, RTOS, Linux... voor deze toepassing is een 32bit microcontroller gekozen.

Dit geeft de mogelijk om te werken met een traditionele microcontroller en toch de parallel processing te benutten waar nodig. Zo maken alle bouwstenen rond de softcore gebruik van de parallel processing.



Figuur 76 Microblaze met local memory

Bij een standaard microcontroller zijn alle inwendige adressen bepaald, bij een softcore krijg je de mogelijk om dit zelf te regelen. Dit wordt bepaald in de Address Editor.

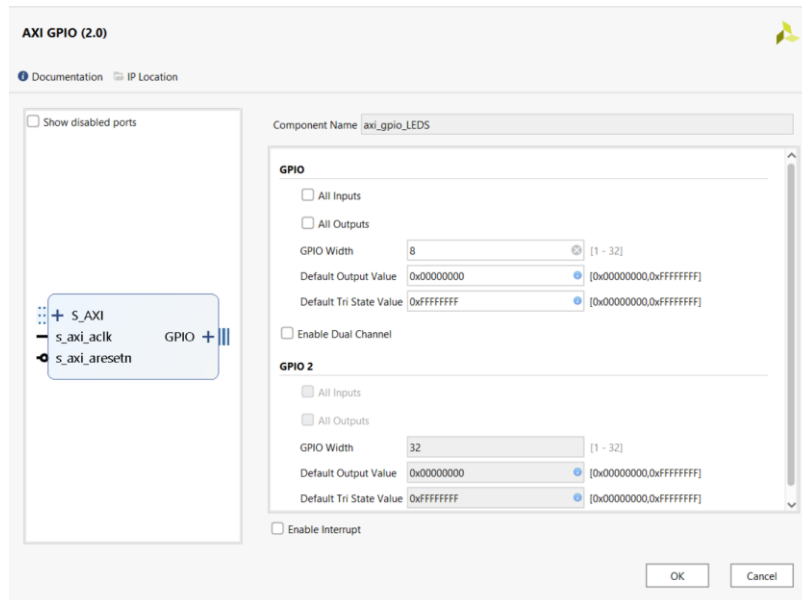
| Name  | Interface  | Slave Segment | Master Base Address | Range | Master High Address |
|---|------------|---------------|---------------------|-------|---------------------|
| Network 0   |            |               |                     |       |                     |
| /axi_dma_0  |            |               |                     |       |                     |
| /axi_dma_0/Data_MM2S (64 address bits : 16E)        |            |               |                     |       |                     |
| /axi_bram_ctrl_0/S_AXI                              | S_AXI      | Mem0          | 0xC000_0000         | 16K   | 0xC000_3FFF         |
| /axi_dma_0/S_AXI_LITE                               | S_AXI_LITE | Reg           | 0x41E0_0000         | 64K   | 0x41E0_FFFF         |
| /xadc_wiz_0/s_axi_lite                              | s_axi_lite | Reg           | 0x44A0_0000         | 64K   | 0x44A0_FFFF         |
| Excluded (1)  |            |               |                     |       |                     |
| /axi_uartlite_0/S_AXI                               | S_AXI      | Reg           | 0x4060_0000         | 64K   | 0x4060_FFFF         |
| /axi_dma_0/Data_S2MM (64 address bits : 16E)        |            |               |                     |       |                     |
| /axi_bram_ctrl_0/S_AXI                              | S_AXI      | Mem0          | 0xC000_0000         | 16K   | 0xC000_3FFF         |
| /axi_dma_0/S_AXI_LITE                               | S_AXI_LITE | Reg           | 0x41E0_0000         | 64K   | 0x41E0_FFFF         |
| /xadc_wiz_0/s_axi_lite                              | s_axi_lite | Reg           | 0x44A0_0000         | 64K   | 0x44A0_FFFF         |
| Excluded (1)  |            |               |                     |       |                     |
| /axi_uartlite_0/S_AXI                               | S_AXI      | Reg           | 0x4060_0000         | 64K   | 0x4060_FFFF         |
| /microblaze_0                                       |            |               |                     |       |                     |
| /microblaze_0/Data (32 address bits : 4G)           |            |               |                     |       |                     |
| /axi_bram_ctrl_0/S_AXI                              | S_AXI      | Mem0          | 0xC000_0000         | 16K   | 0xC000_3FFF         |
| /axi_dma_0/S_AXI_LITE                               | S_AXI_LITE | Reg           | 0x41E0_0000         | 64K   | 0x41E0_FFFF         |
| /axi_uartlite_0/S_AXI                               | S_AXI      | Reg           | 0x4060_0000         | 64K   | 0x4060_FFFF         |
| /microblaze_0_local_memory/lmb_bram_if_cntlr_0/SLMB | SLMB       | Mem           | 0x0                 | 64K   | 0xFFFF              |
| /xadc_wiz_0/s_axi_lite                              | s_axi_lite | Reg           | 0x44A0_0000         | 64K   | 0x44A0_FFFF         |
| Network 1   |            |               |                     |       |                     |
| /microblaze_0                                       |            |               |                     |       |                     |
| /microblaze_0/Instruction (32 address bits : 4G)    |            |               |                     |       |                     |
| /microblaze_0_local_memory/lmb_bram_if_cntlr_1/SLMB | SLMB       | Mem           | 0x0                 | 8K    | 0x1FFF              |

Figuur 77 Address Editor Vivado Microblaze



## 9.2.5. GPIO

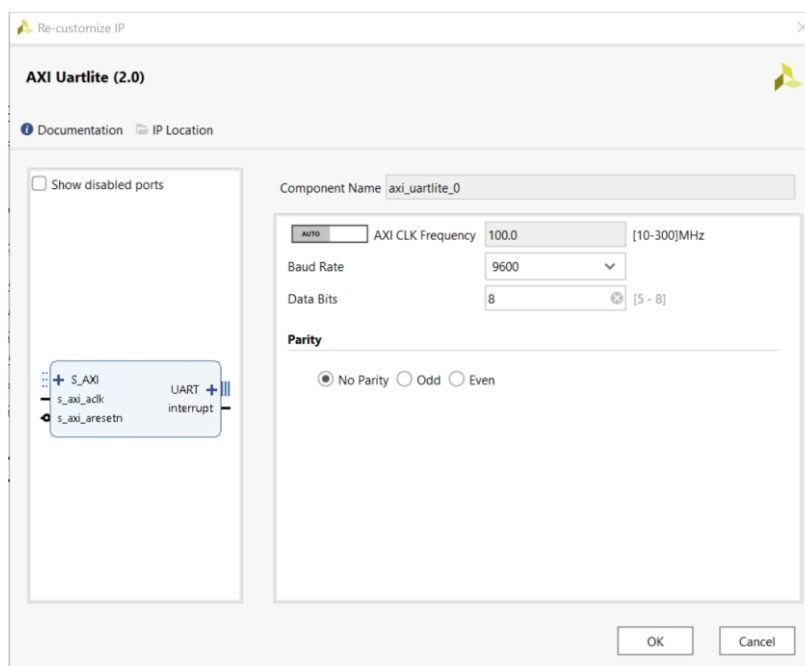
General Purpose Inputs/Outputs (GPIO) kunnen heel eenvoudig als bouwsteen worden toegevoegd en geconfigureerd, doormiddel van auto generate connections worden deze blokken op de Axi bus aangesloten en hoef je verder niets te doen.



Figuur 78 AXI GPIO configuration

## 9.2.6. UART

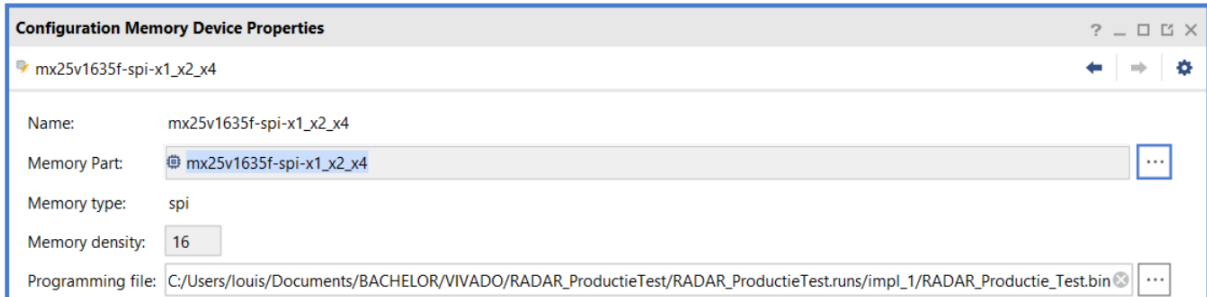
UART-communicatie kan heel simpel worden toegevoegd, je voegt de bouwsteen "AXI Uartlite" toe stelt de Baud rate correct in en bepaald welke pin de Tx pin is en welke de Rx pin is. Vervolgens klik je generate connection en de uart block wordt aangesloten op de AXI bus.



Figuur 79 UARTlite configuration

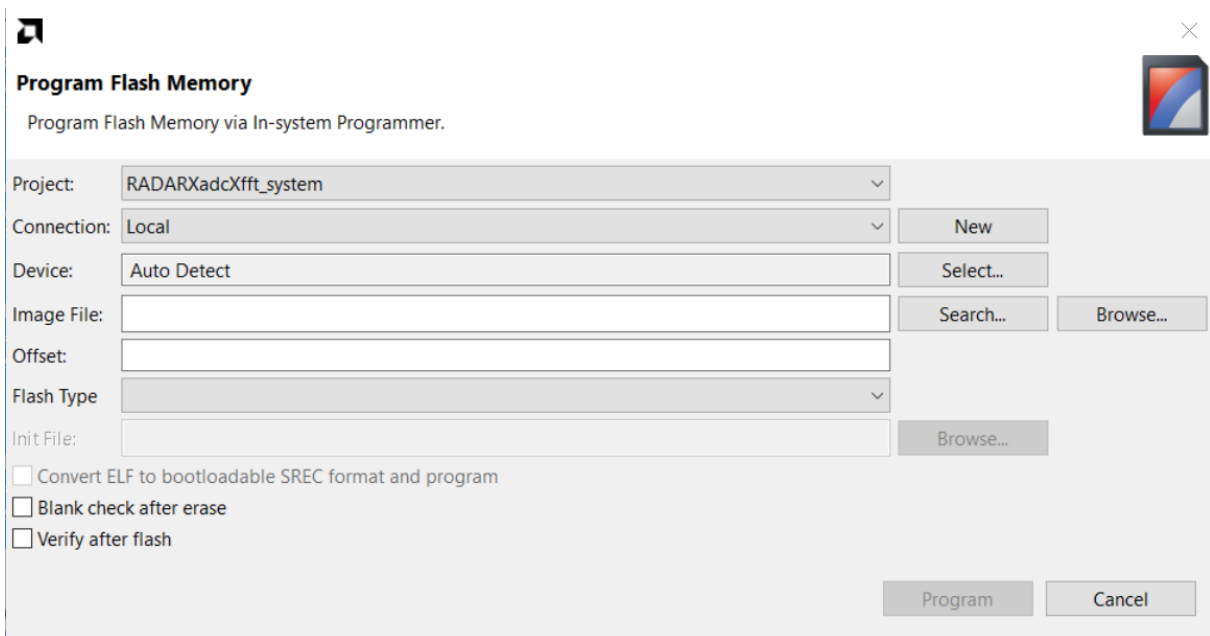
## 9.2.7. FLASH

In de Vivado hardware manager kan een configuration memory device toegevoegd worden. Voeg hierbij de gebruikte Macronix mx25v1635f-spi-x1\_x2\_x4 toe en zet in de settings in vivado aan om bij de bitstream ook een .bin file te genereren. Vervolgens geef je de .bin file mee om in de flash te stoppen, deze zal bij boot automatisch ingeladen worden.



Figuur 80 Programming FLASH

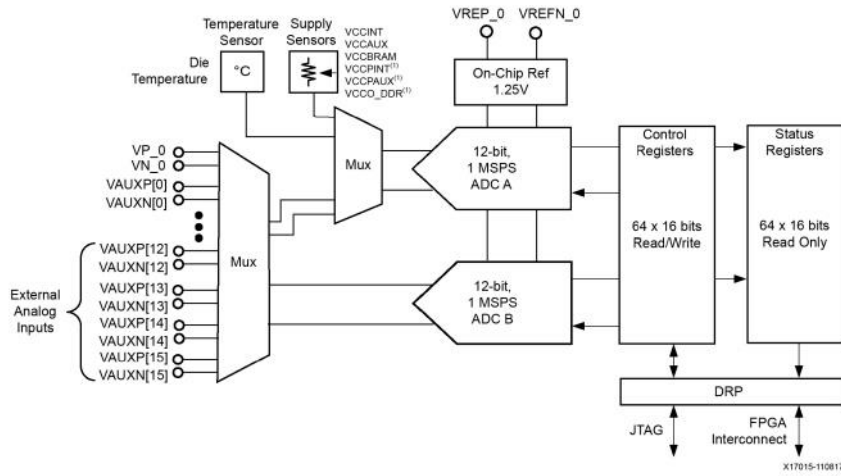
In Vitis 2032.1 is een bug aanwezig waardoor het programmeren van een FLASH memory device niet mogelijk is, gebruik hiervoor een andere versie.



Figuur 81 Program FLASH in Vitis

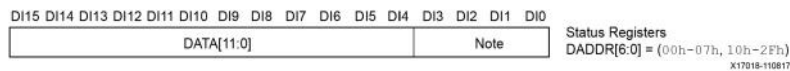
## 9.2.8. AnalooG Digitaal Converter (XADC)

De interne ADC op de FPGA is een 12bit, 1M sample per second Pipelined ADC met on-chip sensoren.



Figuur 82 Block diagram XADC (UG480)

De uitgang van de ADC wordt opgeslagen in een 16-bit register waarvan de 12 Most Significant Bits (MSB) de gemeten data zijn. De 4 Least Significant Bits (LSB) kunnen gebruikt worden om quantizatie te minimaliseren of om de resolutie te verbeteren op basis van averaging/filtering.



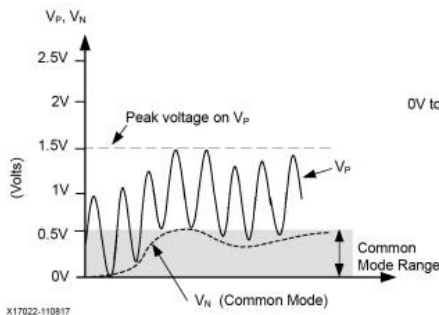
Figuur 83 16-bit Status Register XADC (UG480)

Er zijn 2 verschillende operating modes in de ADC, Unipolar & Bipolar mode.

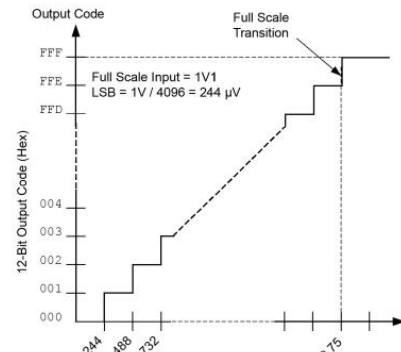
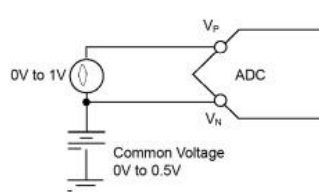
### Unipolar mode:

De nominale analoge input van de ADC ligt tussen 0V en +1V (1Vpp). De ADC geeft een output van 0 (000h) bij een spanning van 0V en een max output waarde van FFFh bij een spanning van +1V.

Unipolar mode biedt de mogelijkheid om een spanning tot +1.5V te meten door gebruik te maken van een common mode spanning van +0.5V waardoor de +1Vpp gerespecteerd blijft.



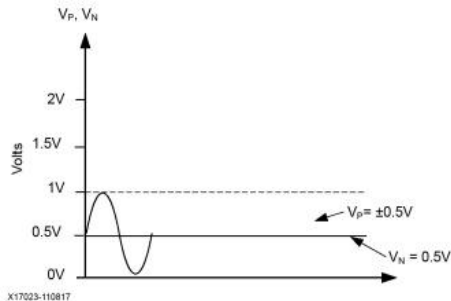
Figuur 85 XADC Unipolar Input Signal Range (UG480)



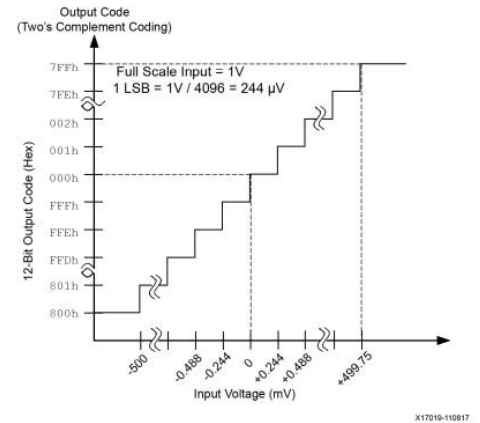
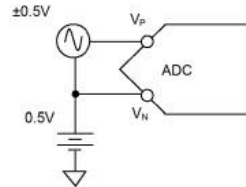
Figuur 84 XADC Unipolar Transfer Function (UG480)

### Bipolar Mode:

Net zoals bij de unipolar mode ligt de nominale analoge input van de ADC tussen 0V en +1V (1Vpp). In tegenstelling tot de unipolar mode geeft de ADC een output van 0 (000h) bij een spanning van 0.5V, meestal de common mode spanning. Alles boven de common mode spanning wordt bepaald tussen de 000h en 7FFh (de helft van 12bit), alles onder de common mode spanning wordt bepaald tussen de FFFh en 800h (de helft van 12bit).



Figuur 87 XADC Bipolar Input Signal Range (UG480)



Figuur 86 XADC Bipolar Transfer Function (UG480)

### Resolutie:

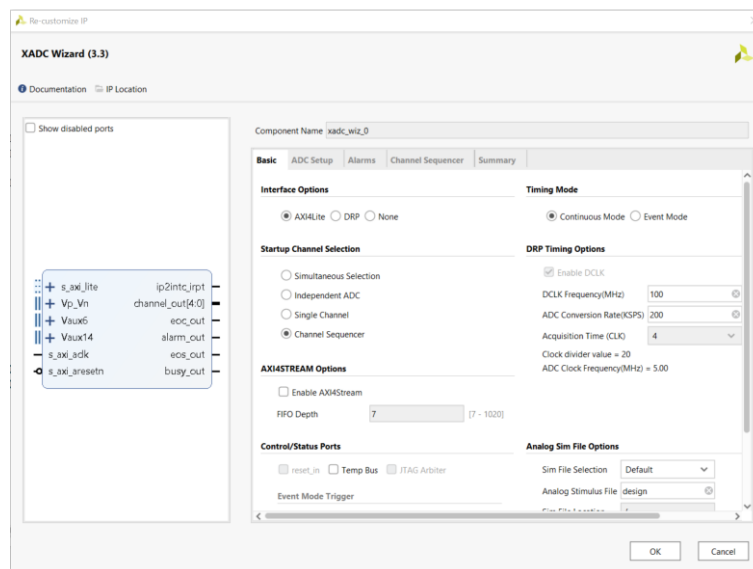
Wetende dat het signaal +1Vpp is kunnen we de resolutie bepalen:

$$\frac{V_{inmax} - V_{inmin}}{Resolution} = \frac{1V - 0V}{12bit(= 4096 = 2^{12})} = 244\mu V$$

### IP-catalog settings:

Als we de ADC willen gebruiken in een project is het nodig om de XADC wizard block toe te voegen aan het project.

De XADC wizard block heeft verschillende instellingen die belangrijk zijn binnen dit project. Alle analoge ingangen kunnen gemonitord worden via de Xilinx dashboard monitor.



Figuur 88 XADC Basic Configuration

De belangrijkste basic settings zijn Timing Mode, Startup Channel Selection en DRP Timing Options.

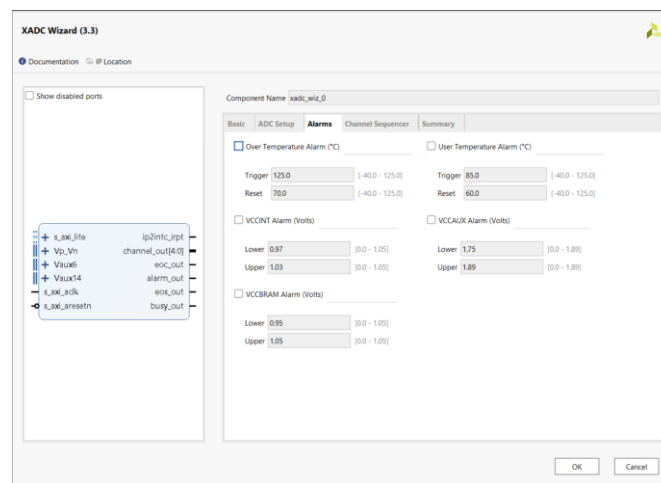
De Timing Mode geeft de mogelijkheid om op timer basis te gaan samplen, dit is nodig bij tijdgevoelige metingen.

De Startup Channel Selection bepaald de operating mode van de XADC, hier is gekozen voor de Channel Sequencer, dit geeft de mogelijk om alle ADC uitgangen tegelijk in te lezen.

In de DRP Timing options kunnen we de Sampling rate gaan bepalen, deze is op minimaal 120K samples per second gekozen. Deze keuze is gemaakt met respect voor de "Shannon-Nyquist theorem" die bepaald dat de sampling rate minimaal het dubbele moet zijn van de maximaal te meten frequentie die in dit geval 60kHz bedraagt.

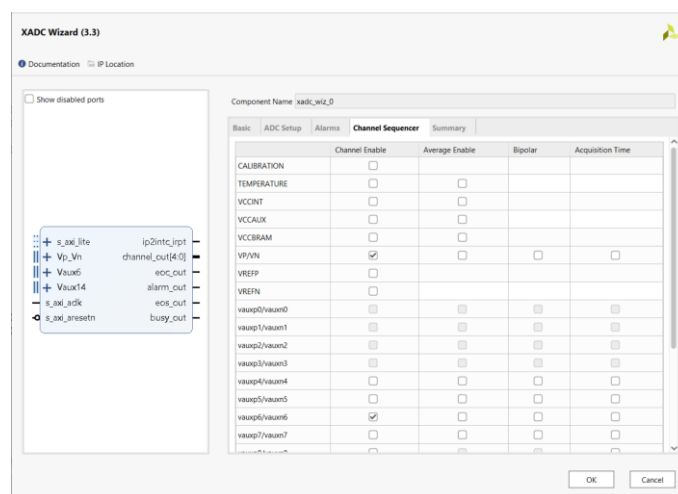
$$\frac{f_s}{f_{max}} \geq 2 = \frac{120k}{60k} = 2$$

Er is ook de mogelijkheid om alarmen in te stellen voor Temperatuur en overspanning.



Figuur 89 XADC ALARMS Configuration

In de Channel Sequencer Configuration kan er bepaald worden welke analoge inputs er enabled worden en of deze in unipolar (standaard) of bipolar mode werken.



Figuur 90 XADC Channel Sequencer Configuration

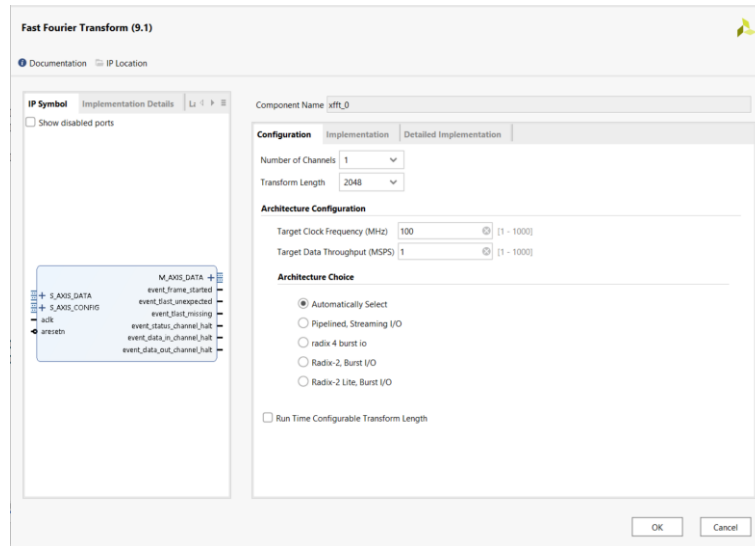
### 9.2.9. Fast Fourier Transform (XFFT)

De FFT is zoals eerder vermeld een algoritme om een tijdsdomein signaal om te zetten naar een frequentiedomein signaal. Zo kan een frequentie van een analogo signaal softwarematig bepaald worden.

In de Vivado IP-catalog is een FFT-bouwsteen voorzien, deze zal automatisch het algoritme uitvoeren op de toegekende input. De FFT-bouwsteen heeft volgende settings:

**Number of Channels:** Aantal input channels (1 in dit geval)

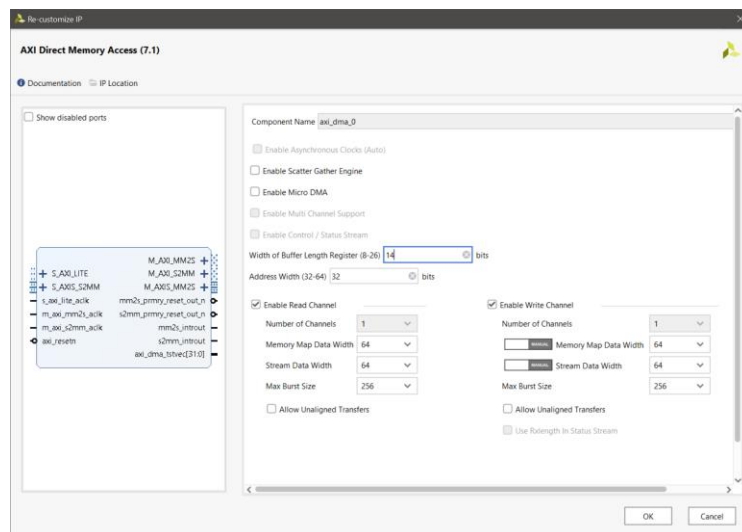
**Transform Length (Block Length):** Block length is zoals eerder besproken het aantal input samples dat verwacht is.



Figuur 91 XFFT configuration

### 9.2.10. Direct Memory Access (DMA)

De DMA heeft als taak om grote hoeveelheden data snel te verplaatsen zodat de microcontroller zo min mogelijk belast wordt. Deze gaat dus de samples die opgeslagen zijn in het intern geheugen verplaatsen naar de FFT input, de FFT output zal verplaatst worden naar het interne geheugen.



Figuur 92 XDMA configuration

### 9.3. Vitis

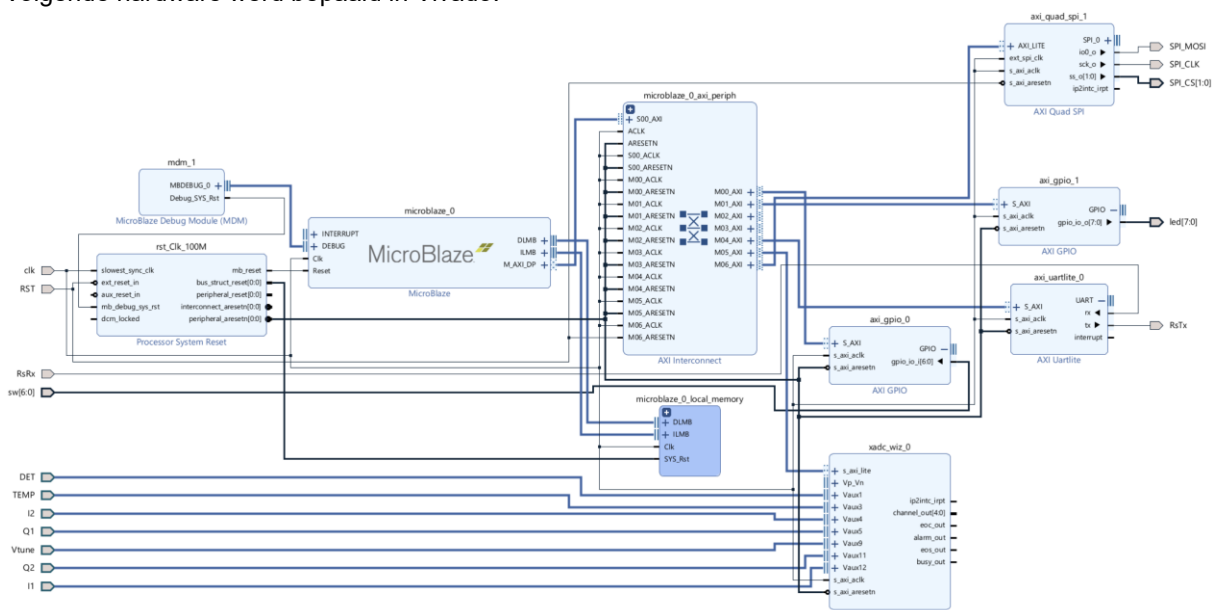
In Vitis wordt de hardware die beschreven is in Vivado geprogrammeerd in C code. Dit heeft als voordeel dat het veel eenvoudiger en overzichtelijker is als VHDL-code.

Om aan een Vitis project te kunnen beginnen is eerst de geëxporteerde hardware uit vivado nodig (.xsa) dit moet geïncludeerd worden zodat Vitis zijn “\_wrapper” files kan genereren dit houdt in: Include files, linker script...

Er is ook een mogelijkheid om de geschreven test code te exporteren als .elf file, van deze file kan een testbench gegenereerd worden, zeer handig tijdens het debuggen.

#### 9.3.1. ADC testcode

Deze testcode test de basisfunctionaliteit van de ADC test configuratie. Volgende hardware werd bepaald in Vivado:



Figuur 93 Vivado IP block, ADC test

De testcode is terug te vinden onder 13.4 bijlagen.

Bij het testen van de code in samenwerking met de FFT is gebleken dat de samplerate op deze manier onbepaald en niet constant was, uiteindelijk is deze code dus herschreven op timer/interrupt basis.





De Xilinx Spartan-7 Series XC7S25 is gekozen aangezien dit het eerste model is dat een interne ADC ondersteunt en zowel beschikt over voldoende resources. Als package is de CSGA225 gekozen aangezien de eenvoudigere package (FTGB196) niet beschikt over een interne ADC.

## Spartan-7 FPGAs

|                            |                        |                                    | I/O Optimization at the Lowest Cost and Highest Performance (1.0V, 0.95V) |                                      |           |           |
|----------------------------|------------------------|------------------------------------|---|--------------------------------------|-----------|-----------|
|                            |                        | Part Number                        | XC7S6   | XC7S15                               | XC7S25    | XC7S50    |
| Logic Resources            |                        | Logic Cells                        | 6,000   | 12,800                               | 23,360    | 52,160    |
|                            |                        | Slices                             | 938   | 2,000                                | 3,650     | 8,150     |
|                            |                        | CLB Flip-Flops                     | 7,500   | 16,000                               | 29,200    | 65,200    |
| Memory Resources           |                        | Max. Distributed RAM (Kb)          | 70  | 150                                  | 313       | 600       |
|                            |                        | Block RAM/FIFO w/ ECC (36 Kb each) | 5   | 10                                   | 45        | 75        |
|                            |                        | Total Block RAM (Kb)               | 180   | 360                                  | 1,620     | 2,700     |
| Clock Resources            |                        | Clock Mgmt Tiles (1 MMCM + 1 PLL)  | 2   | 2                                    | 3         | 5         |
| I/O Resources              |                        | Max. Single-Ended I/O Pins         | 100   | 100                                  | 150       | 250       |
|                            |                        | Max. Differential I/O Pairs        | 48  | 48                                   | 72        | 120       |
| Embedded Hard IP Resources |                        | DSP Slices                         | 10  | 20                                   | 80        | 120       |
|                            |                        | Analog Mixed Signal (AMS) / XADC   | 0   | 0                                    | 1         | 1         |
|                            |                        | Configuration AES / HMAC Blocks    | 0   | 0                                    | 1         | 1         |
| Speed Grades               |                        | Commercial Temp (C)                | -1,-2   | -1,-2                                | -1,-2     | -1,-2     |
|                            |                        | Industrial Temp (I)                | -1,-2,-1L   | -1,-2,-1L                            | -1,-2,-1L | -1,-2,-1L |
|                            |                        | Expanded Temp (Q)                  | -1  | -1                                   | -1        | -1        |
|                            | Package <sup>(1)</sup> | Body Area (mm)                     | Ball Pitch (mm)   | Available User I/O: 3.3V SelectIO™ H |           |           |
|                            | CPGA196                | 8x8                                | 0.5   | 100                                  | 100       |           |
|                            | CSGA225                | 13x13                              | 0.8   | 100                                  | 100       | 150       |
|                            | CSGA324                | 15x15                              | 0.8   |                                      |           | 150       |
|                            | FTGB196                | 15x15                              | 1.0   | 100                                  | 100       | 100       |
|                            | FGGA484                | 23x23                              | 1.0   |                                      |           | 250       |
|                            | FGGA676                | 27x27                              | 1.0   |                                      |           |           |

Figuur 96 Product Selection Guide Xilinx 7 Series, XC7S25

### 9.3.3. CW-RADAR software

Volgende code werd toegepast om via de continuous wave modulatietechniek snelheid en rijrichting te meten. Dit toont de basis-functionaliteit van dit project aan. De code is terug te vinden onder 13.4 Bijlagen Vitis.

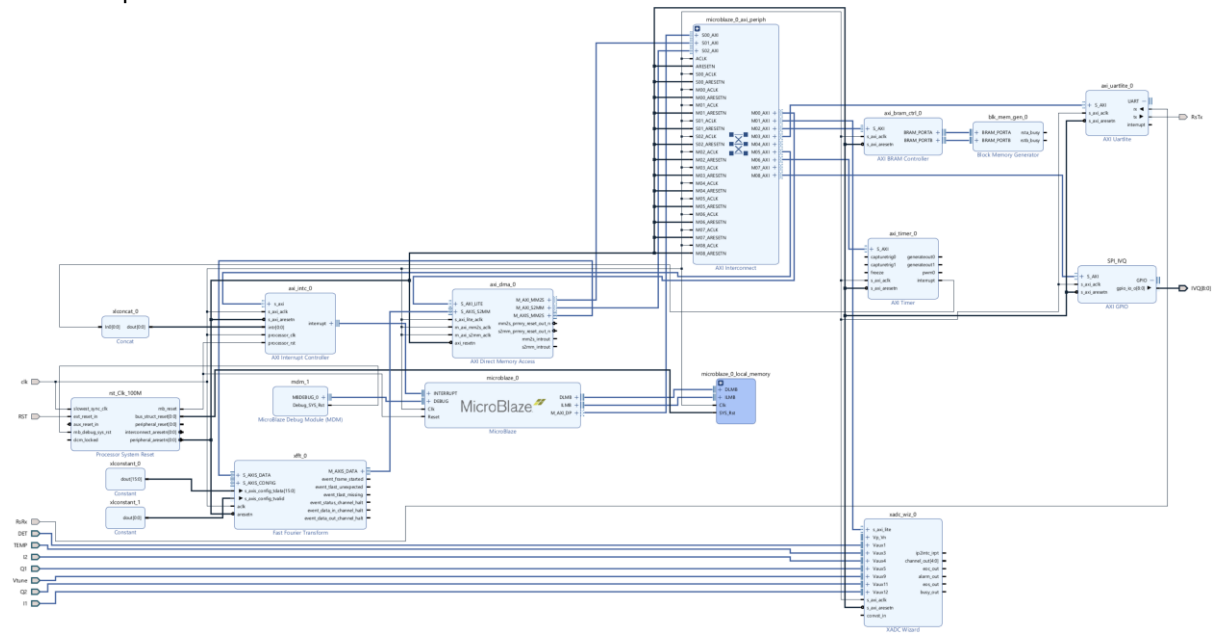
De code heeft volgende parameters: 128bin FFT, 20ksamples/sec samplerate

De resolutie kan als volgt bepaald worden  $20000/128 = 156\text{Hz}$  resolutie

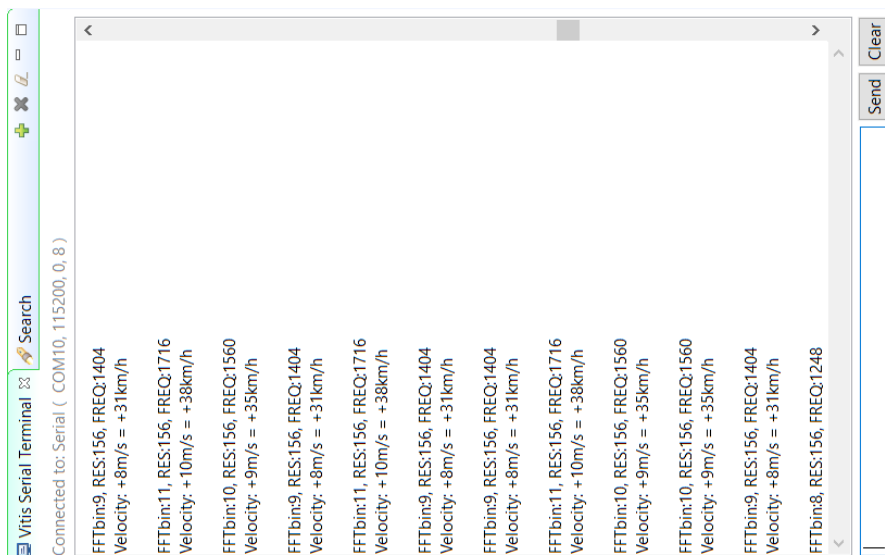
De snelheid wordt als volgt berekend:  $v = \lambda * fd/2$ , vanwege de frequentie resolutie (156Hz) is er een snelheidsresolutie van 3km/h.

De resolutie kan verbeterd worden door de FFT-length te verhogen of de sampling rate te verlagen.

De FFT-length verhogen vraagt meer processing power, de sampling rate verlagen beperkt de maximum te meten frequentie.



Figuur 97 VIVADO defined hardware CW-RADAR IP-CATALOG



Figuur 98 Serial Monitor Vitis: Output van een wagen die ongeveer 40km/h rijdt.

## 10. Besluit

Eerst en vooral wil ik graag zeggen dat dit voor mij een heel leuk, interessant en uitdagend project was waar ik heel veel heb kunnen bijleren over hoe je een complex project zoals dit uitwerkt en hoe je best kan omgaan met het productieproces en externe leveranciers. Zo sprak ik met Duitse, Japanse en Chinese componenten leveranciers om informatie en samples te bemachtigen. Soms bots je op tijd, cultuur en taalbarrières, dit hoort er allemaal bij. Een eerste stap in de “echte” elektronica wereld.

Ik heb een mooi prototype kunnen verwezenlijken dat in staat is om zijn basis functionaliteit te tonen en zijn intrede te doen in de echte wereld. Echter is de volledige potentie softwarematig nog niet bereikt, ik ben heel trots op wat ik zelf heb kunnen realiseren maar om een finaal topproduct te kunnen garanderen trakteer ik mezelf graag op een extra DSP-cursus om dit verder af te werken.

Verder ben ik heel tevreden met de keuzes die ik heb gemaakt om een bachelorproef te vinden die zich richt op FPGA en RF. Hiermee heb ik mijn persoonlijke interesses verder kunnen uitbouwen en een hele hoop ervaring kunnen opdoen die later goed van pas zal komen in mijn professionele carrière.

# 11. Lijst met afbeeldingen

|   |    |
|---|----|
| Figuur 1 OAC Electronics te Herentals   | 10 |
| Figuur 2 Elektromagnetisch spectrum   | 12 |
| Figuur 3 Doppler-Effect   | 14 |
| Figuur 4 IQ-signaal   | 15 |
| Figuur 5 Gemeten IQ-signaal in de praktijk  | 15 |
| Figuur 6 Voorbeeld Chirp met 3 reflecties op verschillende afstanden              | 17 |
| Figuur 7 Afstands/snelheids bepaling zaagtand FMCW-radar                          | 18 |
| Figuur 8 Modulatie patronen voor een FMCW-radar                                   | 18 |
| Figuur 9 Verband afstand & snelheid binnen driehoeksmodulatie                     | 19 |
| Figuur 10 Microcontroller VS FPGA DSP   | 20 |
| Figuur 11 Sampling voorbeeld Shannon-Nyquist                                      | 21 |
| Figuur 12 Resolutie Analooq signaal voorbeeld                                     | 21 |
| Figuur 13 Discrete Fourier Transform Time domain to frequency domain              | 22 |
| Figuur 14 Voorbeeld FFT met Fourier Blokgolf                                      | 23 |
| Figuur 15 Chirp rate example zaagtand   | 24 |
| Figuur 16 Beat Frequencies Example  | 24 |
| Figuur 17 FMCW-RADAR doppler meting   | 25 |
| Figuur 18 FMCW-RADAR doppler bepalen a.d.h.v. fase verschuiving                   | 25 |
| Figuur 19 FMCW-RADAR 3 targets, 2 afstanden, 3 snelheden                          | 26 |
| Figuur 20 FMCW-RADAR 2D-DATA  | 27 |
| Figuur 21 FMCW-RADAR 3D RADAR DATA CUBE   | 27 |
| Figuur 22 RSP1 evaluation board   | 28 |
| Figuur 23 RSP1 Test software  | 28 |
| Figuur 24 Prototype Custom Amplifier PCB  | 29 |
| Figuur 25 InnoSent IVQ-3005 Radar-Module  | 30 |
| Figuur 26 Blokschema IVQ-3005   | 31 |
| Figuur 27 Simpel blokschema radar   | 33 |
| Figuur 28 Spanningsdeler (TPS65265 Datasheet)                                     | 34 |
| Figuur 29 Power-up sequence (TPS65265 Datasheet)                                  | 36 |
| Figuur 30 Xilinx 7 Series: Recommended Operating Conditions (DS189)               | 37 |
| Figuur 31 Required PCB Capacitor Quantities per Device: Spartan-7 Devices (UG483) | 38 |
| Figuur 32 EEPROM Interface, FT2232H Datasheet                                     | 39 |
| Figuur 33 Bootmode configuration, (UG470)   | 40 |
| Figuur 34 Reference Voltage XADC, (UG480)   | 43 |
| Figuur 35 Common Mode Noise Rejection, (UG480)                                    | 44 |
| Figuur 36 SPI write timing IVQ-3005   | 44 |
| Figuur 37 Attenuator voltage vs TX-Power Example                                  | 47 |
| Figuur 38 PCB Layer-Stackup   | 49 |
| Figuur 39 Layout Example TPS65265   | 51 |

|  |   |    |
|--|---|----|
| Figuur 40 CSGA225 Package - Pinout diagram   | Figuur 41 CSGA225 Package - I/O Banks                 | 52 |
| Figuur 42 RADAR PCB, BGA ROOM exit rules   |   | 53 |
| Figuur 43 Recommended Layout Dimensions within BGA Area for 0.8mm Pitch Devices (UG1099) |   | 54 |
| Figuur 44 BGA routing PCB  |   | 54 |
| Figuur 45 voorbeeld BGA Fanout,  |   | 55 |
| Figuur 46 Example Capacitor Land and Mounting Geometries (UG483)                         |   | 56 |
| Figuur 47 Example Cutaway View of PCB with Capacitor Mounting (UG483)                    |   | 56 |
| Figuur 48 Decoupling op de bottom layer van de PCB                                       | Figuur 49 Decoupling op de bottom layer van de PCB 3D | 56 |
| Figuur 50 Via stitching, gezien vanuit de power plane                                    |   | 57 |
| Figuur 51 Design Rule Check output RADAR   |   | 58 |
| Figuur 52 FPGA PCB-Footprint   |   | 58 |
| Figuur 53 Voorbeeld V-Scoring (V-Cut)  |   | 59 |
| Figuur 54 Voorbeeld not fitted through hole componenten altium production variant        |   | 59 |
| Figuur 55 ERP code scanner OAC electronics   |   | 60 |
| Figuur 56 Componenten kast OAC Electronics   |   | 61 |
| Figuur 57 Feeders P&P OAC Electronics  |   | 62 |
| Figuur 58 Stencil Radar board opgespannen kader  |   | 63 |
| Figuur 59 QFN pasta  |   | 63 |
| Figuur 60 BGA pasta  |   | 63 |
| Figuur 61 Stencil machine met pasta  |   | 63 |
| Figuur 62 Stencil machine met PCB  |   | 63 |
| Figuur 63 Tray FPGA  |   | 64 |
| Figuur 64 STF-1005 Tray machine  |   | 64 |
| Figuur 65 Smans Hanwha-Decan S2  |   | 64 |
| Figuur 66 Smans Hanwha-SM482   |   | 64 |
| Figuur 67 .csv file in excel   |   | 64 |
| Figuur 68 AOI productie  |   | 65 |
| Figuur 69 Reflow soldering Vapor phase vs Infrared / Convective heating                  |   | 66 |
| Figuur 70 Typical Solder Reflow Profile for BGA (CSGA225) (UG112/XAPP427)                |   | 66 |
| Figuur 71 PCB na volledige productie   |   | 67 |
| Figuur 72 Near field meting IVQ-3005 op 14dBm, resultaat op ~60cm = 0.4V/m               |   | 68 |
| Figuur 73 Takachi WP13-18-5C   |   | 69 |
| Figuur 74 Program FTDI commando TCL console  |   | 70 |
| Figuur 75 Vivado IP-catalog  |   | 71 |
| Figuur 76 Microblaze met local memory  |   | 72 |
| Figuur 77 Address Editor Vivado Microblaze   |   | 72 |
| Figuur 78 AXI GPIO configuration   |   | 73 |
| Figuur 79 UARTlite configuration   |   | 73 |
| Figuur 80 Programming FLASH  |   | 74 |
| Figuur 81 Program FLASH in Vitis   |   | 74 |
| Figuur 82 Block diagram XADC (UG480)   |   | 75 |

|   |    |
|---|----|
| Figuur 83 16-bit Status Register XADC (UG480)                                   | 75 |
| Figuur 84 XADC Unipolar Transfer Function (UG480)                               | 75 |
| Figuur 85 XADC Unipolar Input Signal Range (UG480)                              | 75 |
| Figuur 86 XADC Bipolar Transfer Function (UG480)                                | 76 |
| Figuur 87 XADC Bipolar Input Signal Range (UG480)                               | 76 |
| Figuur 88 XADC Basic Configuration  | 76 |
| Figuur 89 XADC ALARMS Configuration   | 77 |
| Figuur 90 XADC Channel Sequencer Configuration                                  | 77 |
| Figuur 91 XFFT configuration  | 78 |
| Figuur 92 XDMA configuration  | 78 |
| Figuur 93 Vivado IP block, ADC test   | 79 |
| Figuur 94 Vivado IP block, FFT DMA test   | 80 |
| Figuur 95 Vivado Utilized Resources, Project Manager                            | 80 |
| Figuur 96 Product Selection Guide Xilinx 7 Series, XC7S25                       | 81 |
| Figuur 97 VIVADO defined hardware CW-RADAR IP-CATALOG                           | 82 |
| Figuur 98 Serial Monitor Vitis: Output van een wagen die ongeveer 40km/h rijdt. | 82 |

## 12. Literatuurlijst

'Elektromagnetisch spectrum & foto IQ-signaal + RADAR basisprincipe'

<https://www.infineon.com/cms/en/product/promopages/makeradar/makeradar-school/radar-theory/>

'Penetreerbaarheid materialen'

[https://www.innosent.de/fileadmin/media/dokumente/Downloads/Application\\_Note\\_I\\_-\\_web.pdf](https://www.innosent.de/fileadmin/media/dokumente/Downloads/Application_Note_I_-_web.pdf)

'Foto Doppler-effect'

<https://flypaper.soundfly.com/discover/what-is-the-doppler-effect/>

'FMCW-RADAR'

<https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html>

'FPGA vs Microcontroller DSP'

[https://www.researchgate.net/figure/Difference-between-a-DSP-processor-operation-and-FPGA-based-operation\\_fig2\\_292670980](https://www.researchgate.net/figure/Difference-between-a-DSP-processor-operation-and-FPGA-based-operation_fig2_292670980)

'Shannon Nyquist'

[http://195.134.76.37/applets/AppletNyquist/ Appl\\_Nyquist2.html](http://195.134.76.37/applets/AppletNyquist/ Appl_Nyquist2.html)

'Sampling'

[https://www.researchgate.net/figure/Voice-signal-encoding-a-The-different-sample-rates-13-b-The-different-bit-depths\\_fig2\\_328532594](https://www.researchgate.net/figure/Voice-signal-encoding-a-The-different-sample-rates-13-b-The-different-bit-depths_fig2_328532594)

'Foto DFT'

<https://www.pngall.com/frequency-png/download/70989>

'Foto FFT'

[https://www.youtube.com/watch?v=EyVJtPg\\_Vr0&ab\\_channel=NTiAudio](https://www.youtube.com/watch?v=EyVJtPg_Vr0&ab_channel=NTiAudio)

'RFBEAM eval kit'

<https://rfbeam.ch/product/rsp1-evaluation-kit/>

'InnoSent IVQ-3005'

<https://www.innosent.de/en/sensors/ivq-3005/>

'TPS65265 Datasheet'

[https://www.ti.com/lit/ds/symlink/tps65265.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-ww&ts=1715007515984&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsppproductinfo.tsp%253Fdistld%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Ftps65265](https://www.ti.com/lit/ds/symlink/tps65265.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-ww&ts=1715007515984&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsppproductinfo.tsp%253Fdistld%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Ftps65265)

'Xilinx Datasheets'

<https://docs.amd.com/search/all?content-lang=en-US>

'Takachi Case'

<https://www.takachi-enclosure.com/assets/attachments/images/WP13-18-5C.pdf>

'Foto V-Cut'

<https://www.multi-circuit-boards.eu/en/pcb-design-aid/mechanics/v-scoring.html>

'XADC EXAMPLE'

<https://digilent.com/reference/programmable-logic/cmod-s7/demos/xadc>

'FMCW RADAR DSP'

[https://www.infineon.com/dgdl/Infineon-FMCW\\_RADAR\\_Digital\\_Signal\\_Processing\\_Handout-Training-v01\\_00-EN.pdf?fileId=8ac78c8c8929aa4d018a178075b06be9](https://www.infineon.com/dgdl/Infineon-FMCW_RADAR_Digital_Signal_Processing_Handout-Training-v01_00-EN.pdf?fileId=8ac78c8c8929aa4d018a178075b06be9)

<https://wirelesspi.com/fmcw-radar-part-2-velocity-angle-and-radar-data-cube/>

<https://www.radartutorial.eu/02.basics/rp08.nl.html>

## **13. Bijlagen:**

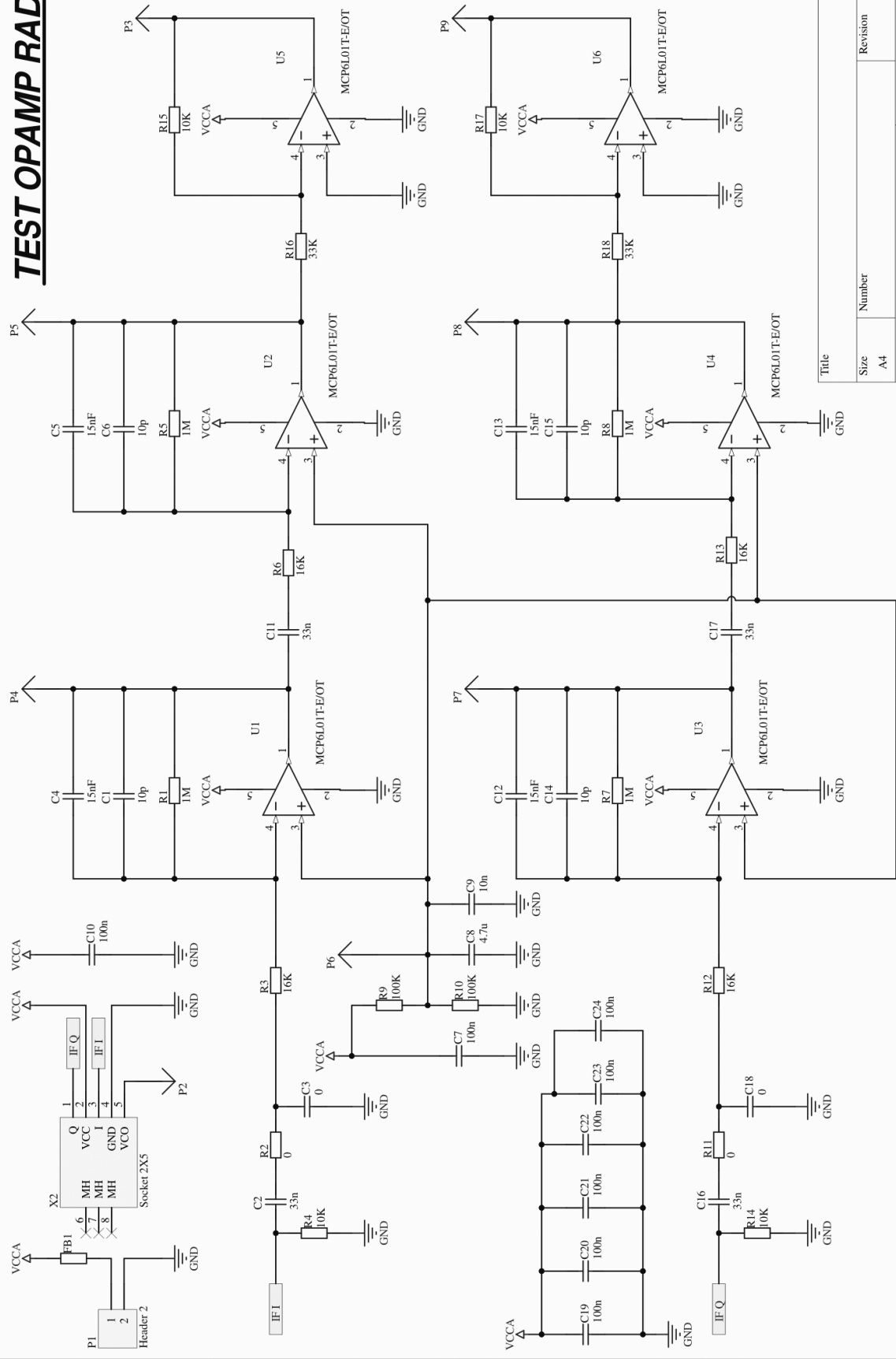
### **13.1. Bijlagen Custom Amplifier K-LC6:**

- 1) SCHEMA**
- 2) PCB TOP LAYER**
- 3) PCB BOTTOM LAYER**
- 4) PCB ALL LAYERS**
- 5) PCB 3D-BOTTOM**
- 6) PCB 3D-TOP**
- 7) BILL OF MATERIALS**

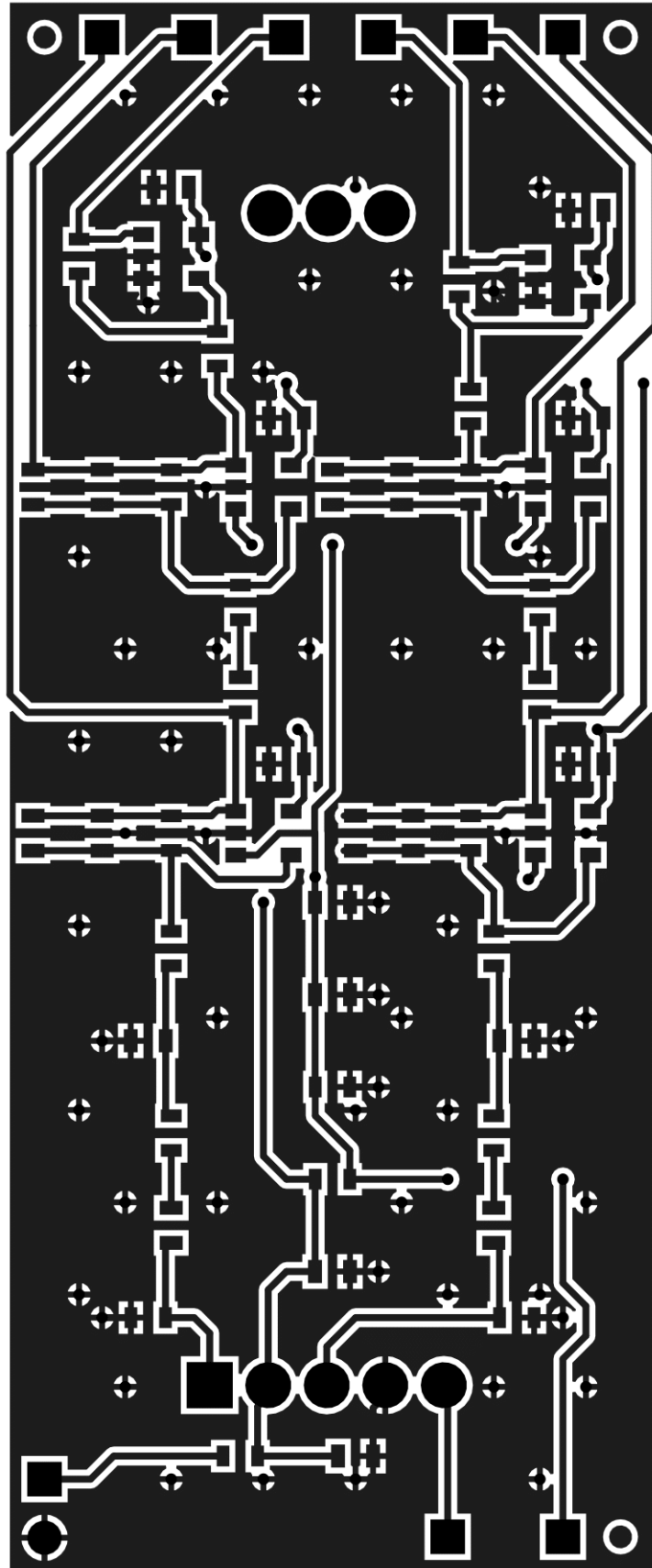


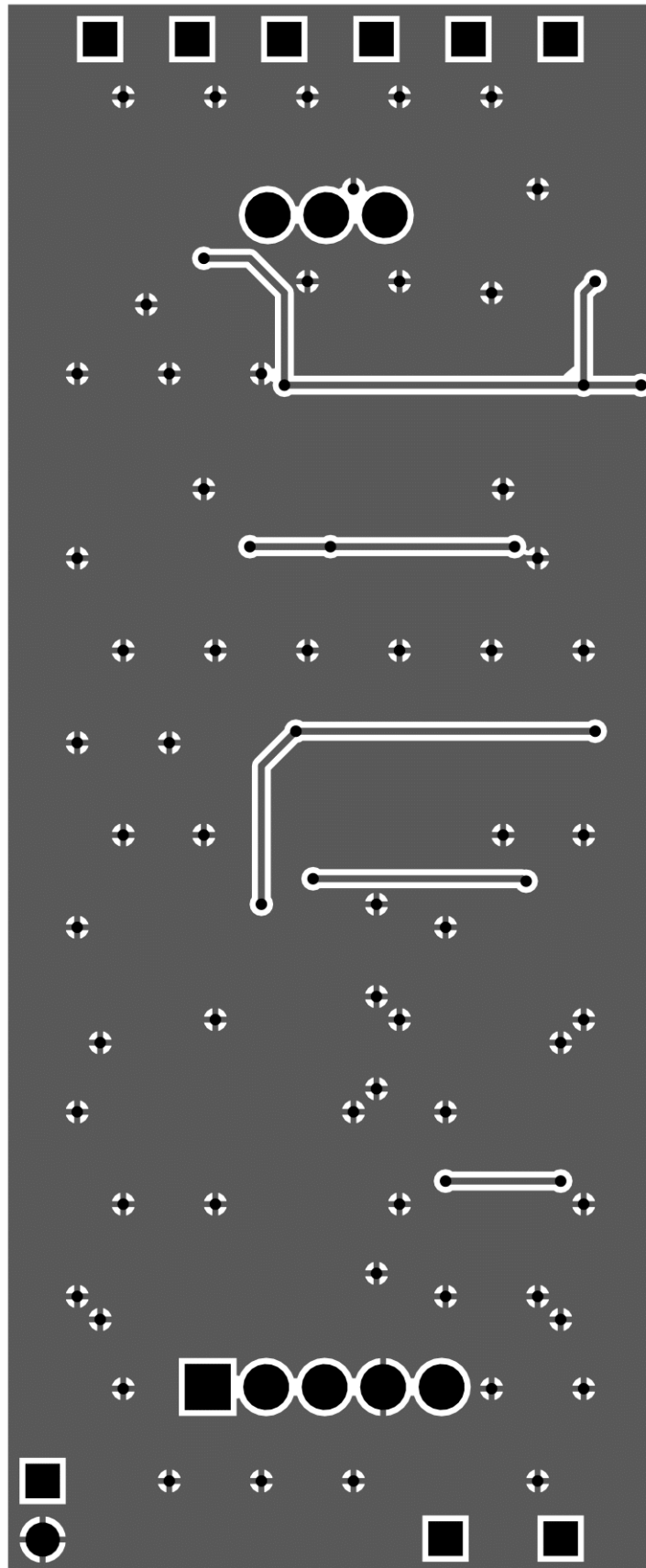
# TEST OPAMP RADAR

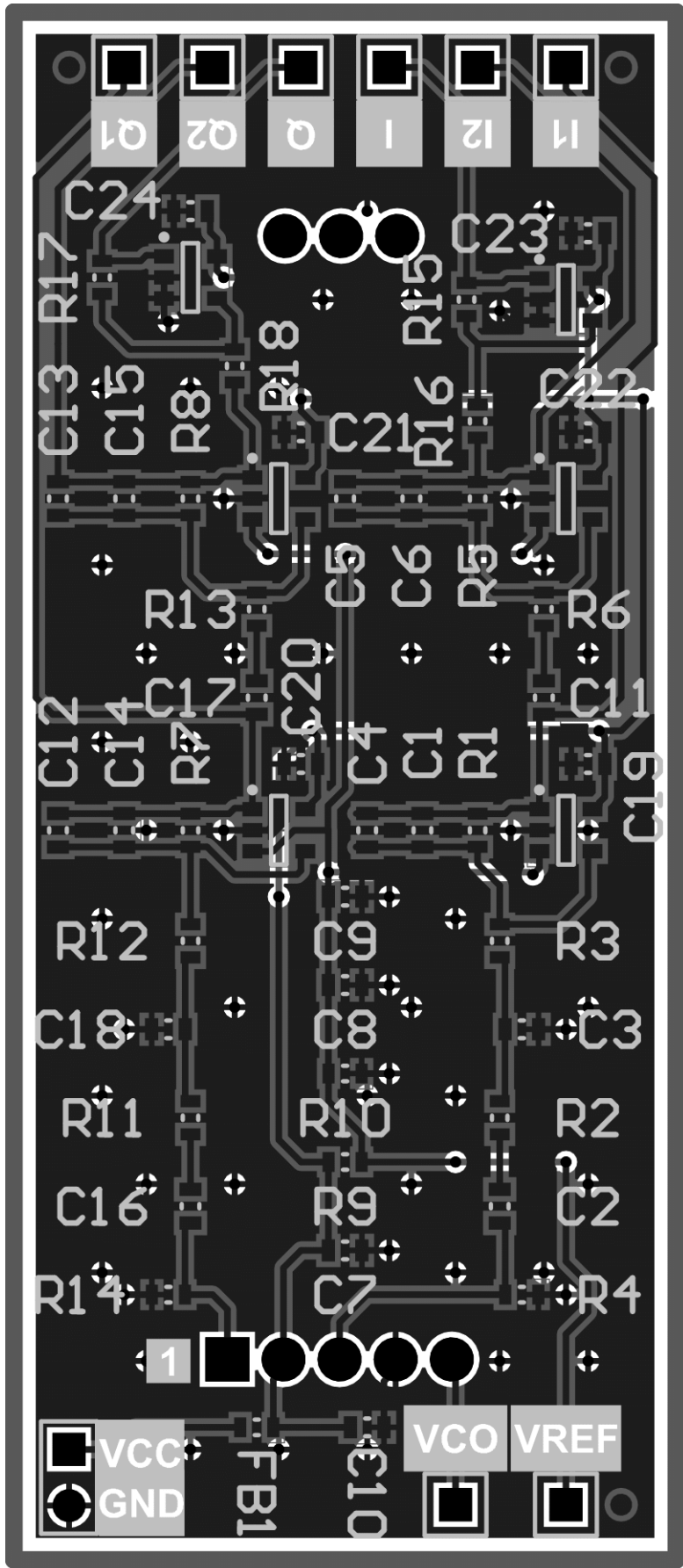
CLAMPING DIODES TOEGEGEN VOOR FPGA BEVEILIGING

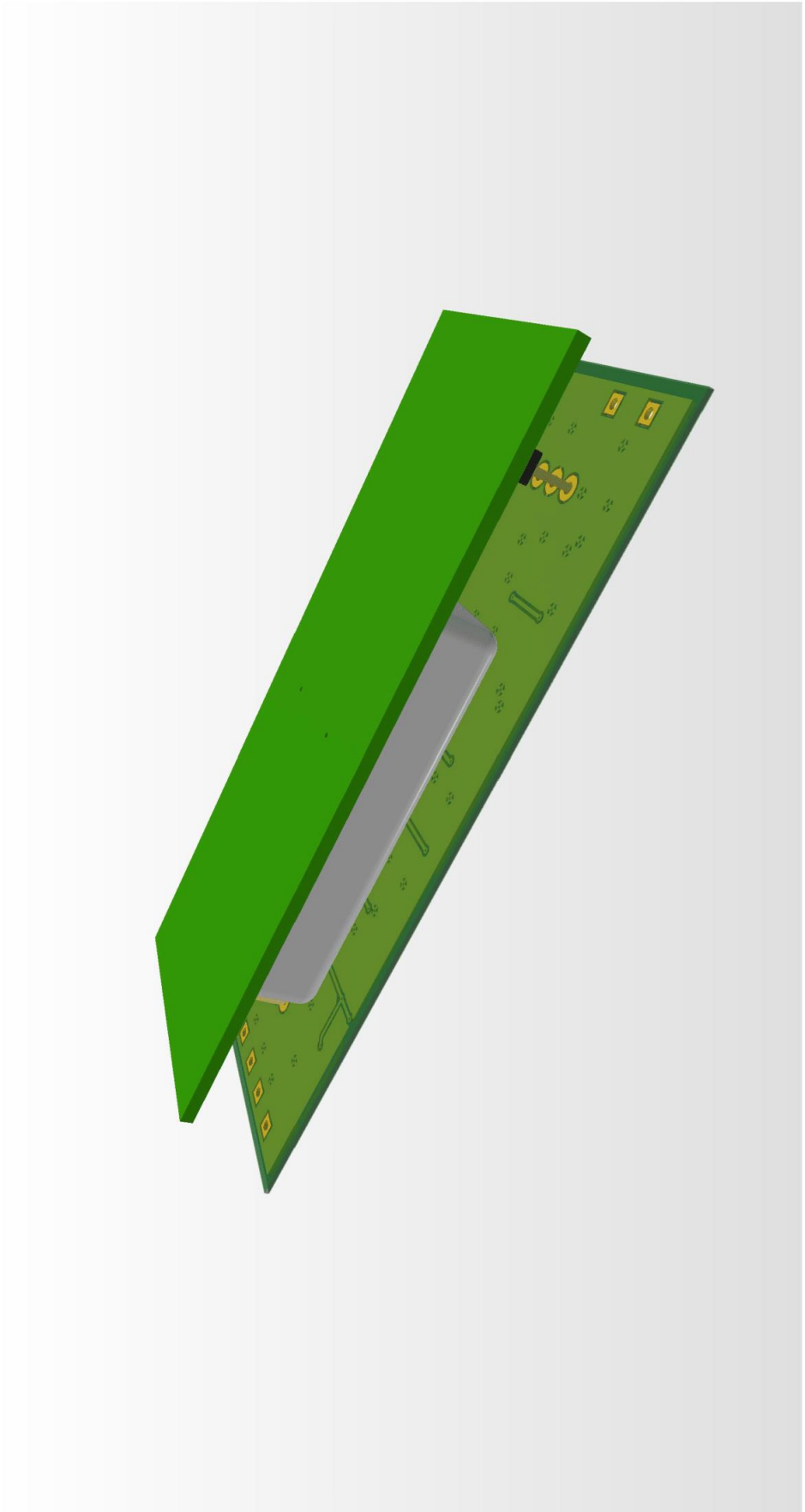


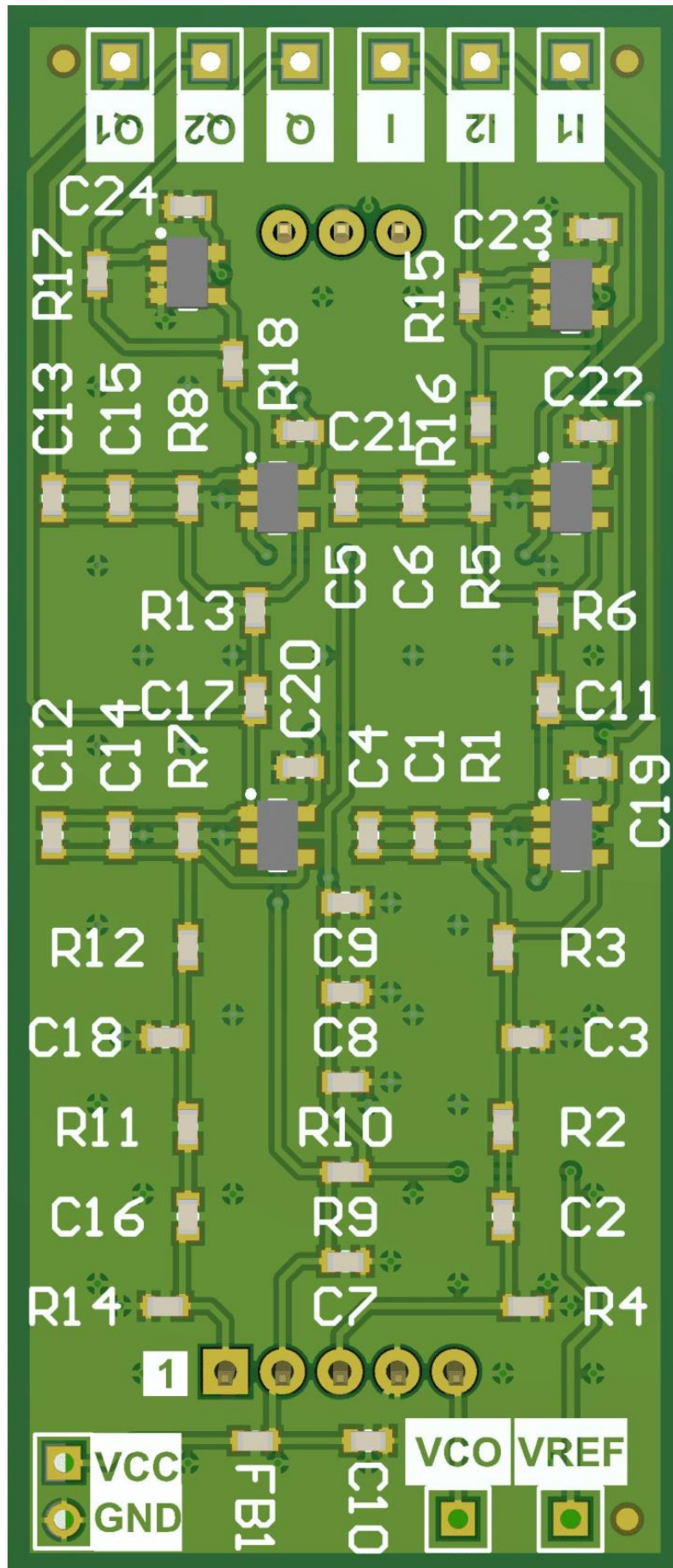
|                              |           |
|------------------------------|-----------|
| Title                        |           |
| Size                         | Number    |
| A4                           |           |
| Date:                        | Revision  |
| 4/05/2024                    |           |
| File:                        | Sheet of  |
| C:\Users\... \TESTAMP.SchDoc | 4         |
|                              | Drawn By: |
|                              |           |









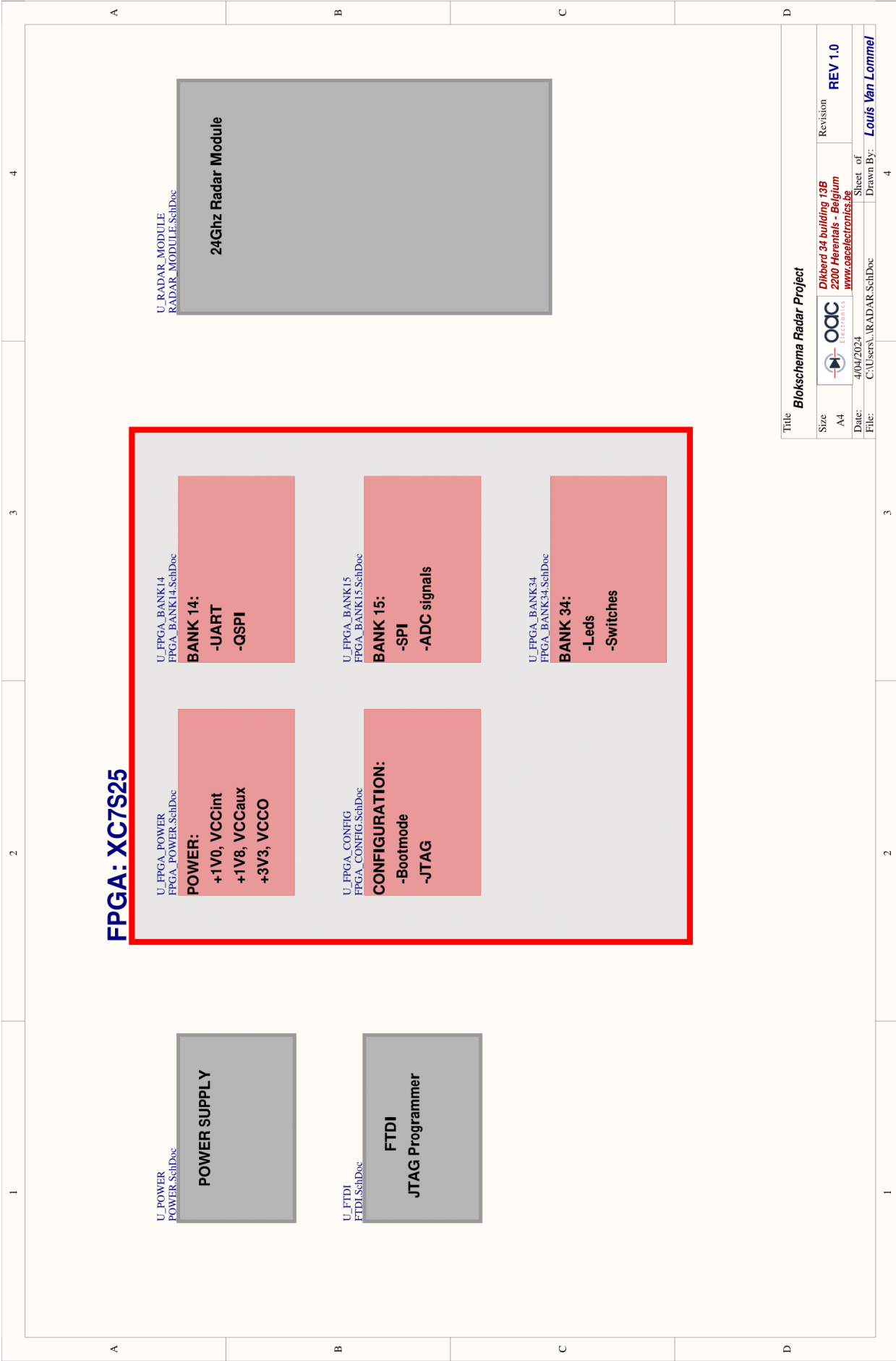


| Comment      | Description             | Designator                            | Footprint    | LibRef           | Quantity | ERP     | Value |
|--------------|-------------------------|---------------------------------------|--------------|------------------|----------|---------|-------|
| 10p          | Capacitor               | C1, C6, C14, C15                      | CAP_0603     | Cap              | 4        | C000285 | 10p   |
| 10uF         | Capacitor               | C2, C11, C16, C17                     | CAP_0603     | Cap              | 4        | C000037 | 33n   |
| 0            | Capacitor               | C3, C18                               | CAP_0603     | Cap              | 2        | DNP     | 0     |
| 15n          | Capacitor               | C4, C5, C12, C13                      | CAP_0603     | Cap              | 4        | C000351 | 15nF  |
| 100n         | Capacitor               | C7, C10, C19, C20, C21, C22, C23, C24 | CAP_0603     | Cap              | 8        | C000137 | 100n  |
| 4.7u         | Capacitor               | C8                                    | CAP_0603     | Cap              | 1        | C000003 | 4.7u  |
| 10n          | Capacitor               | C9                                    | CAP_0603     | Cap              | 1        | C000041 | 10n   |
| Header 2     | Resistor                | FB1                                   | RES_0603     | Res2             | 1        | L000085 |       |
| Header 2     | Header, 2-Pin           | P1                                    | HDR1X2       | Header2          | 1        |         |       |
| TP           | Test Point              | P2, P3, P4, P5, P6, P7, P8, P9        | PIN1         | Plug             | 8        |         |       |
| 1M           | Resistor                | R1, R5, R7, R8                        | RES_0603     | Res2             | 4        | F000024 | 1M    |
| 0            | Resistor                | R2, R11                               | RES_0603     | Res2             | 2        | F000010 | 0     |
| 16K          | Resistor                | R3, R6, R12, R13                      | RES_0603     | Res2             | 4        | F000645 | 16K   |
| 10K          | Resistor                | R4, R14, R15, R17                     | RES_0603     | Res2             | 4        | F000180 | 10K   |
| 100K         | Resistor                | R9, R10                               | RES_0603     | Res2             | 2        | F000086 | 100K  |
| 33K          | Resistor                | R16, R18                              | RES_0603     | Res2             | 2        | F000294 | 33K   |
| MCP6L01TE/OT | None                    | U1, U2, U3, U4, U5, U6                | SOT-23-OT5_L | CMP-0187-00432-3 | 6        | U000395 |       |
| Socket 2X5   | Header, 5-Pin, Dual row | X2                                    | K-L06        | MHDR2X5          | 1        |         |       |

## **13.2. BIJLAGEN RADAR BOARD:**

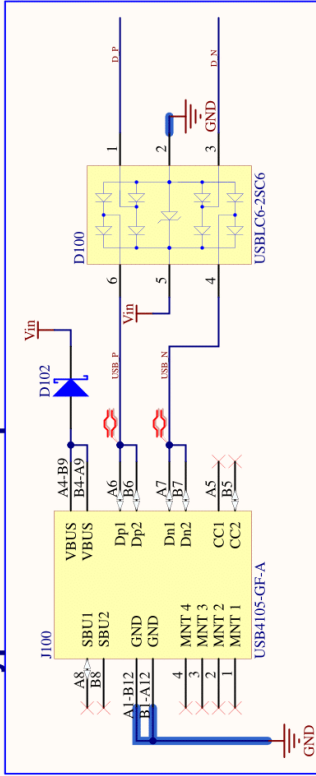
- 1) BLOKSCHEMA**
- 2) POWER**
- 3) FPGA POWER**
- 4) FTDI**
- 5) FPGA CONFIGURATION**
- 6) FPGA BANK14**
- 7) FPGA BANK15**
- 8) FPGA BANK34**
- 9) RADAR MODULE**
- 10) PCB LAYER1: TOP**
- 11) PCB LAYER2: GROUND**
- 12) PCB LAYER3: POWER**
- 13) PCB LAYER4: BOTTOM**
- 14) PCB ALL LAYERS**
- 15) PCB 3D-TOP**
- 16) PCB 3D-BOTTOM**
- 17) BILL OF MATERIALS**



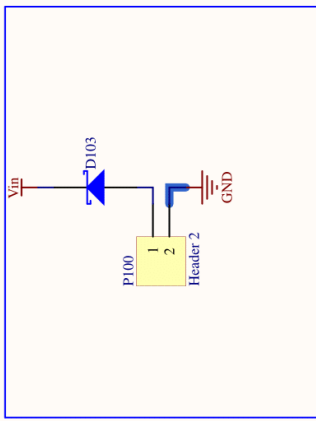


|       |                            |                                 |                         |
|-------|----------------------------|---------------------------------|-------------------------|
| Title |                            | <b>Blokschema Radar Project</b> |                         |
| Size  | A4                         | Revision                        | <b>REV 1.0</b>          |
| Date: | 4/04/2024                  | Sheet of                        | 4                       |
| File: | C:\Users\... \RADAR.SchDoc | Drawn By:                       | <b>Louis Van Lommel</b> |

### USB Type C met ESD protection



### External Power: MAX 17V



### Calculated Values:

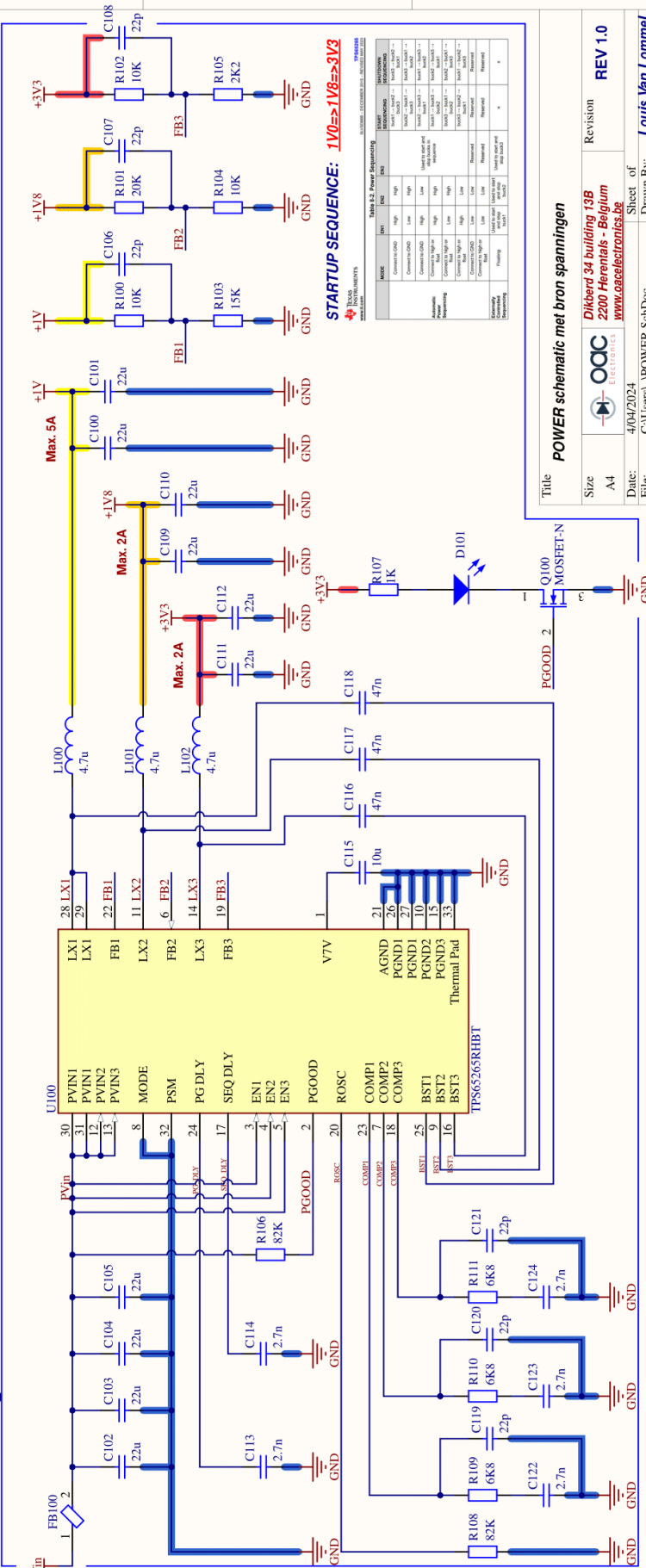
$R2 = R1 * (0.6/V_{out} - 0.6)$   
 $15K = 10K * (0.6/1.0V - 0.6)$   
 $10K = 20K * (0.6/1.8V - 0.6)$   
 $2K2 = 10K * (0.6/3.3V - 0.6)$

$L = (V_{inmax} - V_{out}) / (I_o * LIR) * (V_{out} / (V_{inmax} * f_{sw}))$   
 $L1 = (17V - 1V) / (5A * 1.5A) * (1.0 / (17V * 612K)) = 0.2\mu H$   
 $L2 = (17V - 1.8) / (2A * 0.6A) * (1.8 / (17V * 612K)) = 2.2\mu H$   
 $L3 = (17V - 3.3) / (2A * 0.6A) * (3.3 / (17V * 612K)) = 3.6\mu H \Rightarrow 4.7\mu H$

Voor BOM consideration word alles 4.7uH genomen LIR= 30% van I\_max

$C_{OUT} = (I_o * f_{sw}) / (V_{in} * V_{out})$   
 $C_{out1} V0 = (2 * 5) / (612K * 1V) = 22\mu F$   
 $C_{out1} V8 = (2 * 2) / (612K * 1.8V) = 1.2\mu F \Rightarrow 22\mu F$   
 $C_{out1} V3 = (2 * 2) / (612K * 3.3V) = 22\mu F$

### Tri-Buck Regulator:



MAX17 Power Sequencing

| MODE      | EN1  | EN2  | EN3  | PGOOD | ROSC | COMP1 | COMP2 | COMP3 | BST1 | BST2 | BST3 | AGND | PGND1 | PGND2 | PGND3 | Thermal Pad |
|-----------|------|------|------|-------|------|-------|-------|-------|------|------|------|------|-------|-------|-------|-------------|
| Standby   | High | High | High | High  | High | High  | High  | High  | High | High | High | High | High  | High  | High  | High        |
| Power-On  | High | High | High | High  | High | High  | High  | High  | High | High | High | High | High  | High  | High  | High        |
| Power-Off | Low  | Low  | Low  | Low   | Low  | Low   | Low   | Low   | Low  | Low  | Low  | Low  | Low   | Low   | Low   | Low         |
| Shutdown  | Low  | Low  | Low  | Low   | Low  | Low   | Low   | Low   | Low  | Low  | Low  | Low  | Low   | Low   | Low   | Low         |

**POWER schematic met bron spanningen**  
 Title  
 Size A4  
 Date: 4/04/2024  
 File: C:\Users\... \POWER\_SchDoc  
 Revision REV 1.0  
 Sheet of  
 Drawn By: **Louis Van Lommel**

# FPGA POWER

Power-on sequence: +1V0, +1V8, +3V4

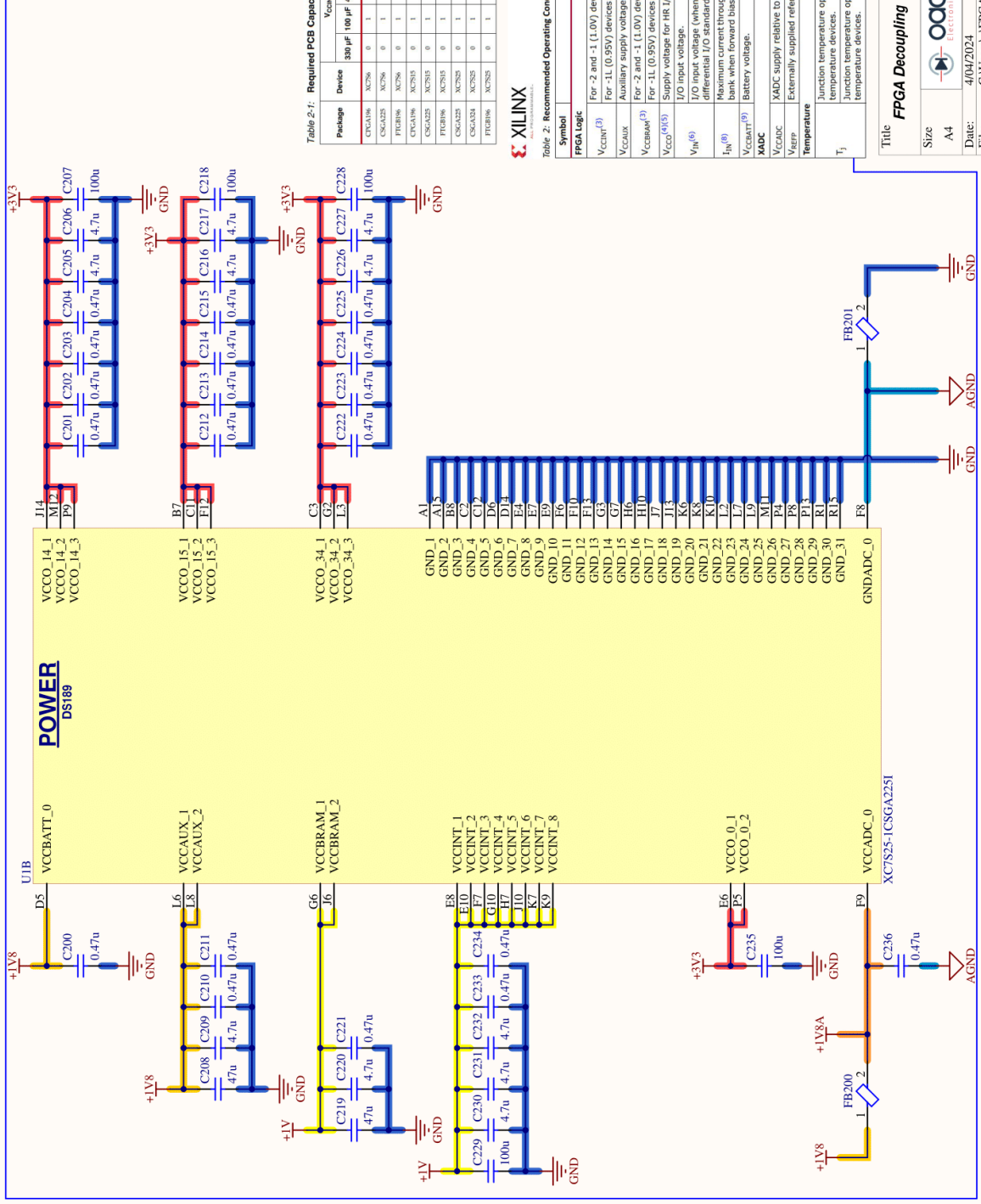


Table 2-1: Required PCB Capacitor Quantities per Device: Spartan-7 Devices<sup>(1)(2)</sup>

| Package  | Device | Vccint |        |       | Vccaux |       |        | Vcco (per Bank) |        |       |        |   |   |   |
|----------|--------|--------|--------|-------|--------|-------|--------|-----------------|--------|-------|--------|---|---|---|
|          |        | 330 µF | 100 µF | 47 µF | 100 µF | 47 µF | 100 µF | 47 µF           | 100 µF | 47 µF | 100 µF |   |   |   |
| CPLB106  | XC7S06 | 0      | 1      | 1     | 1      | 0     | 1      | 1               | 1      | 1     | 2      | 1 | 2 | 4 |
| FCPLB106 | XC7S06 | 0      | 1      | 1     | 1      | 0     | 1      | 1               | 1      | 1     | 2      | 1 | 2 | 4 |
| CPLB106  | XC7S15 | 0      | 1      | 1     | 1      | 0     | 1      | 1               | 1      | 1     | 2      | 1 | 2 | 4 |
| FCPLB106 | XC7S15 | 0      | 1      | 1     | 1      | 0     | 1      | 1               | 1      | 1     | 2      | 1 | 2 | 4 |
| CPLB106  | XC7S25 | 0      | 1      | 1     | 1      | 0     | 1      | 1               | 1      | 1     | 2      | 1 | 2 | 4 |
| FCPLB106 | XC7S25 | 0      | 1      | 1     | 1      | 0     | 1      | 1               | 1      | 1     | 2      | 1 | 2 | 4 |
| CPLB106  | XC7S35 | 0      | 1      | 1     | 1      | 0     | 1      | 1               | 1      | 1     | 2      | 1 | 2 | 4 |
| FCPLB106 | XC7S35 | 0      | 1      | 1     | 1      | 0     | 1      | 1               | 1      | 1     | 2      | 1 | 2 | 4 |

Table 2: Recommended Operating Conditions<sup>(1)(2)</sup>

| Symbol                 | Description   | Min   | Typ  | Max         | Units |
|------------------------|---|-------|------|-------------|-------|
| <b>FPGA Logic</b>      |   |       |      |             |       |
| Vccint <sup>(3)</sup>  | For -2 and -1 (1.0V) devices: internal supply voltage.  | 0.95  | 1.00 | 1.05        | V     |
| Vccaux                 | For -2 and -1 (1.0V) devices: internal supply voltage.  | 0.92  | 0.95 | 0.98        | V     |
| Vccbram <sup>(3)</sup> | Auxiliary supply voltage.   | 1.71  | 1.80 | 1.89        | V     |
| Vcco <sup>(4)(5)</sup> | For -2 and -1 (1.0V) devices: black RAM supply voltage.   | 0.95  | 1.00 | 1.05        | V     |
|                        | For -1L (0.9V) devices: black RAM supply voltage.   | 0.92  | 0.95 | 0.98        | V     |
| Vih <sup>(6)</sup>     | Supply voltage for HR I/O banks.  | 1.14  | -    | 3.465       | V     |
| Ih <sup>(6)</sup>      | I/O input voltage.  | -0.20 | -    | Vcco + 0.20 | V     |
| Vccint <sup>(7)</sup>  | I/O input voltage (when Vcco = 3.3V) for Vieg and differential I/O standards except TMD5_33. <sup>(7)</sup> | -0.20 | -    | 2.625       | V     |
| Vccint <sup>(8)</sup>  | Maximum current through any pin in a powered or unpowered bank when forward biasing the clamp diode.        | -     | -    | 10          | mA    |
| Vccint <sup>(9)</sup>  | Bittery voltage.  | 1.0   | -    | 1.89        | V     |
| <b>ADC</b>             |   |       |      |             |       |
| Vccadc                 | XADC supply relative to GNDADC.   | 1.71  | 1.80 | 1.89        | V     |
| Vrefp                  | Externally supplied reference voltage.  | 1.20  | 1.25 | 1.30        | V     |
| <b>Temperature</b>     |   |       |      |             |       |
| Tj                     | Junction temperature operating range for commercial (C) temperature devices.                                | 0     | -    | 85          | °C    |
|                        | Junction temperature operating range for industrial (I) temperature devices.                                | -40   | -    | 100         | °C    |

**POWER DS189**

VCCBATT\_0

VCCAU\_X\_1

VCCAU\_X\_2

VCCBRAM\_1

VCCBRAM\_2

VCCINT\_1

VCCINT\_2

VCCINT\_3

VCCINT\_4

VCCINT\_5

VCCINT\_6

VCCINT\_7

VCCINT\_8

VCCO\_0\_1

VCCO\_0\_2

VCCADC\_0

VCCO\_14\_1

VCCO\_14\_2

VCCO\_14\_3

VCCO\_15\_1

VCCO\_15\_2

VCCO\_15\_3

VCCO\_34\_1

VCCO\_34\_2

VCCO\_34\_3

GND\_1

GND\_2

GND\_3

GND\_4

GND\_5

GND\_6

GND\_7

GND\_8

GND\_9

GND\_10

GND\_11

GND\_12

GND\_13

GND\_14

GND\_15

GND\_16

GND\_17

GND\_18

GND\_19

GND\_20

GND\_21

GND\_22

GND\_23

GND\_24

GND\_25

GND\_26

GND\_27

GND\_28

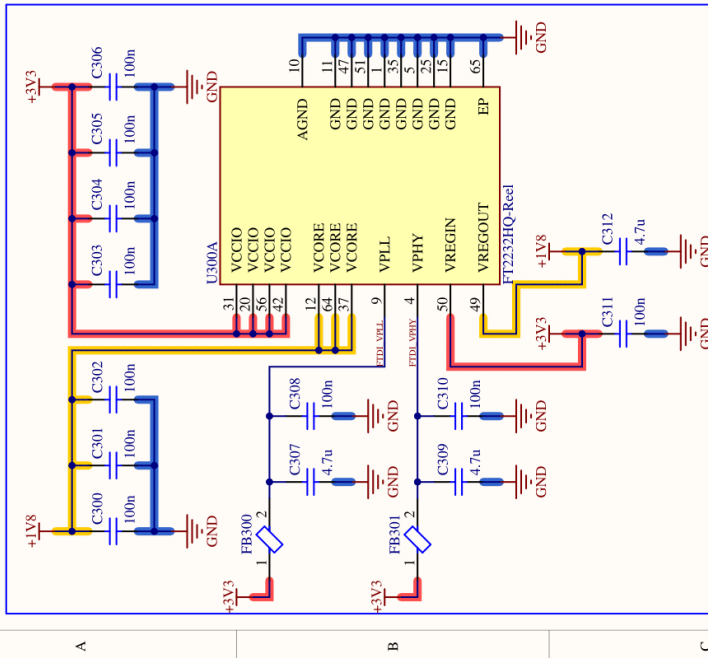
GND\_29

GND\_30

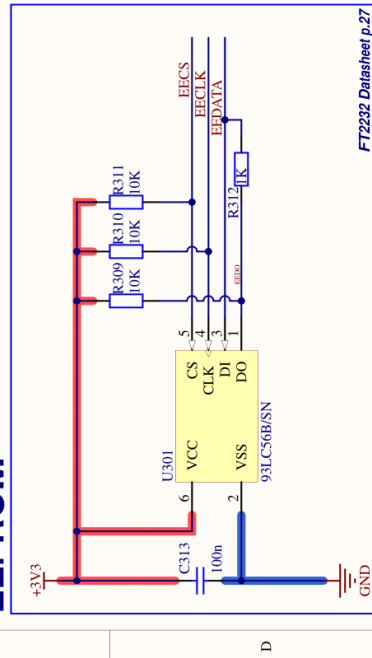
GND\_31

GNDADC\_0

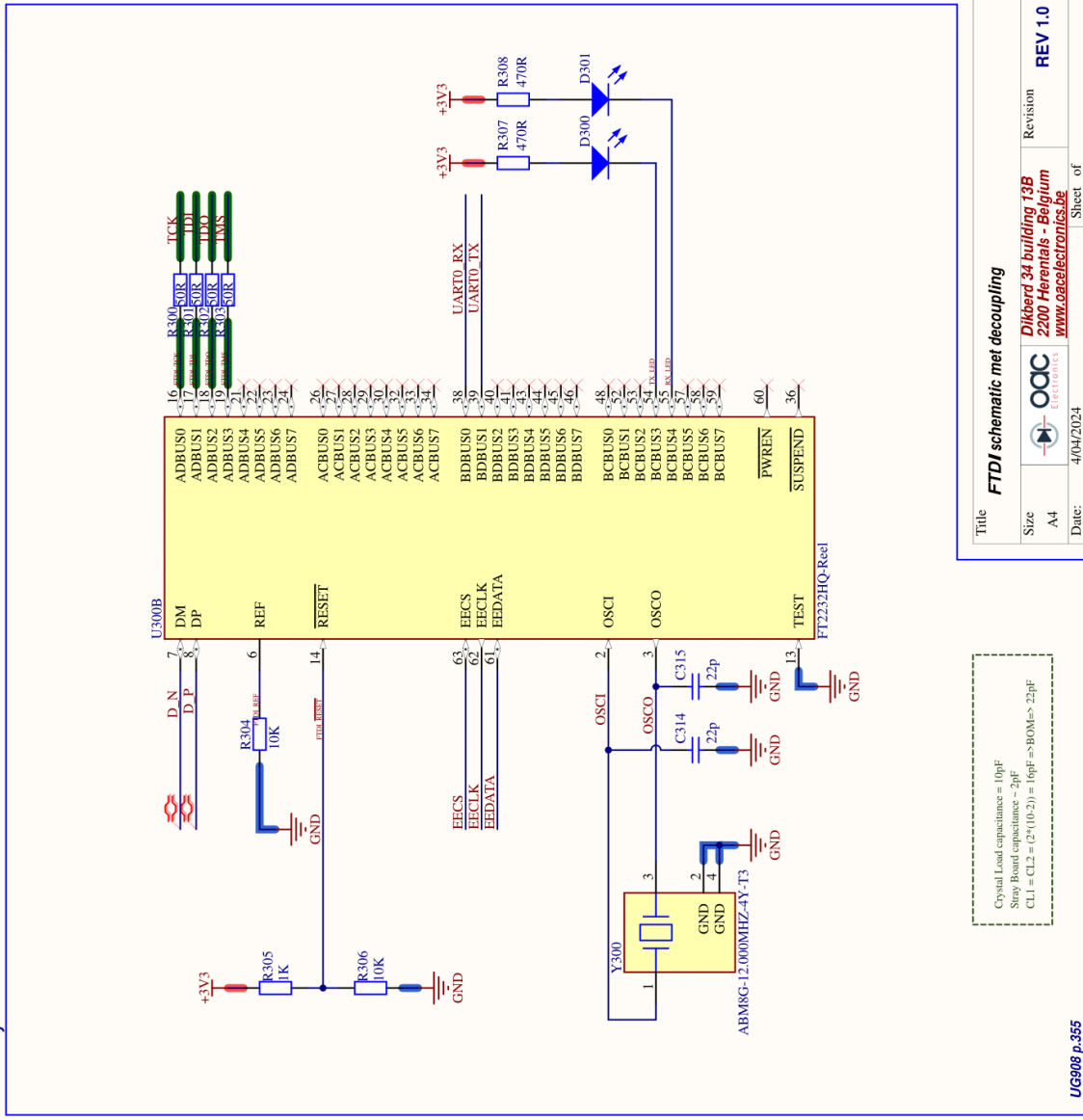
### FTDI POWER



### EEPROM



### FTDI, USB to UART/JTAG



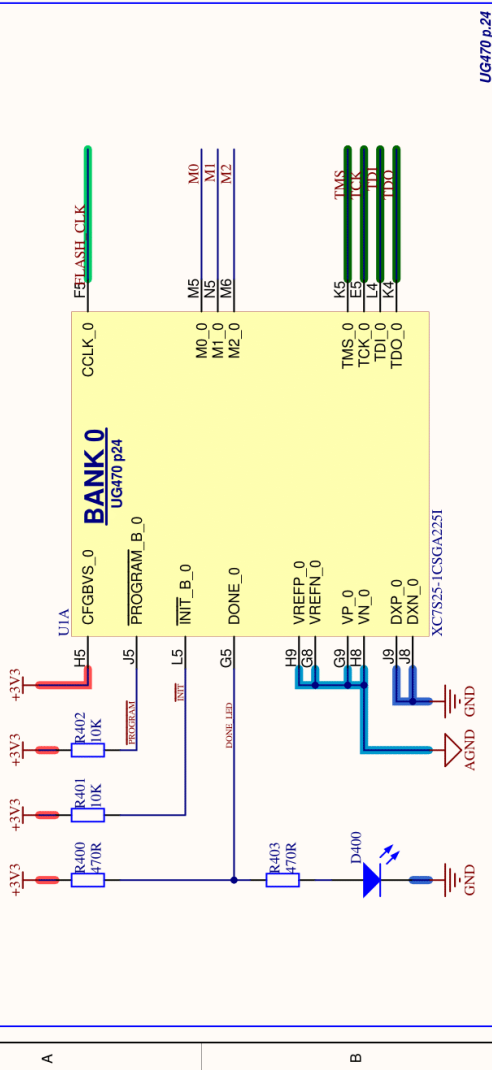
Crystal Load capacitance = 10pF  
 Strip Board capacitance = 2pF  
 $CL1 = CL2 = (2 * (10-2)) = 16pF \Rightarrow BOM \Rightarrow 22pF$

|       |                           |  |                  |
|-------|---------------------------|--|------------------|
| Title |                           | FTDI schematic met decoupling  |                  |
| Size  | A4                        | Revision   | REV 1.0          |
| Date: | 4/04/2024                 | Sheet of   | 4                |
| File: | C:\Users\... \FTDI_SchDoc | Drawn By:  | Louis Van Lommel |
|       |                           | Dijkberd 34 building 13B<br>2200 Herentals - Belgium<br><a href="http://www.ooc.be">www.ooc.be</a> |                  |

UG908 p.355

FT2232 Datasheet p.27

# FPGA Configuration Pins



# Bootmode Pins

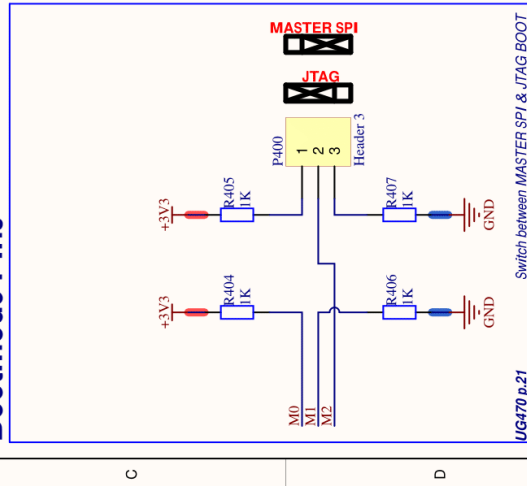
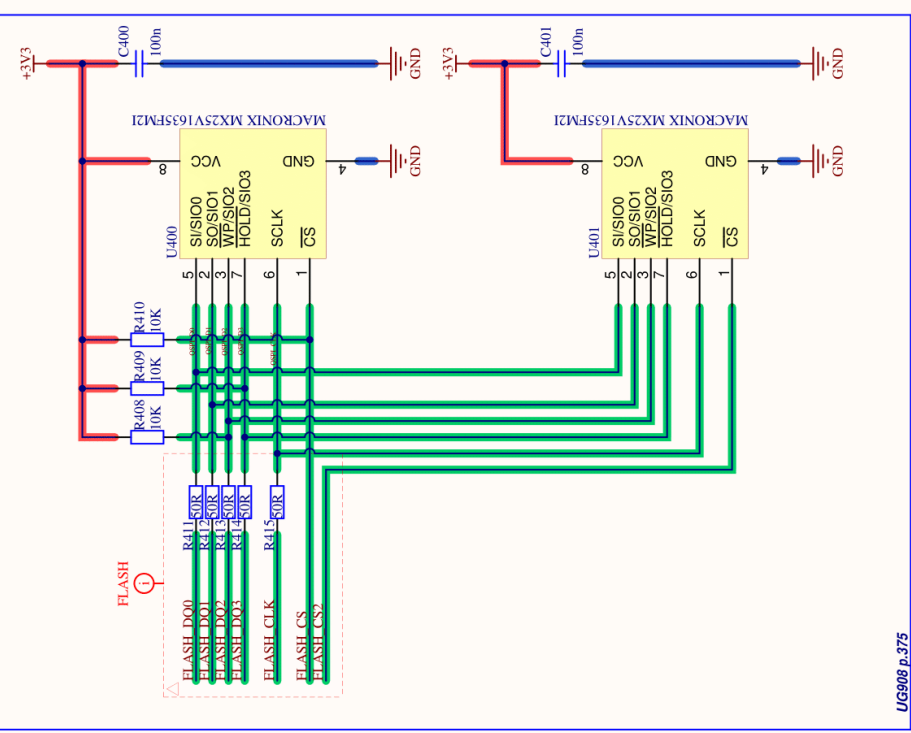


Table 2-1: 7 Series FPGA Configuration Modes

| Configuration Mode          | M[2:0] | Bus Width                   | CCLK Direction |
|-----------------------------|--------|-----------------------------|----------------|
| Master Serial               | 000    | x1                          | Output         |
| Master SPI                  | 001    | x1, x2, x4                  | Output         |
| Master BPI                  | 010    | x8, x16                     | Output         |
| Master SelectMAP            | 100    | x8, x16                     | Output         |
| JTAG                        | 101    | x1                          | Not Applicable |
| Slave SelectMAP             | 110    | x8, x16, x32 <sup>(1)</sup> | Input          |
| Slave Serial <sup>(2)</sup> | 111    | x1                          | Input          |

# Flash Memory, Serial NOR, 16Mbit, 2M x 8bit



Title: **FPGA configuratie BANK 0 + FLASH memory**

Size: A4

Date: 4/04/2024

Revision: **REV 1.0**

Sheet of: 4

Filer: C:\Users\... \FPGA\_CONFIG\_SchDoc

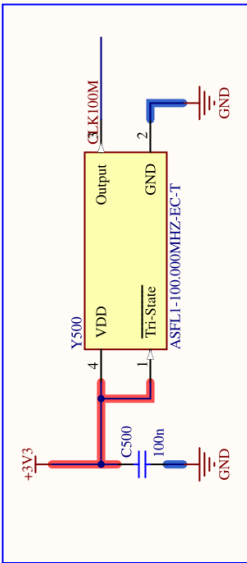
Drawn By: **Louis Van Lommel**

OCG ELECTRONICS

Dijkbaard 34 building 13B  
2200 Herentals - Belgium  
[www.ocgelectronics.be](http://www.ocgelectronics.be)

|   | 1  | 2  | 3  | 4              |
|---|--|--|--|----------------|
| A | <p>U1C</p> <p>FLASH_D00<br/>H14<br/>FLASH_D01<br/>H15</p> <p>FLASH_D02<br/>J12<br/>FLASH_D03<br/>K13</p> <p>K14<br/>L13<br/>L15</p> <p>J15<br/>K15</p> <p>FLASH_CS<br/>L11<br/>L12</p> <p>L13<br/>L14</p> <p>M13<br/>N13</p> <p>FLASH_CS2<br/>M14<br/>M15</p> <p>SCLK<br/>N14<br/>SYNC<br/>N15</p> <p>Internal Pullup config_PUDDC<br/>K11</p> <p>R508<br/>470R<br/>DISPULLUP</p> <p>R509<br/>470R<br/>ENPULLUP</p> <p>GND</p> | <p>BANK 14<br/>UG475 p28</p> <p>IO_L11P_T1_SRCC_14<br/>IO_L11N_T1_SRCC_14</p> <p>IO_L12P_T1_MRCC_14<br/>IO_L12N_T1_MRCC_14</p> <p>IO_L13P_T2_MRCC_14<br/>IO_L13N_T2_MRCC_14</p> <p>IO_L14P_T0_D04_14<br/>IO_L14N_T0_D05_14</p> <p>IO_L15P_T0_D06_14<br/>IO_L15N_T0_D07_14</p> <p>IO_L16P_T0_FCS_B_14<br/>IO_L16N_T0_D08_VREF_14</p> <p>IO_L17P_T1_D09_14<br/>IO_L17N_T1_D10_14</p> <p>IO_L18P_T1_D11_14<br/>IO_L18N_T1_D12_14</p> <p>IO_L19P_T1_DQ8_14<br/>IO_L19N_T1_DQ8_D13_14</p> <p>IO_L10P_T1_D14_14<br/>IO_L10N_T1_D15_14</p> <p>IO_L115P_T2_DQ8_RDWR_B_14<br/>IO_L115N_T2_DQ8_DOUT_CS0_B_14</p> <p>IO_L13P_T0_DQ8_PUDDC_B_14<br/>IO_L13N_T0_DQ8_EMCCLK_14</p> <p>IO_0_14<br/>IO_25_14</p> | <p>P14<br/>P15</p> <p>R13<br/>R14</p> <p>M9<br/>M10</p> <p>N12<br/>N12</p> <p>P10<br/>P11</p> <p>R9<br/>R10</p> <p>R11<br/>R12</p> <p>M7<br/>M8</p> <p>N6<br/>N7</p> <p>N8<br/>N9</p> <p>R7<br/>R8</p> <p>P6<br/>P7</p> <p>R5<br/>R6</p> | <p>CLK100M</p> |
| B |  |  |  |                |
| C |  |  |  |                |
| D |  |  |  |                |

### 100Mhz Clock



Title: **FPGA BANK 14 Communicatie + Clock**

Size: A4

Date: 4/04/2024

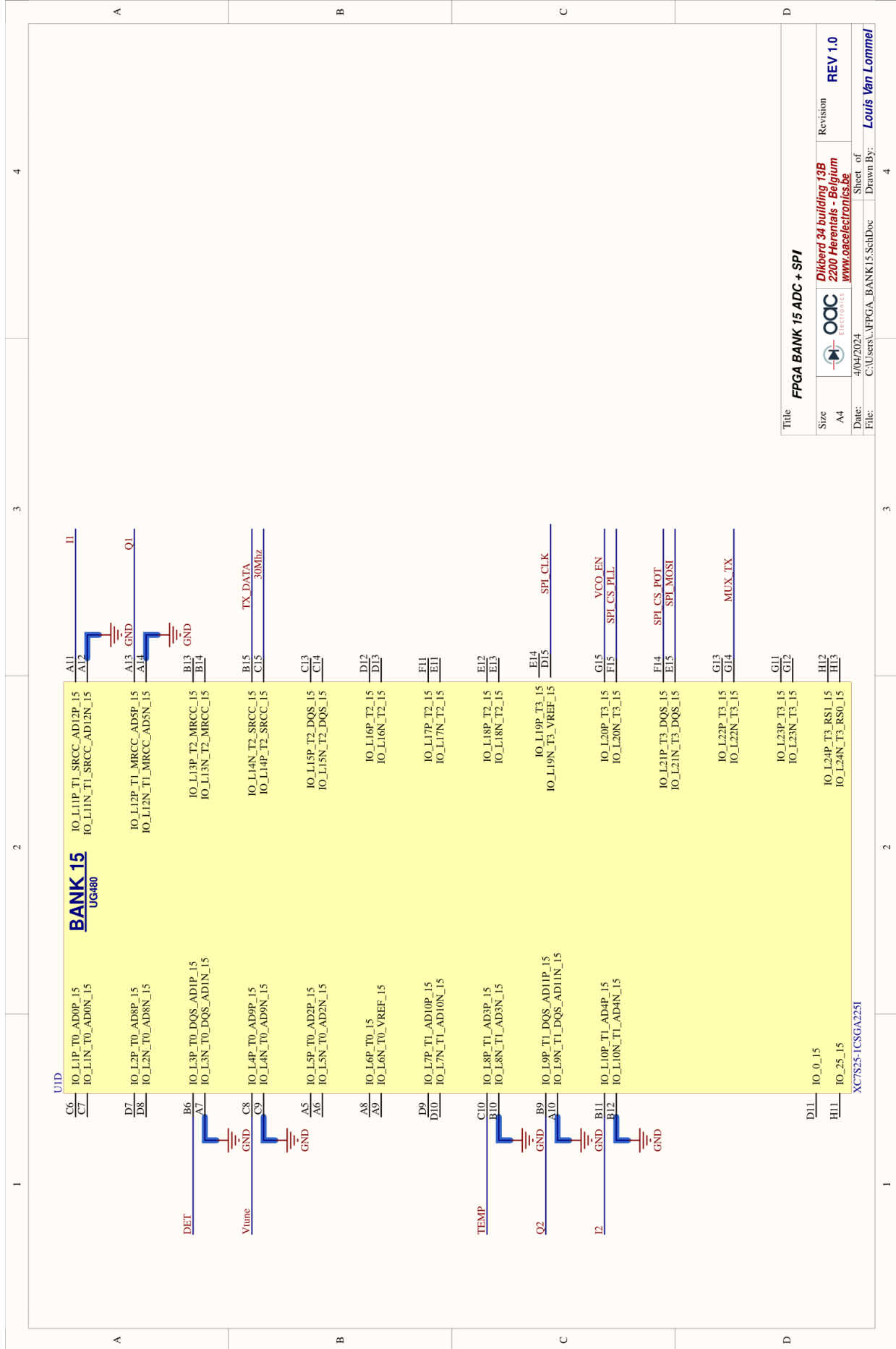
File: C:\Users\... \FPGA\_BANK14.SchDoc

Revision: **REV 1.0**

Sheet of: 4

Drawn By: **Louis Van Lommel**

Company: **OOC Electronics**  
 Dijkberd 34 building 13B  
 2200 Herentals - Belgium  
[www.oocelectronics.be](http://www.oocelectronics.be)



|       |                                |                        |                  |
|-------|--------------------------------|------------------------|------------------|
| Title |                                | FPGA BANK 15 ADC + SPI |                  |
| Size  | A4                             | Revision               | REV 1.0          |
| Date: | 4/04/2024                      | Sheet of               | 4                |
| File: | C:\Users\A\FPGA_BANK\15_SchDoc | Drawn By:              | Louis Van Lommel |

|   |  |   |  |   |  |  |  |  |
|---|--|---|--|---|--|--|--|--|
| 1 | A  | <p>UIE</p> <p>A4 IO_L11P_T0_34<br/>A3 IO_L11N_T0_34</p> <p>B2 IO_L12P_T0_34<br/>A2 IO_L12N_T0_34</p> <p>B4 IO_L13P_T0_DQS_34<br/>B3 IO_L13N_T0_DQS_34</p> <p>C1 IO_L14P_T0_34<br/>B1 IO_L14N_T0_34</p> <p>C5 IO_L15P_T0_34<br/>B5 IO_L15N_T0_34</p> <p>D2 IO_L16P_T0_34<br/>D1 IO_L16N_T0_VREF_34</p> <p>D4 IO_L17P_T1_34<br/>C4 IO_L17N_T1_34</p> <p>E2 IO_L18P_T1_34<br/>E1 IO_L18N_T1_34</p> <p>E3 IO_L19P_T1_DQS_34<br/>D3 IO_L19N_T1_DQS_34</p> <p>F2 IO_L10P_T1_34<br/>F1 IO_L10N_T1_34</p> <p>G4 IO_0_34<br/>N4 IO_25_34</p> <p>XC7S25-1CSGA2251</p> | 2  | <p>BANK 34<br/>UG475 p28</p> <p>F4 IO_L11P_T1_SRCC_34<br/>E4 IO_L11N_T1_SRCC_34</p> <p>H1 IO_L12P_T1_MRCC_34<br/>G1 IO_L12N_T1_MRCC_34</p> <p>H4 IO_L13P_T2_MRCC_34<br/>H3 IO_L13N_T2_MRCC_34</p> <p>I2 IO_L14P_T2_SRCC_34<br/>I1 IO_L14N_T2_SRCC_34</p> <p>I4 IO_L15P_T2_DQS_34<br/>I3 IO_L15N_T2_DQS_34</p> <p>K1 IO_L16P_T2_34<br/>J1 IO_L16N_T2_34</p> <p>K3 IO_L17P_T2_34<br/>K2 IO_L17N_T2_34</p> <p>M1 IO_L18P_T2_34<br/>L1 IO_L18N_T2_34</p> <p>M4 IO_L19P_T3_34<br/>M3 IO_L19N_T3_VREF_34</p> <p>N2 IO_L20P_T3_34<br/>N1 IO_L20N_T3_34</p> <p>P3 IO_L21P_T3_DQS_34<br/>N3 IO_L21N_T3_DQS_34</p> <p>P1 IO_L22P_T3_34<br/>N1 IO_L22N_T3_34</p> <p>R4 IO_L23P_T3_34<br/>R3 IO_L23N_T3_34</p> <p>R2 IO_L24P_T3_34<br/>P2 IO_L24N_T3_34</p> | 3  | <p>SW1 R716<br/>SW2 R717<br/>SW3 R718<br/>SW4 R719<br/>SW5 R720<br/>SW6 R721<br/>SW7 R722<br/>SW8 R723</p> <p>S700</p> <p>SW-DIP8</p> <p>GND</p> | 4  | <p><b>Dip Switches</b></p>   |
| B | <p>LED1<br/>LED2<br/>LED3<br/>LED4<br/>LED5<br/>LED6<br/>LED7<br/>LED8</p> <p>R724 470R<br/>R725 470R<br/>R726 470R<br/>R727 470R<br/>R728 470R<br/>R729 470R<br/>R730 470R<br/>R731 470R</p> <p>D702<br/>D703<br/>D704<br/>D705<br/>D706<br/>D707<br/>D708<br/>D709</p> <p>GND GND GND GND GND GND GND GND</p>  | <p>SW8</p>  | <p><b>Indication LEDs</b></p>                                      | <p>K1<br/>K2<br/>L1<br/>L2<br/>M1<br/>M2<br/>M3<br/>N1<br/>N2<br/>N3<br/>P1<br/>P2<br/>P3<br/>P4</p>  | <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p> | <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p>   | <p><b>Testpoints</b></p>   | <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p> |
| C | <p>P700</p> <p>Header 10</p> <p>+3V3</p> <p>R700 500R<br/>R701 500R<br/>R702 500R<br/>R703 500R<br/>R704 500R<br/>R705 500R<br/>R706 500R<br/>R707 500R</p> <p>D711<br/>D710<br/>D709<br/>D708<br/>D707<br/>D706<br/>D705<br/>D704<br/>D703<br/>D702<br/>D701<br/>D700</p> <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p> <p>GND GND GND GND GND GND GND GND</p> | <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p>  | <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p> | <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p>  | <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p> | <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p>   | <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p> | <p>TP1<br/>TP2<br/>TP3<br/>TP4<br/>TP5<br/>TP6<br/>TP7<br/>TP8</p> |
| D | <p>IO_0_34<br/>IO_25_34</p>  | <p>IO_0_34<br/>IO_25_34</p>   | <p>IO_0_34<br/>IO_25_34</p>  | <p>IO_0_34<br/>IO_25_34</p>   | <p>IO_0_34<br/>IO_25_34</p>  | <p>IO_0_34<br/>IO_25_34</p>  | <p>IO_0_34<br/>IO_25_34</p>  | <p>IO_0_34<br/>IO_25_34</p>  |

Title: **FPGA BANK 34 GPIO**

Size: A4

Date: 4/04/2024

File: C:\Users\... \FPGA\_BANK34\_SchDoc

Revision: **REV 1.0**

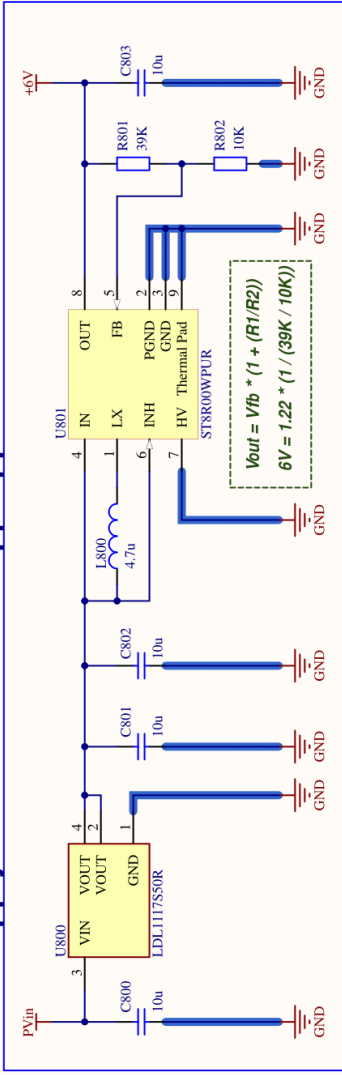
Sheet of: 4

Drawn By: **Louis Van Lommel**

OCG ELECTRONICS  
Dijkberd 34 building 19B  
2200 Herentals - Belgium  
www.ocelectronics.be



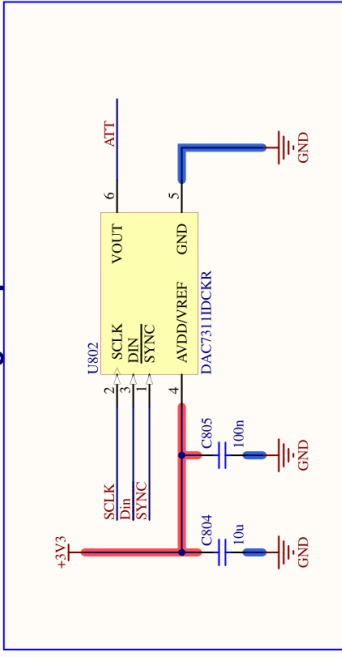
## Power Supply IVQ-3005: max 10Vpp ripple



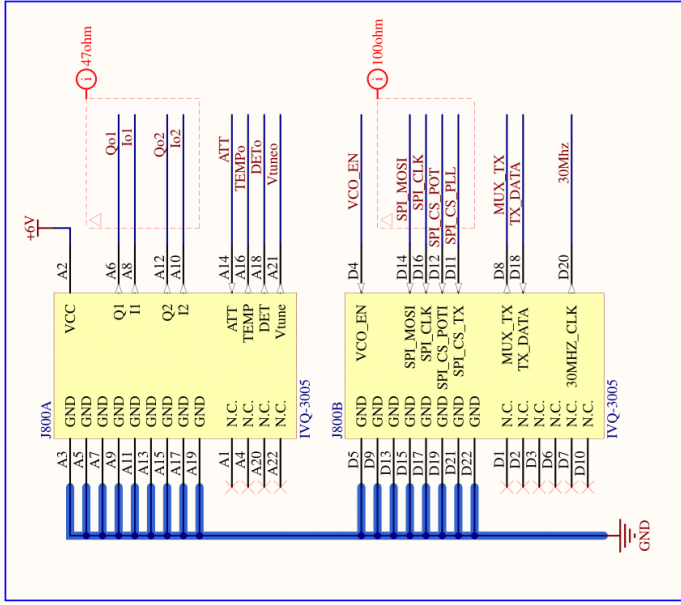
$$V_{out} = V_{fb} * (1 + (R1/R2))$$

$$6V = 1.22 * (1 + (39K / 10K))$$

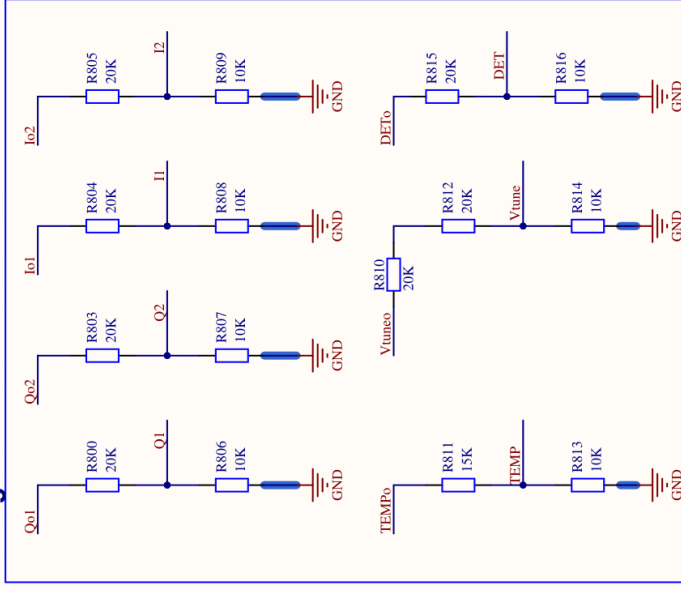
## DAC for controlling output attenuation



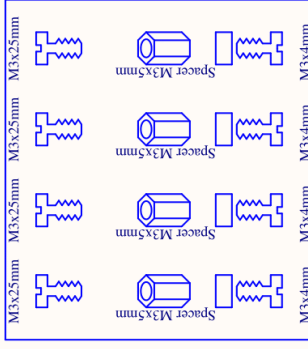
## InnoSent IVQ-3005 Connector



## Voltage dividers: alles naar 0-1V



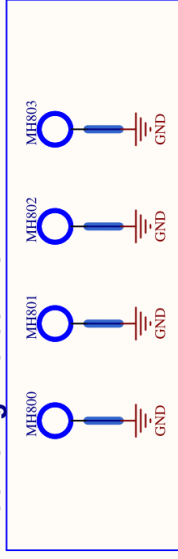
## Mechanical:



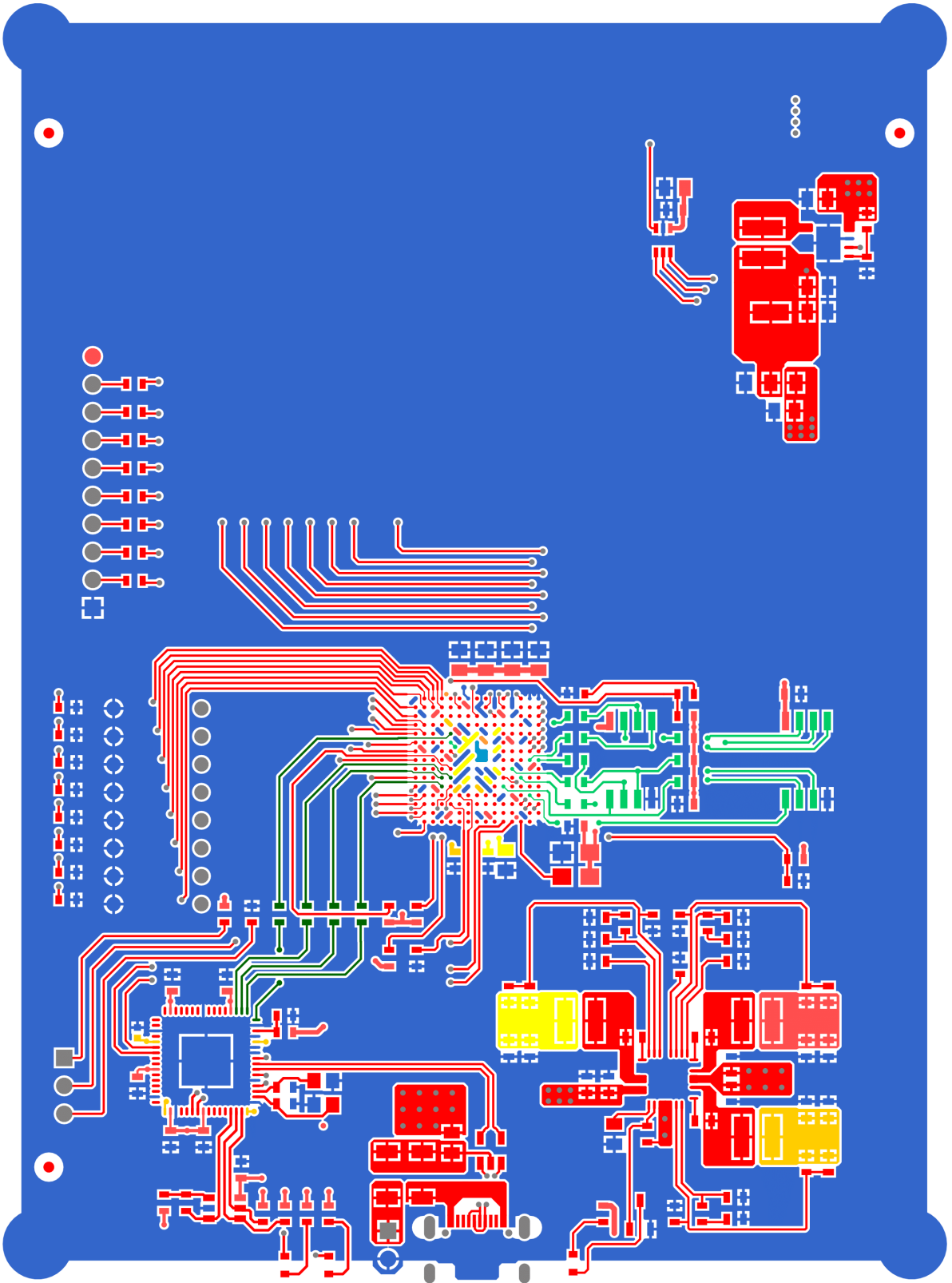
## Specifications:

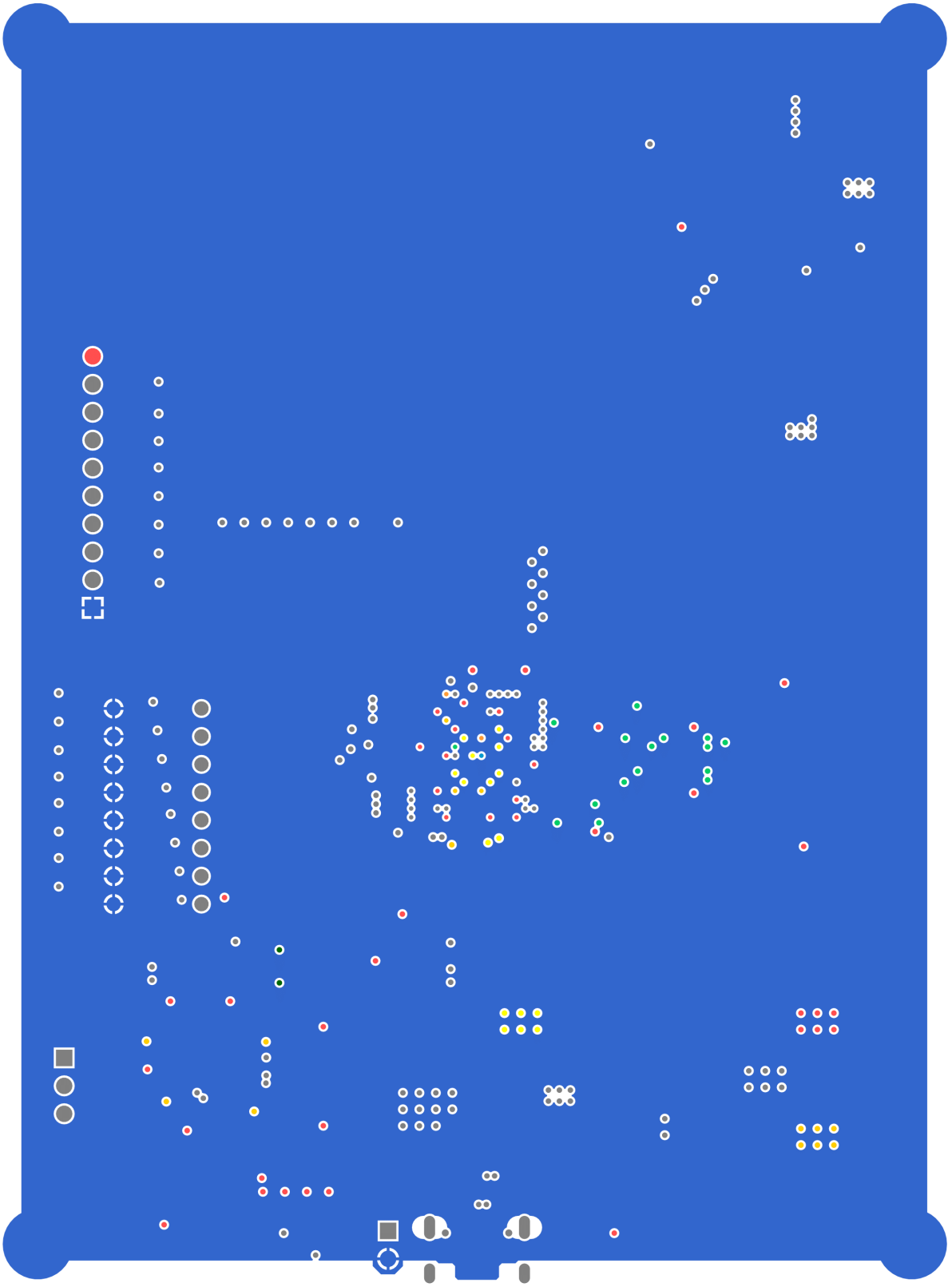
- Tx Freq: 24-24.25GHz
- Tx Pout: 8-30dBm
- IF BW: 25Hz-61kHz
- IF DC-Offset: 1.5V
- IF Out: 0-3V

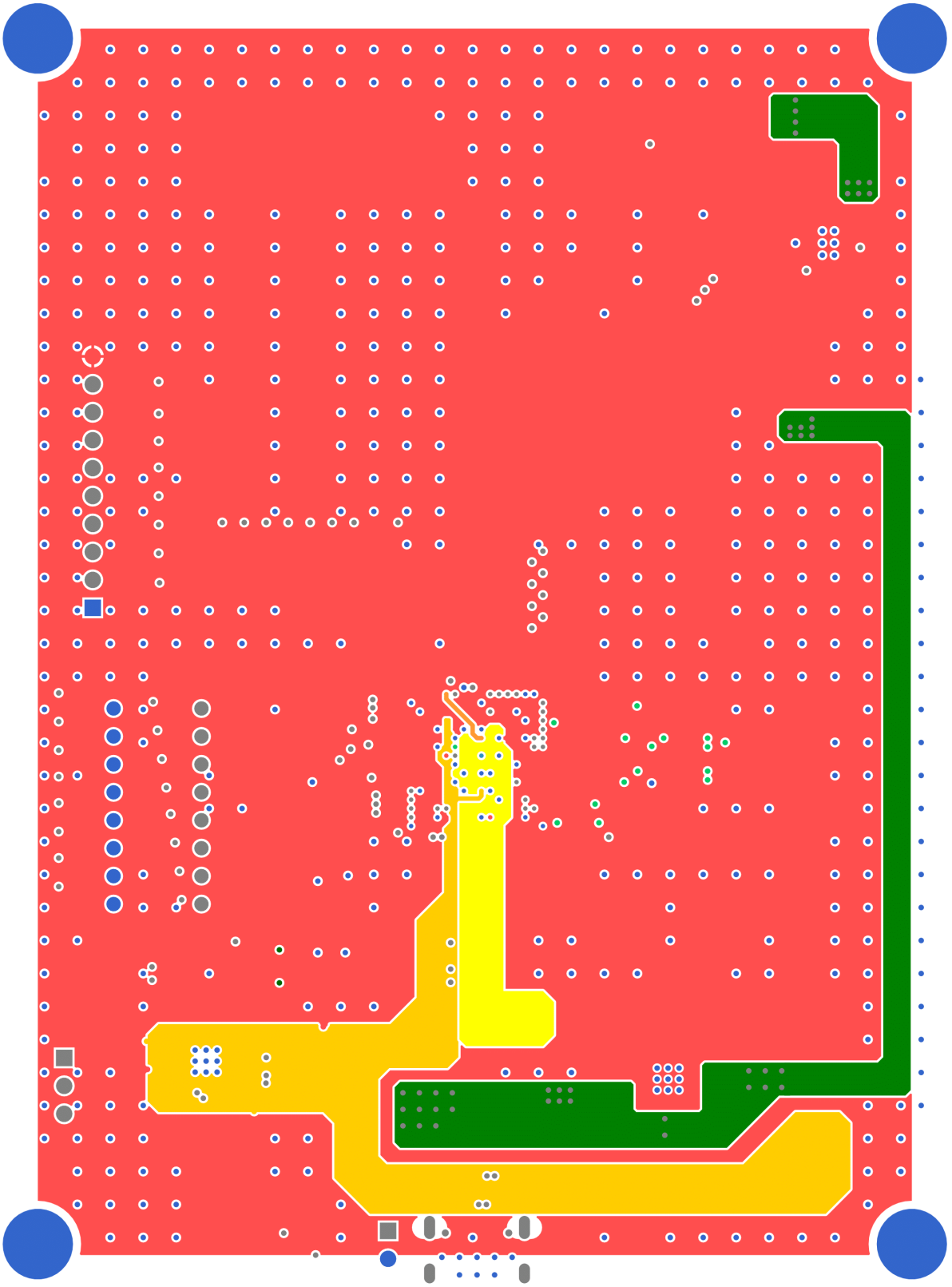
## Mounting Holes M3

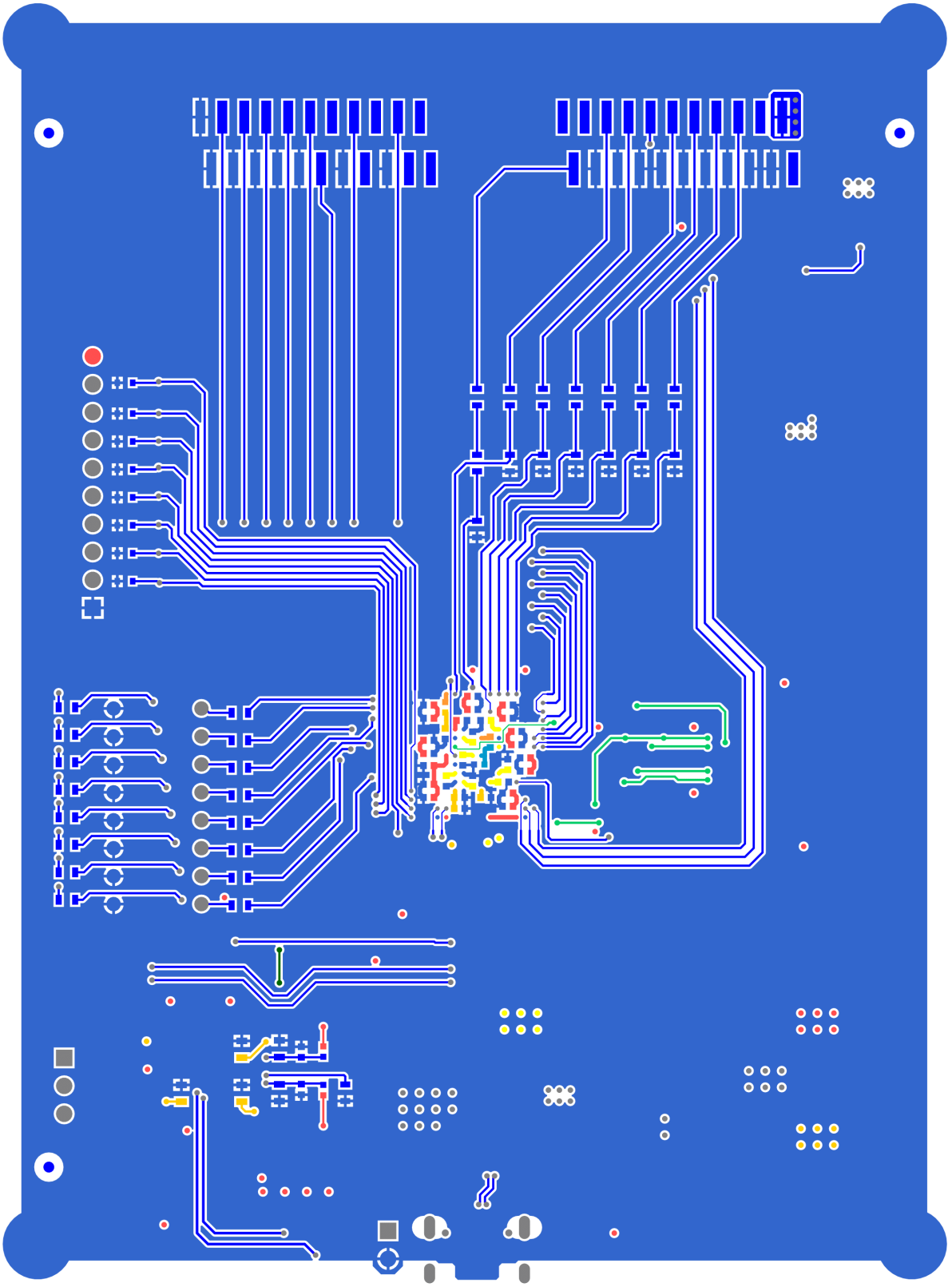


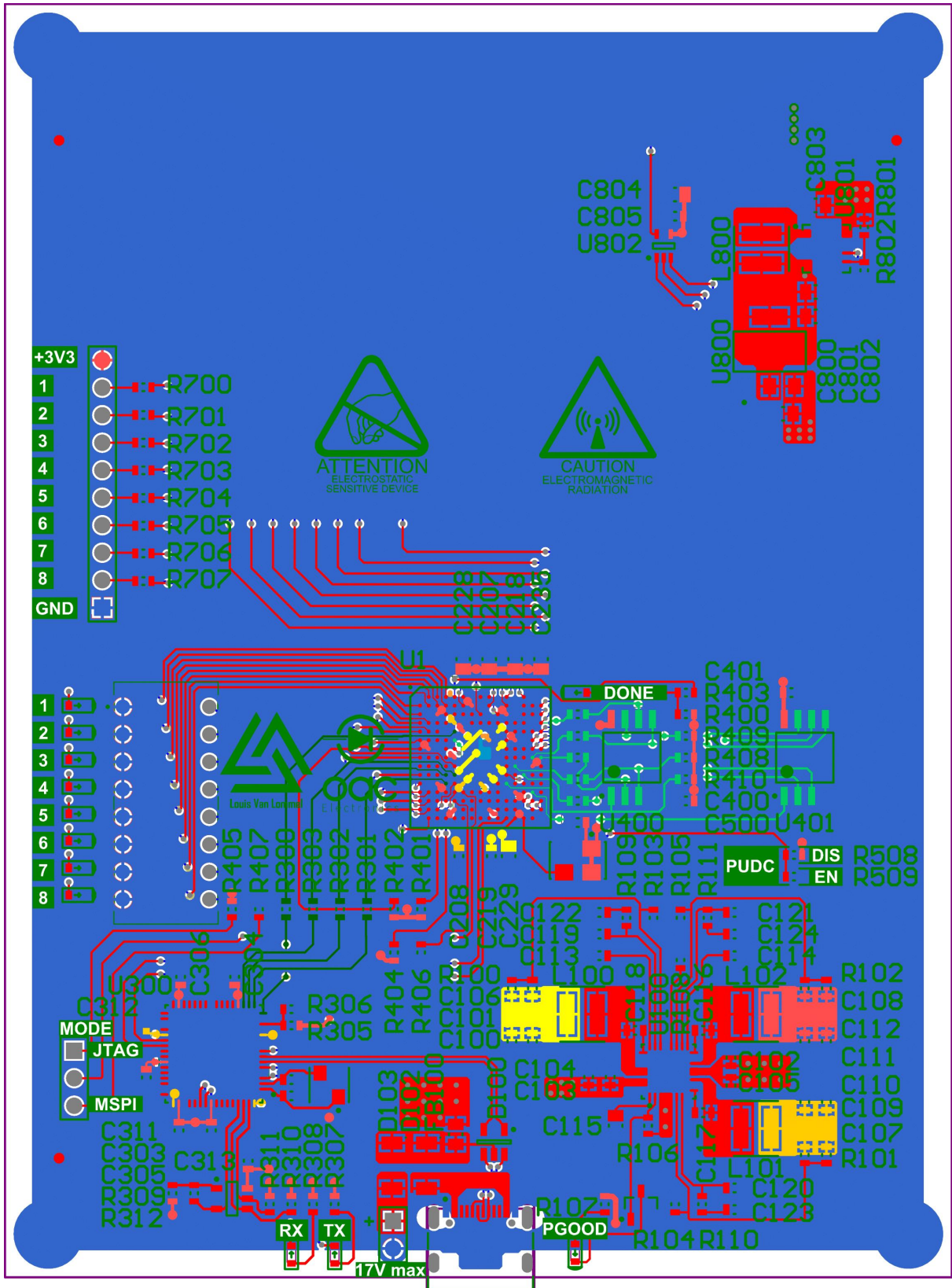
|       |                                |  |                  |
|-------|--------------------------------|--|------------------|
| Title |                                | Radar Module Connectivity + Mechanical   |                  |
| Size  | A4                             | Revision   | REV 1.0          |
| Date: | 4/04/2024                      | Sheet of   | 1                |
| File: | C:\Users\A.RADAR_MODULE\SchDoc | Drawn By:  | Louis Van Lommel |
|       |                                | Dikbeerd 34 building 13B<br>2200 Herentals - Belgium<br><a href="http://www.ocellectronics.be">www.ocellectronics.be</a> |                  |

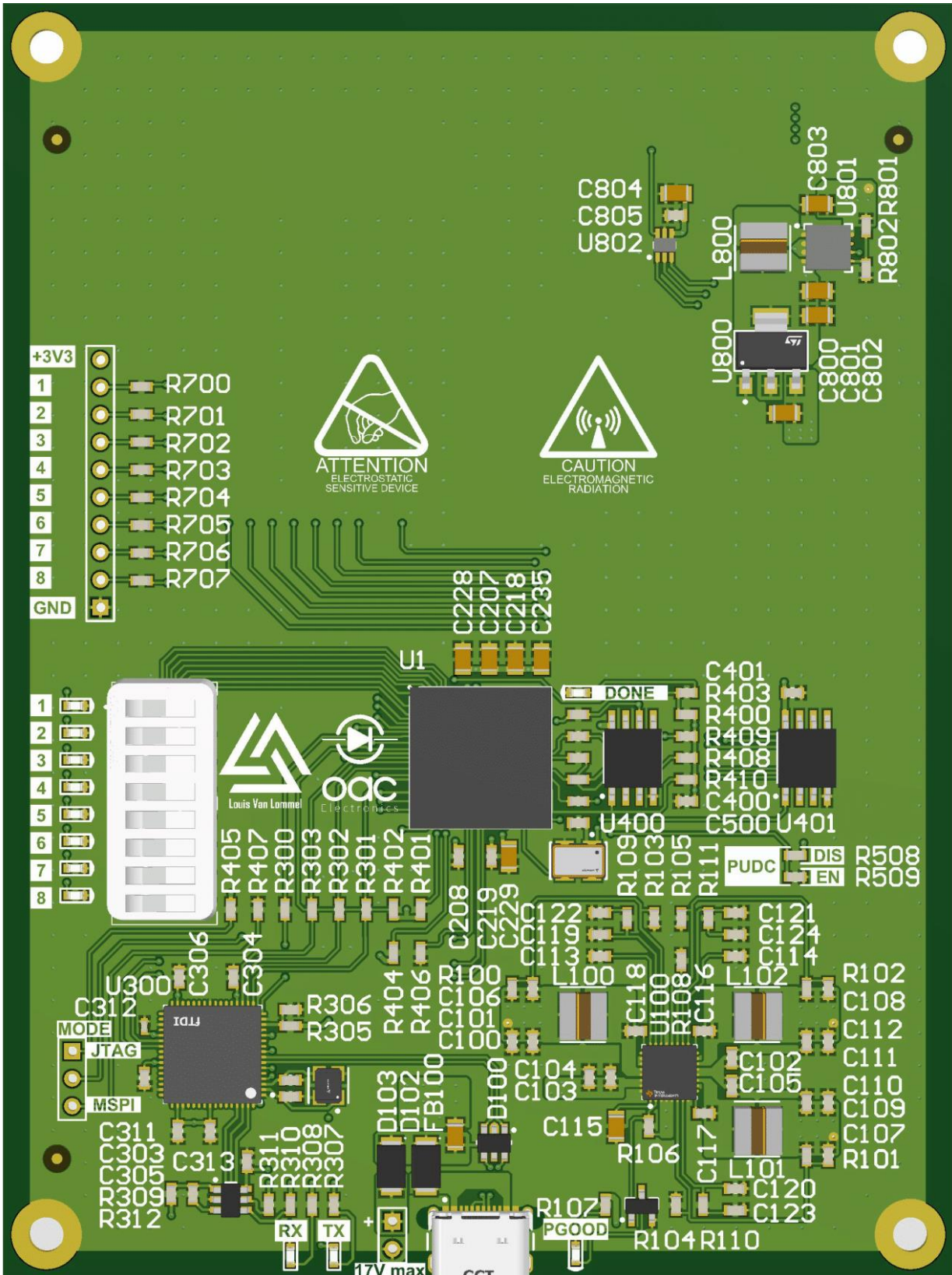


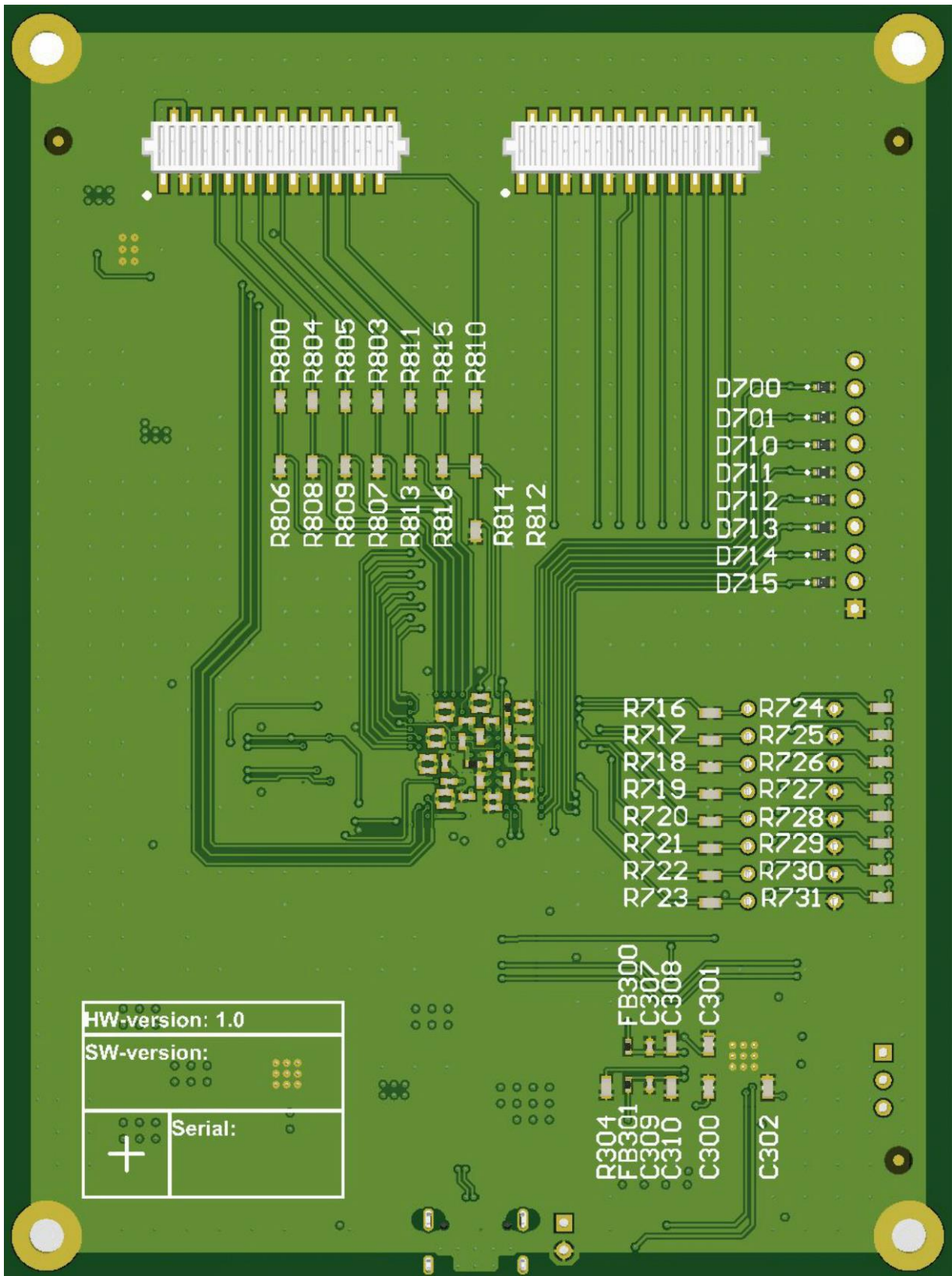








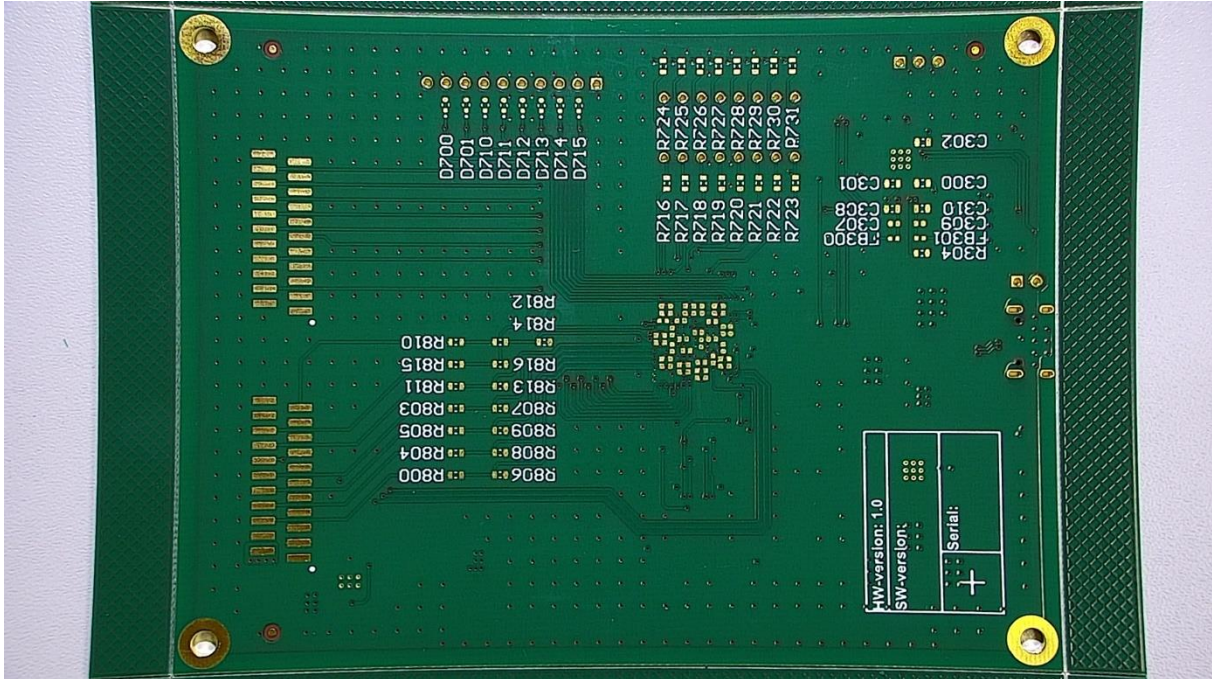
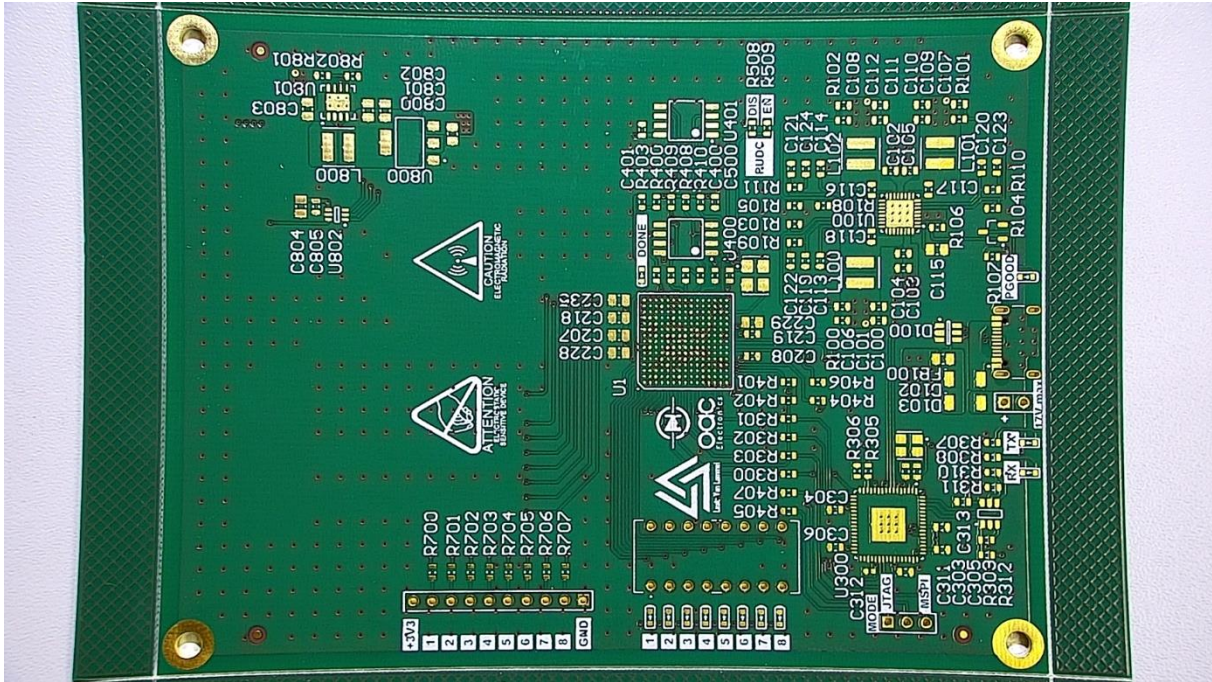


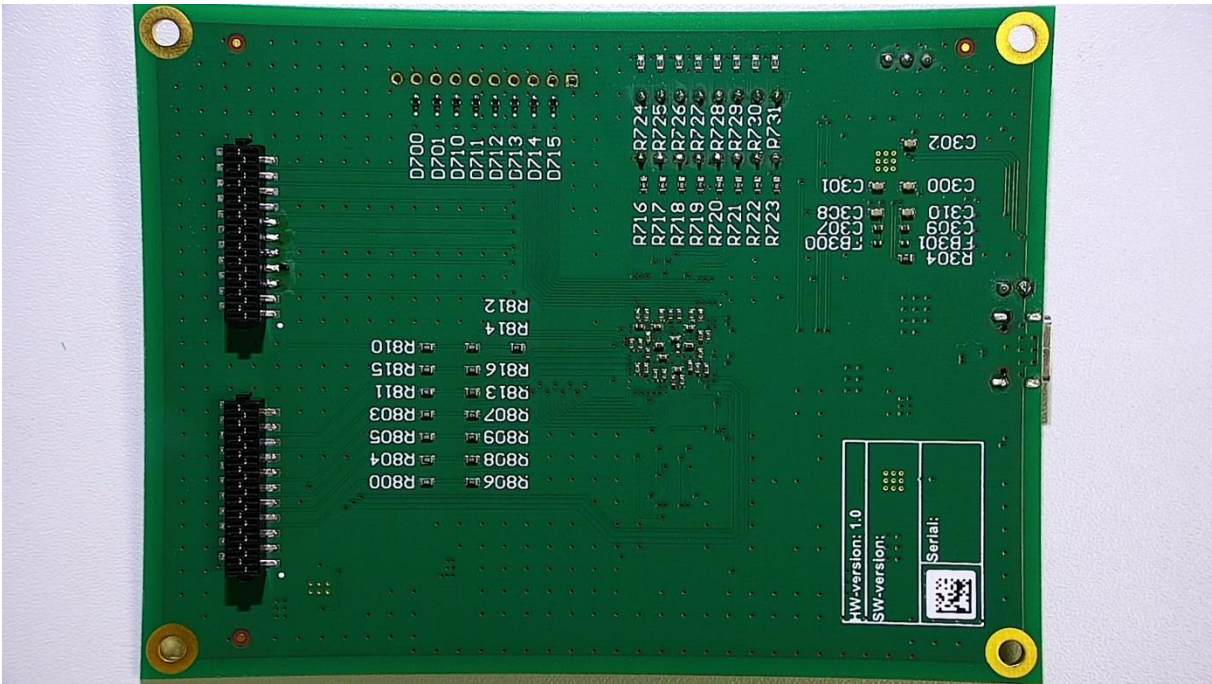
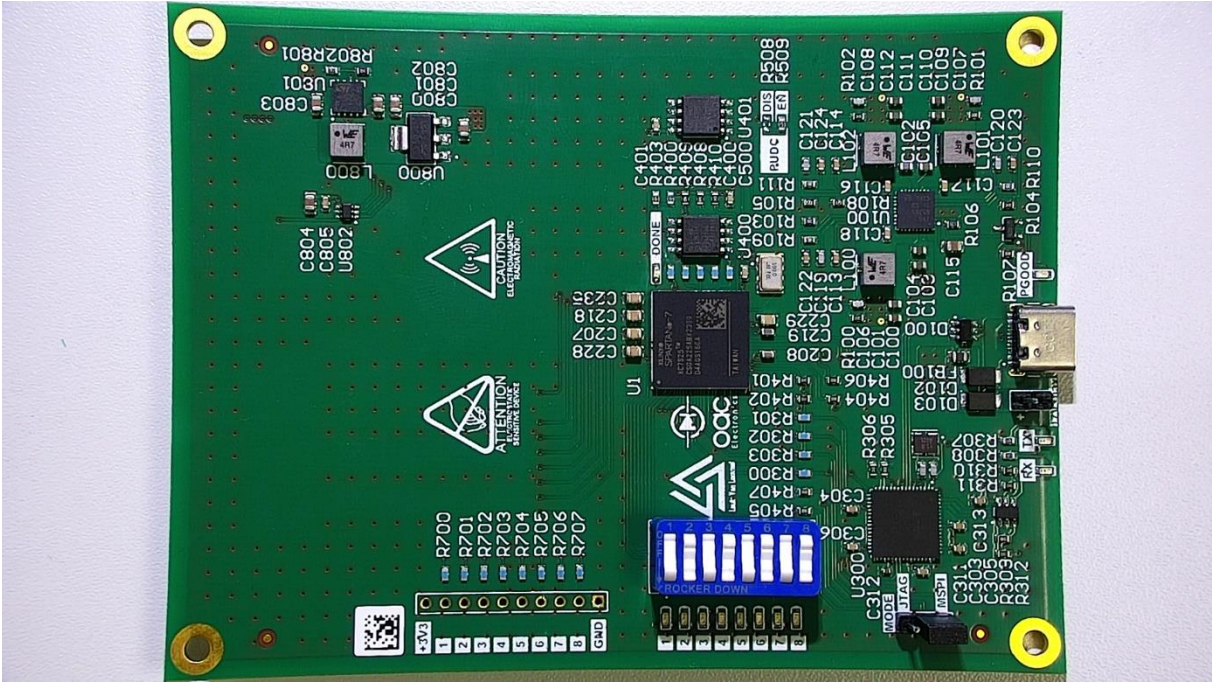




| Name                    | Description  | Designator   | Footprint                       | LibRef               | Quantity | BFP     |
|-------------------------|--|--|---------------------------------|----------------------|----------|---------|
| DAC311DCCR              | 12-Bit, Single Channel, 80µA, 1.8 to 5.5 V DAC in SOT1 Package -40 to 125 deg.C, 6-pin SOT23 (DCK6), Green (PbHS-A no Sn Br)   | U802   | T1-DCK6_V                       | CMP-0004-00559-1     | 1        | U000096 |
| SW-DIP8                 | 4009 Series DIP Switch, Raised actuator  | S700   | 8-DIPSWITCH                     | SW-DIP8              | 1        | 0.00268 |
| 0.47u                   | Capacitor  | C200, C201, C202, C203, C204, C210, C211, C212, C213, C214, C215, C221, C222, C230, C234, C235, C233, C234, C236   | CAP_0402                        | Cap                  | 19       | C000005 |
| 2.7n                    | Capacitor  | C113, C114, C122, C123, C124   | CAP_0603                        | Cap                  | 5        |         |
| 4.7u                    | Capacitor  | C200, C206, C209, C216, C217, C220, C226, C227, C230, C231, C232, C237, C209, C312   | CAP_0402                        | Cap                  | 14       | C000003 |
| 10u                     | Capacitor  | C115, C800, C801, C902, C803, C904   | CAP_0805                        | Cap                  | 6        | C000067 |
| 22p                     | Capacitor  | C105, C107, C108, C119, C120, C121, C114, C315   | CAP_0603                        | Cap                  | 8        | C000021 |
| 22u                     | Capacitor  | C100, C101, C102, C103, C104, C105, C109, C110, C111, C112   | CAP_0603                        | Cap                  | 10       | C000091 |
| 47n                     | Capacitor  | C116, C117, C118   | CAP_0603                        | Cap                  | 3        | C000008 |
| 47u                     | Capacitor  | C208, C219   | CAP_0603                        | Cap                  | 2        | C000060 |
| 100n                    | Capacitor  | C300, C301, C302, C303, C304, C305, C306, C308, C310, C311, C313, C400, C401, C308, C305   | CAP_0603                        | Cap                  | 15       | C000137 |
| 100u                    | Capacitor  | C207, C218, C228, C229, C235   | CAP_0805                        | Cap                  | 5        | C000255 |
| Ferrite Bead            | Chip Ferrite Bead, 1000Q@100MHz, 0.65Q, 25%, 5A  | FB100  | CAP_0805                        | Ferrite Bead         | 1        | L000059 |
| Ferrite Bead            | Chip Ferrite Bead, 1000Q@100MHz, 0.65Q, 25%, 300mA   | FB200, FB201, FB300, FB301   | RES_0402                        | Ferrite Bead         | 4        | L000134 |
| USB4105-GF-A            | COMBI-FUNCTIONAL TYPE-C  | J100   | FP-USB4105-GF-A-MFG             | CMP-245163-000001-1  | 1        |         |
| ABM83-12.000MHZ-4Pin-T3 | Crystal 12MHz ±30ppm (Typ) ±30ppm (Stability) 10pF FUND 12004-Pin SMD T/R  | Y300   | FP-ABM83-MFG                    | CMP-27762-014577-1   | 1        |         |
| PHES2630EP              | DIODE SCHOTTKY 60V 3A SMD 1/8  | D102, D103   | SCD128                          | D Schottky           | 2        | D000277 |
| MACRONIX M25V1635FMI    | Flash Memory, Serial NOR, 16 Mbit, 2M x 8bit, SPI, SOP, 8 Pins   | U400, U401   | SOP-8                           | MACRONIX M25V1635FMI | 2        |         |
| 82K                     | Resistor   | R108   | RES_0603                        | Res2                 | 1        | R000547 |
| Header 2                | Header, 2-Pin  | P100   | HDRX2                           | Header 2             | 1        | DNP     |
| Header 3                | Header, 3-Pin  | P400   | HDRX3                           | Header 3             | 1        | DNP     |
| Header 10               | Header, 10-Pin   | P700   | HDRX10                          | Header 10            | 1        | DNP     |
| TP862069FHB1            | IC REG BUCK ADJ TRPL 3V OVFN   | U100   | FP-TP862069FHB1-IPC             | CMP-04918-000081-1   | 1        |         |
| LDL1175S0R              | IC REG LINEAR SV1.2A SOT23   | U800   | FP-SOT23-094867-14-MFG          | CMP-12153-000005-1   | 1        |         |
| FT2232HQ-Rev1           | IC USB HS DUAL UART FPC64 QFN  | U300   | FP-FT2232HQ-Rev1-MFG            | CMP-0078-00004-5     | 1        |         |
| IV3-3005                | Inductor   | L100, L101, L102, L800   | IND4x6mm                        | Inductor             | 4        | L000008 |
| IV3-3005                | IncoSert IV3-3005  | J800   | IV3-3005                        | IV3-3005             | 1        | DNP     |
| STRF00MPLR              | Micropower step up DC/DC converter, 8-Pin DFN, Tape and Reel   | U801   | DFN(4x4)_N                      | CMP-0240-00155-1     | 1        |         |
| MC9FETN                 | N-Channel MOSFET   | Q100   | SOT23127P800-8N                 | MC9FETN              | 1        | Q000067 |
| 1K                      | Resistor   | R107, R305, R312, R404, R405, R406, R407   | RES_0603                        | Res2                 | 7        | R000009 |
| 2K2                     | Resistor   | R105   | RES_0603                        | Res2                 | 1        | R000032 |
| 68Ω                     | Resistor   | R109, R110, R111   | RES_0603                        | Res2                 | 3        | R000216 |
| 10K                     | Resistor   | R100, R102, R104, R304, R306, R309, R310, R311, R401, R402, R408, R409, R410, R216, R217, R218, R219, R220, R221, R222, R223, R802, R806, R807, R808, R809, R813, R814, R816 | RES_0603                        | Res2                 | 29       | R000180 |
| 15K                     | Resistor   | R103, R811   | RES_0603                        | Res2                 | 2        | R000013 |
| 20K                     | Resistor   | R101, R800, R803, R804, R805, R810, R812, R815   | RES_0603                        | Res2                 | 8        | R000031 |
| 39K                     | Resistor   | R601   | RES_0603                        | Res2                 | 1        | R000222 |
| 50R                     | Resistor   | R200, R201, R202, R203, R411, R412, R413, R414, R415, R416, R417, R418, R419, R420, R421, R422, R423, R424, R425, R426, R427, R428, R429, R430, R431                         | RES_0603                        | Res2                 | 17       | R000030 |
| 82K                     | Resistor   | R106   | RES_0603                        | Res2                 | 1        | R000547 |
| 470R                    | Resistor   | R207, R208, R400, R403, R406, R409, R424, R425, R428, R427, R428, R429, R430, R431   | RES_0603                        | Res2                 | 14       | R000178 |
| M3x25mm                 | Screw M3x25mm cross recessed head screw<br><a href="https://be.farnell.com/tr-fastenings/m3cs-ssd325d0-screw-socket-cap-1-4-20-m3x25-dp/1419949">https://be.farnell.com/tr-fastenings/m3cs-ssd325d0-screw-socket-cap-1-4-20-m3x25-dp/1419949</a> | M800, M801, M802, M803   |                                 | SCHRAUBE             | 4        | DNP     |
| M3x6mm                  | Screw M3x6mm Nut   | M808, M809, M810, M811   |                                 | SCHRAUBE             | 4        | DNP     |
| 93LC56B-SN              | SERIAL EEPROM, 2KBIT, 2MHZ, SOT-23-6   | U301   | SOT23-6                         | MC93C56B-SN          | 1        | U000470 |
| Spacer M3x6mm           | Spacer 5mm M3<br><a href="https://be.farnell.com/harwin/gd-1010502-standoff-hex-h-brass-m3-5mm-dp-075450175-m5x20-spacer">https://be.farnell.com/harwin/gd-1010502-standoff-hex-h-brass-m3-5mm-dp-075450175-m5x20-spacer</a>                     | M804, M805, M806, M807   |                                 | Distanzboizen        | 4        | DNP     |
| PE30V3L1UE115           | TVS DIODE 3.3VMM 11V0 SOD323   | D700, D701, D710, D711, D712, D713, D714, D715   | FP-SOD323-MFG                   | CMP-14432-000009-1   | 8        | D000267 |
| LED                     | Typical LED Blue   | D702, D703, D704, D705, D706, D707, D708, D709   | LED_0603                        | LED0                 | 8        | D000151 |
| LED                     | Typical LED GREEN  | D101   | LED_0603                        | LED0                 | 1        | D000170 |
| LED                     | Typical LED PURPLE   | D300, D301, D400   | LED_0603                        | LED0                 | 3        | D000158 |
| USBL02-25D6             | Very low capacitance ESD protection, 3.5 pF -40 to 125 deg.C, 6-Pin SOT23, PbHS, Tape and Reel   | D100   | STM-SOT23-4L_V                  | CMP-2000-05378-1     | 1        | U000280 |
| KC735-1C3Q425I          | KC735-1C3Q425I   | U1   | BGA225C80PH 5X15_1 300X1300X140 | KC735-1C3Q425I       | 1        |         |
| ASFL1-100.000MHZ-EC-T   | XTAL Oscillator 100MHz ±50ppm HCMOS, TTL, 3.3V, 4-SMD 5mm x 3.2mm  | Y500   | FP-ASFL1-MFG                    | CMP-27762-011904-1   | 1        |         |

**BIJLAGEN PRODUCTIE:**





# 13.3. BIJLAGEN VIVADO:

## TESTBENCH FFT:



## 13.4. BIJLAGEN VITIS:

### XADC TESTCODE:

```
typedef struct XadcInfo {
    int NumChannels;
    int Channels[2];
    int ChannelIsDiff[2];
    char *ChannelNames[2];
} XadcInfo;

typedef struct Xadc_ChannelInfo {
    int SequencerIndex;
    int IsDiff;
    char *Name;
} Xadc_ChannelInfo;

void Xadc_Init(XSysMon *InstancePtr, const Xadc_ChannelInfo DeviceInfo[], const int
NumChannels, const u32 DeviceId);
void Xadc_ReadData(XSysMon *InstancePtr, const Xadc_ChannelInfo DeviceInfo[], const
int NumChannels, u16 RawData[]);

//Initialize the XADC Wizard IP
void Xadc_Init(XSysMon *InstancePtr, const Xadc_ChannelInfo DeviceInfo[], const int
NumChannels, const u32 DeviceId) {
    u32 ChannelIsMask;
    u32 InputModeMask;

    XSysMon_Config *xadc_cfg_ptr;
    xadc_cfg_ptr = XSysMon_LookupConfig(DeviceId);
    XSysMon_CfgInitialize(InstancePtr, xadc_cfg_ptr, xadc_cfg_ptr->BaseAddress);

    ChannelIsMask = 0;
    InputModeMask = 0;

    for(int i = 0; i < NumChannels; i++){
        ChannelIsMask |= 1 << DeviceInfo[i].SequencerIndex;
        if (DeviceInfo[i].IsDiff)
            InputModeMask |= 1 << DeviceInfo[i].SequencerIndex;
    }

    //Disable the channel sequencer before configuring the sequence registers
    XSysMon_SetSequencerMode(InstancePtr, XSM_SEQ_MODE_SAFE);
    //Leave alarm enables default
    //Leave channel averaging default
    //Set single-ended input mode for all channels
    XSysMon_SetSeqInputMode(InstancePtr, XSM_SEQ_MODE_SAFE);
    //Leave acquisition time default
    XSysMon_SetSeqAcqTime(InstancePtr, ChannelIsMask);
    //Enable all channels
    XSysMon_SetSeqChEnables(InstancePtr, ChannelIsMask);
    //Leave the ADCLK frequency default
    //Leave calibration default
    //Enable the channel sequencer in continuous sequencer cycling mode
    XSysMon_SetSequencerMode(InstancePtr, XSM_SEQ_MODE_CONTINPASS);
}

//Captures Data from each XADC Channel
```

```

void Xadc_ReadData(XSysMon *InstancePtr, const Xadc_ChannelInfo DeviceInfo[], const
int NumChannels, u16 RawData[]){
    //Clear the status
    XSysMon_GetStatus(InstancePtr);

    //Wait until the End of Sequence occurs
    while ((XSysMon_GetStatus(InstancePtr) & XSM_SR_EOS_MASK) != XSM_SR_EOS_MASK);
    //Capture data from each channel, return via RawData
    for (int i = 0; i < NumChannels; i++){
        RawData[i] = XSysMon_GetAdcData(InstancePtr, DeviceInfo[i].SequencerIndex);
    }

    return;
}

int main()
{
    init_platform();

    //initialize ADC channels
    const int NumChannels = 7;
    //Zie UG480: Sequencer Modes
    const Xadc_ChannelInfo XadcInfo[7] = {
        {17, 0, "DET"}, //VAUX_1, DETection
        {19, 0, "TEMP"}, //VAUX_3, TEMPerature
        {20, 0, "I2"}, //VAUX_4, I2
        {21, 0, "Q1"}, //VAUX_5, Q1
        {25, 0, "Vtune"}, //VAUX_9, Vtune
        {27, 0, "Q2"}, //VAUX_11, Q2
        {28, 0, "I1"} //VAUX_12, I1
    };

    //initialize ADC
    XSysMon Xadc_Ctrl;
    u16 Xadc_RawData[NumChannels];
    Xadc_Init(&Xadc_Ctrl, XadcInfo, NumChannels, XPAR_XADC_WIZ_0_DEVICE_ID);
    u16 Xadc_Voltage;

    while(1){

        //Read ADC Values
        Xadc_ReadData(&Xadc_Ctrl, XadcInfo, NumChannels, Xadc_RawData);
        for (int i = 0; i < NumChannels; i++){
            Xadc_Voltage = Xadc_RawData[i];
            xil_printf("%s: %d\n", XadcInfo[i].Name, Xadc_Voltage);
        }
    }
    cleanup_platform();
    return 0;
}

```

## XFFT TESTCODE:

De input samples worden opgeslagen op een apart stuk BRAM (zie AXI BRAM controller, figuur 82.):

```
int addr = XPAR_BRAM_0_BASEADDR;
for(int i=0; i < 2*N; i++){
    XIo_Out32(addr, FFT_Input[i]);
    addr = addr + 4;
}
```

Om de input samples naar de FFT te sturen is de DMA nodig, deze gaat de de input samples naar de FFT sturen en de FFT output verder opslaan in het apart stuk BRAM waar de input samples zijn opgeslagen:

```
int statuscheck = XAxiDma_SimpleTransfer(&my_dma, XPAR_BRAM_0_BASEADDR,
sizeof(float)*N*2, XAXIDMA_DMA_TO_DEVICE);
if (statuscheck != XST_SUCCESS){
    xil_printf("XST_FAILURE\r\n");
}

statuscheck = XAxiDma_SimpleTransfer(&my_dma, XPAR_BRAM_0_BASEADDR +
sizeof(float)*N*2, sizeof(float)*N*2, XAXIDMA_DEVICE_TO_DMA);
if (statuscheck != XST_SUCCESS){
    xil_printf("XST_FAILURE\r\n");
}

statuscheck = checkIdle(XPAR_AXI_DMA_0_BASEADDR, 0x34); //PG021 p29
while (statuscheck != 2) {
    statuscheck = checkIdle(XPAR_AXI_DMA_0_BASEADDR, 0x34);
    xil_printf("0x34: %d\n", statuscheck);
}

statuscheck = checkIdle(XPAR_AXI_DMA_0_BASEADDR, 0x04); //PG021 p18
while (statuscheck != 2) {
    statuscheck = checkIdle(XPAR_AXI_DMA_0_BASEADDR, 0x04);
    xil_printf("0x4: %d\n", statuscheck);
}
```

Zodra de input samples door de FFT verwerkt zijn wordt de uitgang als volgt gelezen:

```
addr = XPAR_BRAM_0_BASEADDR + sizeof(float)*N*2;
xil_printf("FFT OUTPUT:REAL+IMAG\r\n");
for(int i = 0; i < N*2; i++){
    fft[i] = XIo_In32(addr);
    addr = addr + 4;
    xil_printf("%1u\n", (u32)fft[i]);
    if (fft[i] > max){
        freq = i;
        max = fft[i];
    }
}
xil_printf("\nfftbin:%d, RES:%d, FREQ:%d\r\n", ((N*2-freq)/2), (120000/N),
((120000/N)*((N*2-freq)/2)));
}
```

## CW-RADAR TESTCODE:

```
#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"
#include "xsysmon.h"
#include "xparameters.h"
#include "xgpio.h"
#include "xaxidma.h"
#include "xio.h"
#include "xintc.h"
#include "xtmrctr.h"
#include "xil_exception.h"
#include "math.h"
#include "sleep.h"
/*
 * The following variables are shared between non-interrupt processing and
 * interrupt processing such that they must be global.
 */
#define N 128 //amount of samples
#define SampleRate 20000
#define RESET_VALUE (0xFFFFFFFF-(XPAR_CPU_CORE_CLOCK_FREQ_HZ/SampleRate)) //
sample rate ADC timer
#define GRAPH 0 //enables(1) the graph print function, make sure to delete all
other prints.
int flag = 0;
volatile int TimerExpired = 0;
u16 I[N];
u16 Q[N];

XSysMon Xadc_Ctrl;
XTmrCtr TimerCtr;
XIntc Interrupt;
XAXiDma my_dma;
XAXiDma_Config *my_dma_config;

typedef struct XadcInfo {
    int NumChannels;
    int Channels[2];
    int ChannelIsDiff[2];
    char *ChannelNames[2];
} XadcInfo;

typedef struct Xadc_ChannelInfo {
    int SequencerIndex;
    int IsDiff;
    char *Name;
} Xadc_ChannelInfo;

void Xadc_Init(XSysMon *InstancePtr, const Xadc_ChannelInfo DeviceInfo[], const
int NumChannels, const u32 DeviceId);
void Xadc_ReadData(XSysMon *InstancePtr, const Xadc_ChannelInfo DeviceInfo[],
const int NumChannels, u16 RawData[]);
void INIT_PLL(XGpio *InstancePtr, int data);
void serialwrite(XGpio *InstancePtr, int data);
void spiwrite(XGpio *InstancePtr, int data);
void timer_counter_handler(void* callbackref,u8 timer_counter_number);
```



```

//Initialize the XADC Wizard IP
void Xadc_Init(XSysMon *InstancePtr, const Xadc_ChannelInfo DeviceInfo[], const
int NumChannels, const u32 DeviceId) {
    u32 ChannelIsMask;
    u32 InputModeMask;

    XSysMon_Config *xadc_cfg_ptr;
    xadc_cfg_ptr = XSysMon_LookupConfig(DeviceId);
    XSysMon_CfgInitialize(InstancePtr, xadc_cfg_ptr, xadc_cfg_ptr->BaseAddress);

    ChannelIsMask = 0;
    InputModeMask = 0;

    for(int i = 0; i < NumChannels; i++){
        ChannelIsMask |= 1 << DeviceInfo[i].SequencerIndex;
        if (DeviceInfo[i].IsDiff)
            InputModeMask |= 1 << DeviceInfo[i].SequencerIndex;
    }

    //Disable the channel sequencer before configuring the sequence registers
    XSysMon_SetSequencerMode(InstancePtr, XSM_SEQ_MODE_SAFE);

    //Leave alarm enables default
    //Leave channel averaging default

    //Set single-ended input mode for all channels
    XSysMon_SetSeqInputMode(InstancePtr, XSM_SEQ_MODE_CONTINPASS);

    //Leave acquisition time default
    XSysMon_SetSeqAcqTime(InstancePtr, ChannelIsMask);

    //Enable all channels
    XSysMon_SetSeqChEnables(InstancePtr, ChannelIsMask);

    //Leave the ADCLK frequency default
    XSysMon_SetAdcClkDivisor(InstancePtr, 4);
    //Leave calibration default

    //Enable convst event based sampling
    XSysMon_SetSequencerEvent(InstancePtr, TRUE);

    //Enable the channel sequencer in continuous sequencer cycling mode
    XSysMon_SetSequencerMode(InstancePtr, XSM_SEQ_MODE_CONTINPASS);
}
//Initialize the DMA
void DmaInit(){
    my_dma_config = XAxiDma_LookupConfig(XPAR_AXI_DMA_0_DEVICE_ID);
    if (!my_dma_config) {
        xil_printf("No config found for dma\r\n");
    }
    int status = XAxiDma_CfgInitialize(&my_dma, my_dma_config);
    if (status != XST_SUCCESS) {
        xil_printf("DMA initialize failed\r\n");
    }
    if (XAxiDma_HasSg(&my_dma)) {
        xil_printf("Device configuration as SG mode\r\n");
    }
    // Reset DMA
}

```

```

XAxiDma_Reset(&my_dma);
while (!XAxiDma_ResetIsDone(&my_dma)) {
}
//xil_printf("DMA INITSUCCESS\n");
}
//initialize ADF4158
void INIT_PLL(XGpio *InstancePtr, int data){
    //Programming the PLL
    data = 0x00000007;
    spiwrite(InstancePtr, data);
    data = 0x00000006;
    spiwrite(InstancePtr, data);
    data = 0x00800006;
    spiwrite(InstancePtr, data);
    data = 0x00000005;
    spiwrite(InstancePtr, data);
    data = 0x00800005;
    spiwrite(InstancePtr, data);
    data = 0x00000004;
    spiwrite(InstancePtr, data);
    data = 0x00000003;
    spiwrite(InstancePtr, data);
    data = 0x00000002;
    spiwrite(InstancePtr, data);
    data = 0x00000001;
    spiwrite(InstancePtr, data);
    data = 0x00000000;
    spiwrite(InstancePtr, data);
    //FMCW mode
    data = 0x070007d2;//R-Divider register CLK1=250
    spiwrite(InstancePtr, data);
    data = 0x000C0004;//Test Register Enable Ramp Div mode
    spiwrite(InstancePtr, data);
    data = 0x00000003;//Function Register continious sawtooth
    spiwrite(InstancePtr, data);
    data = 0x80000000;//Start FMCW RAMP
    spiwrite(InstancePtr, data);

    //END OF CONVERSATION
    XGpio_DiscreteWrite(InstancePtr, 1, 0b111101111);
}
//write serial data to DAC7311 falling edge
void serialwrite(XGpio *InstancePtr, int data){

    usleep(100);
    XGpio_DiscreteWrite(InstancePtr, 1, 0b00000011); //PULL SYNC LOW
    //PD0/1 LOW normal operating conditions
    XGpio_DiscreteWrite(InstancePtr, 1, 0b00000010); //CLOCK HIGH
    usleep(10);
    XGpio_DiscreteWrite(InstancePtr, 1, 0b00000000); //CLOCK LOW
    usleep(10);
    XGpio_DiscreteWrite(InstancePtr, 1, 0b00000010); //CLOCK HIGH
    usleep(10);
    XGpio_DiscreteWrite(InstancePtr, 1, 0b00000000); //CLOCK LOW
    usleep(10);

    for (int i = 0; i < 12; i++){
        // Write data bit (MSB first)

```

```

    int bit = (data >> (11 - i)) & 0x01;
    XGpio_DiscreteWrite(InstancePtr, 1, 0b00000010 + bit); //CLOCK HIGH
    usleep(10);

    XGpio_DiscreteWrite(InstancePtr, 1, 0b00000000 + bit); //CLOCK LOW
    usleep(10);
}
//DB1/0 LOW
XGpio_DiscreteWrite(InstancePtr, 1, 0b00000010); //CLOCK HIGH
usleep(10);
XGpio_DiscreteWrite(InstancePtr, 1, 0b00000000); //CLOCK LOW
usleep(10);
XGpio_DiscreteWrite(InstancePtr, 1, 0b00000010); //CLOCK HIGH
usleep(10);
XGpio_DiscreteWrite(InstancePtr, 1, 0b00000000); //CLOCK LOW
usleep(10);

XGpio_DiscreteWrite(InstancePtr, 1, 0b00000111); //PULL SYNC HIGH
}
//Write SPI data to ADF4158 rising edge
void spiwrite(XGpio *InstancePtr, int data){
    //SPI_CS_PLL down
    XGpio_DiscreteWrite(InstancePtr, 1, 0b001001111);
    usleep(1);
    for (int i = 0; i < 32; i++){

        // Write data bit (MSB first)
        int bit = (data >> (31 - i)) & 0x07;
        XGpio_DiscreteWrite(InstancePtr, 1, 0b001001111 + bit); //CLOCK LOW
        usleep(1);
        XGpio_DiscreteWrite(InstancePtr, 1, 0b101001111 + bit); //CLOCK HIGH
        usleep(1);

    }
    usleep(1);
    //SPI_CS_PLL high
    XGpio_DiscreteWrite(InstancePtr, 1, 0b001101111);
    usleep(10);
}
//This function is defined to check whether DMA is busy or not
//Offset = 0x4 to check for DMA to DEVICE transaction channel
//Offset = 0x34 to check for DEVICE to DMA transaction channel
int checkIdle(u32 baseAddress, u32 offset){
    int status;
    status = (int)(XAxisDma_ReadReg(baseAddress, offset)) & XAXIDMA_IDLE_MASK;

    return status;
}
void timer_counter_handler(void* callbackref, u8 timer_counter_number){
    XTmrCtr *timer_ptr= (XTmrCtr *)callbackref;

    XSysMon_StartAdcConversion(&Xadc_Ctrl); //convst pulse high
    XSysMon_StartAdcConversion(&Xadc_Ctrl); //convst pulse high
    XSysMon_StartAdcConversion(&Xadc_Ctrl); //convst pulse high
    XSysMon_StartAdcConversion(&Xadc_Ctrl); //convst pulse high

```

```

I[TimerExpired] = XSysMon_GetAdcData(&Xadc_Ctrl, 21);
Q[TimerExpired] = XSysMon_GetAdcData(&Xadc_Ctrl, 28);

/*
 * Check if the timer counter has expired, checking is not necessary
 * since that's the reason this function is executed, this just shows
 * how the callback reference can be used as a pointer to the instance
 * of the timer counter that expired, increment a shared variable so
 * the main thread of execution can see the timer expired
 */
if (XTmrCtr_IsExpired(timer_ptr, timer_counter_number)) {
    TimerExpired++;
    if (TimerExpired >= (N)) {
        flag = 1;
        TimerExpired = 0;
        XTmrCtr_SetOptions(timer_ptr, timer_counter_number, 0);
    }
}
}
int main()
{
    init_platform();

    DmaInit();

    //initialize ADC channels
    const int NumChannels = 2;
    //Zie UG480: Sequencer Modes
    const Xadc_ChannelInfo XadcInfo[2] = {
        //{{17, 0, "DET"}, //VAUX_1, DETection
        //{{19, 0, "TEMP"}, //VAUX_3, TEMPerature
        //{{20, 0, "I2"}, //VAUX_4, I2
        {21, 0, "Q1"}, //VAUX_5, Q1
        //{{25, 0, "Vtune"}, //VAUX_9, Vtune
        //{{27, 0, "Q2"}, //VAUX_11, Q2
        {28, 0, "I1"} //VAUX_12, I1
    };

    //initialize ADC
    Xadc_Init(&Xadc_Ctrl, XadcInfo, NumChannels, XPAR_XADC_WIZ_0_DEVICE_ID);

    //pointers
    XGpio_Config *cfg_ptr;
    XGpio IVQ; //IVQ[0..8] 0DIN, 1SCLK, 2SYNC, 3VCO_en, 4TX_DATA, 5SPI_CS_PLL,
    6SPI_CS_POTI, 7SPI MOSI, 8SPI CLK

    //initialize DAC
    cfg_ptr = XGpio_LookupConfig(XPAR_SPI_IVQ_DEVICE_ID);
    XGpio_CfgInitialize(&IVQ, cfg_ptr, cfg_ptr->BaseAddress);
    XGpio_DiscreteWrite(&IVQ, 1, 0b111100111);
    //serial write DAC to set output power IVQ-3005, 12 bit value last 4 bits are 0
    u32 data = 4095; // Vout = vdd * data/4096, 8dBm (zie datasheet IVQ-3005
    serialwrite(&IVQ, data);

    //initialize VCO_EN HIGH
    XGpio_DiscreteWrite(&IVQ, 1, 0b111101111);

    //Initialize PLL

```

```

INIT_PLL(&IVQ, data);

    float fft[N*2];

    microblaze_disable_interrupts();
//INIT TIMER
int Status = XTmrCtr_Initialize(&TimerCtr, XPAR_TMRCTR_0_DEVICE_ID);
if (Status != XST_SUCCESS) {
    return XST_FAILURE;
}
Status = XTmrCtr_SelfTest(&TimerCtr, 0);
if (Status != XST_SUCCESS) {
    return XST_FAILURE;
}
//INIT INTC
Status = XIntc_Initialize(&Interrupt, XPAR_INTC_0_DEVICE_ID);
if (Status != XST_SUCCESS) {
    return XST_FAILURE;
}
Status = XIntc_Connect(&Interrupt, XPAR_INTC_0_TMRCTR_0_VEC_ID,
(XInterruptHandler)XTmrCtr_InterruptHandler, (void*)&TimerCtr);
if (Status != XST_SUCCESS) {
    return XST_FAILURE;
}
Status = XIntc_Start(&Interrupt, XIN_REAL_MODE);
if (Status != XST_SUCCESS) {
    return XST_FAILURE;
}
XIntc_Enable(&Interrupt, XPAR_INTC_0_TMRCTR_0_VEC_ID);

//XTmrCtr_SetHandler(&TimerCtr, (XTmrCtr_Handler)timer_counter_handler,
&TimerCtr);
//XTmrCtr_SetOptions(&TimerCtr, 0, XTC_INT_MODE_OPTION |
XTC_AUTO_RELOAD_OPTION);
//XTmrCtr_SetResetValue(&TimerCtr,0, RESET_VALUE);
microblaze_enable_interrupts();
//xil_printf("TIMER/INTC INITSUCCESS");
//XTmrCtr_Start(&TimerCtr, 0x00);

while(1){
    flag = 0;
    XTmrCtr_SetHandler(&TimerCtr, (XTmrCtr_Handler)timer_counter_handler,
&TimerCtr);
    XTmrCtr_SetOptions(&TimerCtr, 0, XTC_INT_MODE_OPTION |
XTC_AUTO_RELOAD_OPTION);
    XTmrCtr_SetResetValue(&TimerCtr,0, RESET_VALUE);
    XTmrCtr_SetOptions(&TimerCtr, 0, XTC_INT_MODE_OPTION |
XTC_AUTO_RELOAD_OPTION);
    XTmrCtr_Start(&TimerCtr, 0x00);
    while (flag == 0){
        //print("Getting ADC Data...\r\n");
    }
    u32 addr = XPAR_BRAM_0_BASEADDR;
    int freq = 0;
    float max = 0;

//flush memory
Xil_DCacheFlushRange(XPAR_BRAM_0_BASEADDR, sizeof(float)*N*2);

```

```

Xil_DCacheFlushRange(XPAR_BRAM_0_BASEADDR + sizeof(float)*N*2,
sizeof(float)*N*2);

u32 testI[N];
u32 testQ[N];
//STORE THE MEASURED DATA
for (int i = 0; i < N; i++){
    //I[i] = I[i] * (0.5-0.5*cos(2*M_PI*i/(N-1))); //Hanning window
    //Q[i] = Q[i] * (0.5-0.5*cos(2*M_PI*i/(N-1))); //Hanning window
    testI[i] = (u32)I[i];
    testQ[i] = (u32)Q[i];
    testI[i] = testI[i] << 9;
    testQ[i] = testQ[i] << 9;
    XIo_Out32(addr, (float)testI[i]); //I (REAL)
    addr = addr + 4; //addr + 4byte(32bit)
    XIo_Out32(addr, (float)testQ[i]); //Q (IMAG)
    addr = addr + 4;
}

// Disable interrupts, we use polling mode
XAxiDma_IntrDisable(&my_dma, XAXIDMA_IRQ_ALL_MASK, XAXIDMA_DEVICE_TO_DMA);
XAxiDma_IntrDisable(&my_dma, XAXIDMA_IRQ_ALL_MASK, XAXIDMA_DMA_TO_DEVICE);
//flush memory
Xil_DCacheFlushRange(XPAR_BRAM_0_BASEADDR, sizeof(float)*N*2);
Xil_DCacheFlushRange(XPAR_BRAM_0_BASEADDR + sizeof(float)*N*2,
sizeof(float)*N*2);
//DMA TRANSFER SAMPLES TO FFT
int statuscheck = XAxiDma_SimpleTransfer(&my_dma, XPAR_BRAM_0_BASEADDR,
sizeof(float)*N*2, XAXIDMA_DMA_TO_DEVICE);
if (statuscheck != XST_SUCCESS){
    //xil_printf("XST_FAILURE\r\n");
}
//DMA TRANSFER FFT OUTPUT
statuscheck = XAxiDma_SimpleTransfer(&my_dma, XPAR_BRAM_0_BASEADDR +
sizeof(float)*N*2, sizeof(float)*N*2, XAXIDMA_DEVICE_TO_DMA);
if (statuscheck != XST_SUCCESS){
    //xil_printf("XST_FAILURE\r\n");
}
//WAIT WHILE DMA IS OPERATING
statuscheck = checkIdle(XPAR_AXI_DMA_0_BASEADDR, 0x34); //PG021 p29
while (statuscheck != 2) {
    statuscheck = checkIdle(XPAR_AXI_DMA_0_BASEADDR, 0x34);
    //xil_printf("0x34: %d\n", statuscheck);
}

statuscheck = checkIdle(XPAR_AXI_DMA_0_BASEADDR, 0x04); //PG021 p18
while (statuscheck != 2) {
    statuscheck = checkIdle(XPAR_AXI_DMA_0_BASEADDR, 0x04);
    //xil_printf("0x4: %d\n", statuscheck);
}

//PRINT FFT OUTPUT
addr = XPAR_BRAM_0_BASEADDR + sizeof(float)*N*2;
char sign = '+';
//xil_printf("FFT OUTPUT:REAL+IMAG\r\n");
for(int i = 0; i < N*2; i++){
    fft[i] = (float)XIo_In32(addr);
    addr = addr + 4;
}

```

```

    if (fft[i] <= 0x80000000){
        fft[i] = 0;
    }
    //xil_printf("%1u\n", (u32)fft[i]);
    if (fft[i] > max){
        if (i < (N)){
            freq = (i-1);
            sign = '-';
        }else{
            freq = (N*2-(i-1));
            sign = '+';
        }
        max = fft[i];
    }
}
if (GRAPH == 1){
    xil_printf("[");
    for (int i = 0; i < N*2;i++){
        xil_printf("%1u", (u32)fft[i]);
        if ( i < N*2-1){
            xil_printf(",");
        }
    }
    xil_printf("]\r\n");
    usleep(50000);
}
int res = SampleRate/N;
float v = 0.0125*(freq*res/4); //golflengte*doppler freq/2
xil_printf("\nFFTbin:%d, RES:%d, FREQ:%d\r\nVelocity: %%1um/s =
%%1ukm/h\r\n", freq, res, res*freq, sign, (u32)v, sign, (u32)(v*3.6));
//xil_printf("%%1u\r\n", sign, (u32)(v*3.6));
usleep(100000);
if ((v*3.6) < 3){
    usleep(500000);
}
}
cleanup_platform();
return 0;
}

```

## SINEWAVE GENERATOR:

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

int main() {
    float pi = 3.14159265359;    //pi
    int N = 64;                //number of samples
    //sinewave 1
    float A1 = 0xffff;         //amplitude
    float f1 = 200000.0;       //frequentie
    float fs1 = 1000000.0;     //sample frequentie
    float phi1 = 0;           //phase
    float* x1 = (float*)calloc(N, sizeof(float)); //resulting sequence 1
    float offset1 = 0xffff;

    //sinewave 2
    float A2 = 0xffff;         //amplitude
    float f2 = 200000.0;       //frequentie
    float fs2 = 1000000.0;     //sample frequentie
    float phi2 = (pi / 2);     //phase
    float* x2 = (float*)calloc(N, sizeof(float)); //resulting sequence 2
    float offset2 = 0xffff;

    float xr[N];
    float xi[N];

    for (int n = 0; n < N; n++) {
        xr[n] = A1 * sin(2 * pi * (f1 / fs1) * n + phi1) + offset1;
        xi[n] = A2 * sin(2 * pi * (f2 / fs2) * n + phi2) + offset2;
        printf("%0.f, \n%0.f, \n", xr[n], xi[n]);
    }

    return 0;
}
```



## FFT OUTPUT VISUALISER WEBPAGE:

```
<!DOCTYPE html>
<html>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.js"></script>
<script>

const xValues = [];
const yValues = [];

var input = "";

var port, textEncoder, writableStreamClosed, writer, historyIndex = -1;
const lineHistory = [];
async function connectSerial() {
  try {
    // Prompt user to select any serial port.
    port = await navigator.serial.requestPort();
    await port.open({ baudRate: 115200 , bufferSize: 1000000});
    let settings = {};

    if (localStorage.dtrOn == "true") settings.dataTerminalReady = false;
    if (localStorage.rtsOn == "true") settings.requestToSend = false;
    if (Object.keys(settings).length > 0) await port.setSignals(settings);

    textEncoder = new TextEncoderStream();
    writableStreamClosed = textEncoder.readable.pipeTo(port.writable);
    writer = textEncoder.writable.getWriter();
    await listenToPort();
  } catch (e){
    alert("Serial Connection Failed" + e);
  }
}

async function listenToPort() {
  const textDecoder = new TextDecoderStream();
  const readableStreamClosed = port.readable.pipeTo(textDecoder.writable);
  const reader = textDecoder.readable.getReader();

  // Listen to data coming from the serial device.
  while (true) {
    const { value, done } = await reader.read();
    if (done) {
      // Allow the serial port to be closed later.
      console.log('[readLoop] DONE', done);
      reader.releaseLock();
      break;
    }

    input += value;

    if(input.endsWith("\r\n"))
    {
      processData();
    }
  }
}
```

```

function processData()
{
// Parse the string into an array
console.log(input);

    const array = JSON.parse(input);

    // Make sure all elements are integers (although they should be if parsed from
JSON)
    const integerArray = array.map(Number);
        console.log(integerArray); // Output: [1, 20, 50, 1, 4]

// console.log(xValues);
//console.log(integerArray);

    for(var i = 0; i < integerArray.length; i++)
    {
        xValues[i] = i;
    }
    new Chart("myChart", {
        type: "line",
        data: {
            labels: xValues,
            datasets: [{
                fill: false,
                lineTension: 0,
                backgroundColor: "rgba(0,0,255,1.0)",
                borderColor: "rgba(0,0,255,0.1)",
                data: integerArray
            }]
        },
        options: {
            animation: {duration: 0},
            legend: {display: false},
            scales: {
                yAxes:[{ticks:{min:0x80000000,max:225000000}}],
            }
        }
    });

input = "";

}
</script>
<body>

<button onclick="connectSerial()">Connect</button>
<button onclick="doClear()">Clear</button>
<br>
<canvas id="myChart" style="width:100%"></canvas>

</body>
</html>
//Met dank aan Jonas Aertgeerts

```





## CONTACT

Louis Van Lommel | Student E-ICT  
R0844517@student.thomasmore.be  
Tel. + 32 71 37 58 63

## VOLG ONS

[www.thomasmore.be](http://www.thomasmore.be)  
[fb.com/ThomasMoreBE](https://fb.com/ThomasMoreBE)  
#WeAreMore

THOMAS  
**MORE**