

UML-diagrammen

UML-diagrammen overzicht

Studie: Technische Informatica, Delft

School: Haagse Hogeschool

Startdatum: 10 februari 2014

Einddatum: 6 juni 2014

Begeleidende docenten: Anthony van Geest & Tony Andrioli

Begeleiders Brightsight: Rob Bekkers & Remko Foekema

Stagiair: Tom Conijn, 10010017

Inhoudsopgave

1. Inleiding	1
2. UML-diagrammen oriëntatiefase	3
2.1. Use case diagram	3
2.2. Use case scenario's	4
2.3. Activiteitendiagram	5
2.4. Klassendiagram	7
3. UML-diagrammen implementatie eerste versie Matrix module	9
3.1. Klassendiagram versie 1:	9
3.2. Ingevuld klassendiagram versie 1	10
3.3. Sequentiediagram oscilloscoop werking simpel:	14
3.4. Sequentiediagram oscilloscoop werking:	15
4. UML-diagrammen implementatie tweede versie Matrix module	17
4.1. Klasse diagram versie 2:	17
4.2. Ingevuld klassendiagram versie 2:	18

1. Inleiding

In dit hoofdstuk worden alle gemaakte UML-diagrammen besproken. Dit document bestaat uit drie hoofdstukken waarvan het eerste hoofdstuk de oriëntatiefase is, het tweede hoofdstuk is de eerste sprint die is uitgevoerd en het derde hoofdstuk komt overeen met de tweede sprint van het afstudeerproject.

2. UML-diagrammen oriëntatiefase

In dit hoofdstuk zijn de UML-diagrammen te zien die zijn gemaakt tijdens de oriëntatiefase van het afstudeerproject. De volgende drie diagrammen zijn gemaakt: use case diagram, activiteitendiagram en een klassendiagram.

2.1. Use case diagram

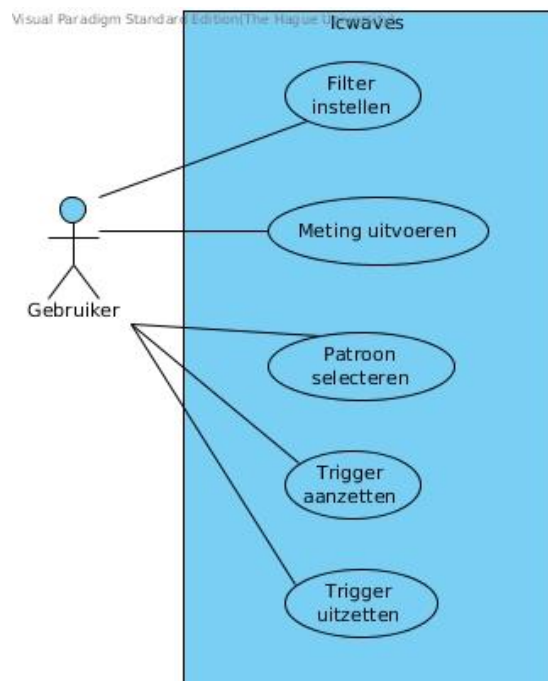
Het eerste diagram dat is gemaakt is een use case diagram, waarin op een hoog level de interactie met de module wordt beschreven. In het use case diagram is één actor te zien, dit is de gebruiker van de Matrix. De acties die de gebruiker wil uitvoeren zijn in overleg met de opdrachtgever opgesteld en beschrijven de vijf belangrijke functionaliteiten van de icWaves trigger box. Het gemaakte use case diagram is te zien in Afbeelding 1 en de use cases worden hieronder kort besproken.

De eerste actie is het instellen van de filter in de icWaves trigger box, dit kan gebruikt worden om zwakke signalen in het ingangssignaal te vinden.

De tweede actie is het uitvoeren van een meting. Zonder deze actie is er geen signaal om een patroon uit te selecteren, waardoor de icWaves geen patronen kan herkennen.

De derde actie is het selecteren van een patroon. Het selecteren van een patroon is belangrijk voor de gebruiker, zodat een goede selectie gemaakt kan worden waarop de icWaves moet reageren.

De laatste twee acties zijn voor het starten en stoppen van een trigger, dit betekent dat de patroon herkenning aan en uit gezet moet kunnen worden.



Afbeelding 1 Use case diagram

2.2. Use case scenario's

Per use case is een scenario geschreven en deze zijn hieronder te zien.

Use case naam	Filter instellen
Samenvatting	De instellingen van de interne filter van de icWaves worden aangepast.
Actoren	Gebruiker
Aannamen	De gebruiker maakt gebruik van de Matrix.
Beschrijving	De gebruiker geeft aan dat de filter ingesteld moet worden. Er wordt aangegeven dat het bereik van de filter opgegeven moet worden en vervolgens moet de filterfrequentie ingesteld worden.
Uitzonderingen	n.v.t
Resultaat	De filter op de icWaves is ingesteld.

Use case naam	Meting uitvoeren
Samenvatting	Een meting wordt uitgevoerd met de icWaves.
Actoren	Gebruiker
Aannamen	De gebruiker maakt gebruik van de Matrix.
Beschrijving	De gebruiker geeft aan dat een meting uitgevoerd moet worden. Hierna moet het bereik van de AD-converter van de ingang worden opgegeven. Vervolgens wordt de bemonsteringssnelheid en het aantal meetpunten ingesteld. Er moet aangegeven worden of gewacht wordt op een trigger en hoelang dit wachten duurt. Na het instellen van deze opties wordt de meting gestart.
Uitzonderingen	n.v.t
Resultaat	Een meting is gestart

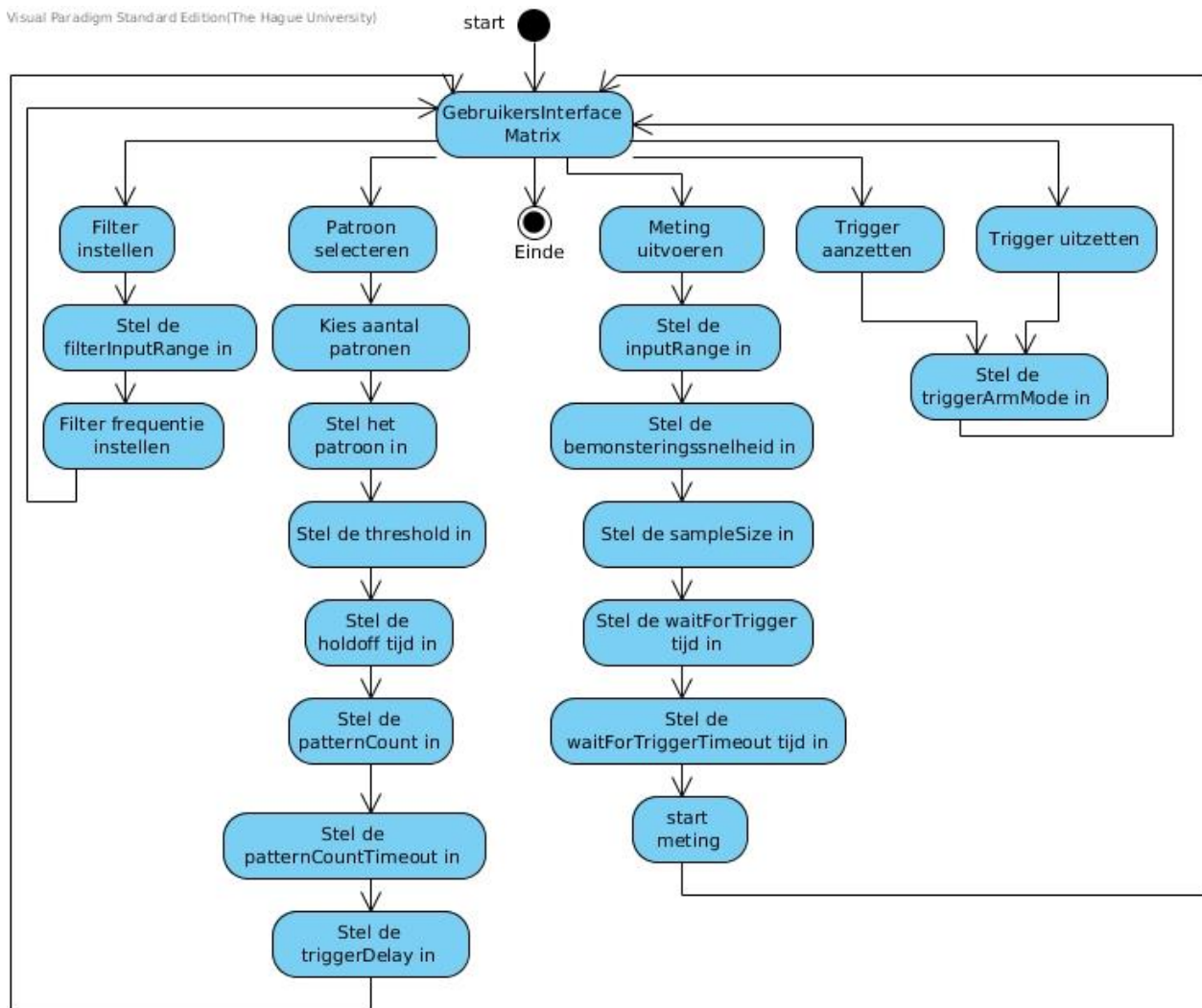
Use case naam	Patroon selecteren
Samenvatting	Het selecteren van een patroon dat herkend moet worden door de icWaves
Actoren	Gebruiker
Aannamen	De gebruiker maakt gebruik van de Matrix.
Beschrijving	De gebruiker geeft aan dat een patroon geselecteerd moet worden. Hierna wordt achterelkaar gevraagd: Het aantal patronen dat herkend moet worden, het patroon, de drempelwaarde die aangeeft wanneer een signaal genoeg lijkt op het opgeslagen patroon. Hierna wordt de tijd ingesteld dat er gewacht wordt op een beter resultaat en het aantal keer dat een patroon herkend moet worden. De laatste twee stappen zijn het instellen van de wachttijd bij het tellen van patronen en de wachttijd voor het versturen van een trigger.
Uitzonderingen	n.v.t
Resultaat	Een patroon is opgeslagen op de icWaves

Use case naam	Trigger aanzetten
Samenvatting	Aangeven op de icWaves dat patronen herkend moeten worden.
Actoren	Gebruiker
Aannamen	De gebruiker maakt gebruik van de Matrix.
Beschrijving	De gebruiker geeft aan dat de trigger aangezet moet worden. Vervolgens kan de trigger mode op patroonherkenning gezet worden.
Uitzonderingen	n.v.t
Resultaat	Patronen worden herkend.

Use case naam	Trigger uitzetten
Samenvatting	Aangeven op de icWaves dat patronen niet meer herkend moeten worden.
Actoren	Gebruiker
Aannamen	De gebruiker maakt gebruik van de Matrix.
Beschrijving	De gebruiker geeft aan dat de trigger uitgezet moet worden. Vervolgens kan de trigger mode op stoppen worden gezet.
Uitzonderingen	n.v.t
Resultaat	Patronen worden niet meer herkend.

2.3. Activiteitendiagram

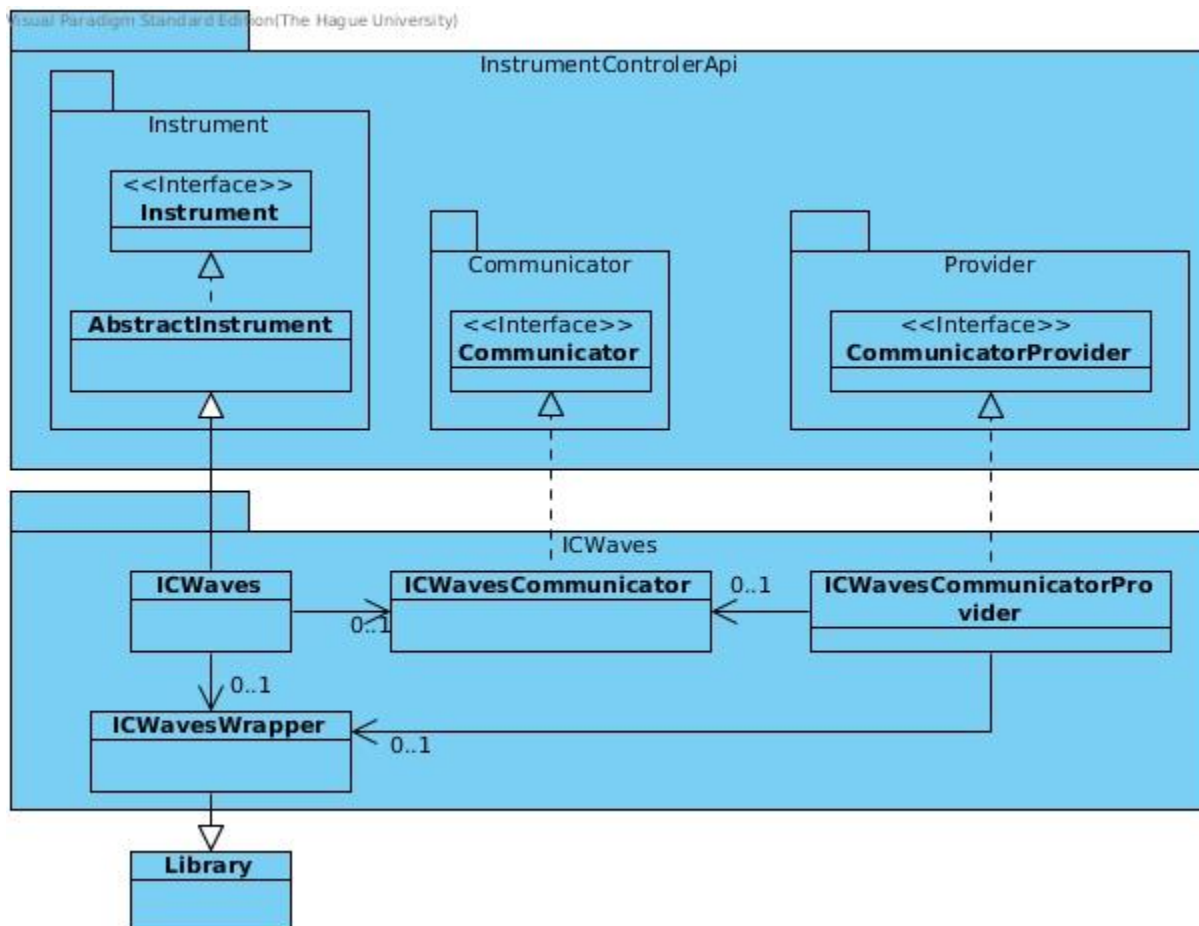
Het activiteitendiagram in Afbeelding 2 begint bij de gebruikersinterface van de Matrix. Vanuit deze activiteit is het mogelijk om alle vijf de use cases te starten. Op het moment dat een use case is gestart worden een aantal activiteiten uitgevoerd om de vereiste opties in te stellen en de actie uit te voeren. De opties die ingesteld moeten worden zijn gevonden door naar een C voorbeeldprogramma te kijken en de documentatie door te lezen die bij de icWaves trigger box is gegeven. Zo moet voor het instellen van de filter het bereik van de filteringang en de filterfrequentie ingesteld worden. Hierna is de activiteit klaar en kan er een nieuwe actie/use case uitgevoerd worden met de gebruikersinterface.



Afbeelding 2 Activiteitendiagram

2.4. Klassendiagram

Tijdens de oriëntatiefase is er ook een klassendiagram gemaakt. In dit diagram staan alleen de klassen die nodig lijken om te kunnen beginnen aan de ontwikkeling van de eerste versie van de Matrix module.



Afbeelding 3 Klassendiagram oriëntatiefase

In het klassendiagram zijn twee belangrijke packages te zien. De bovenste package bestaat uit onderdelen die al in de Matrix zitten en gebruikt moeten worden om een instrument aan te sturen en toe te voegen. De onderste package bevat de klassen die gemaakt moeten worden om de icWaves aan te sturen vanuit de Matrix. Hieronder is per klasse van de onderste package besproken waarvoor deze nodig is.

De ICWaves klasse bevat de code voor het instrument en regelt de communicatie met de ICWavesWrapper

De ICWavesCommunicator doet op dit moment niet zo veel, omdat de communicatie opgezet wordt door de ICWaves klasse. Deze klasse is nog steeds wel nodig, omdat deze verwacht wordt bij het aanmaken van een instrument en bij het zoeken van beschikbare apparaten in de ICWavesCommunicatorProvider.

De ICWavesCommunicatorProvider zorgt ervoor dat het instrument zichtbaar is in het Instrumenten overzicht van de Matrix aan de hand van de ICWavesCommunicator klasse.

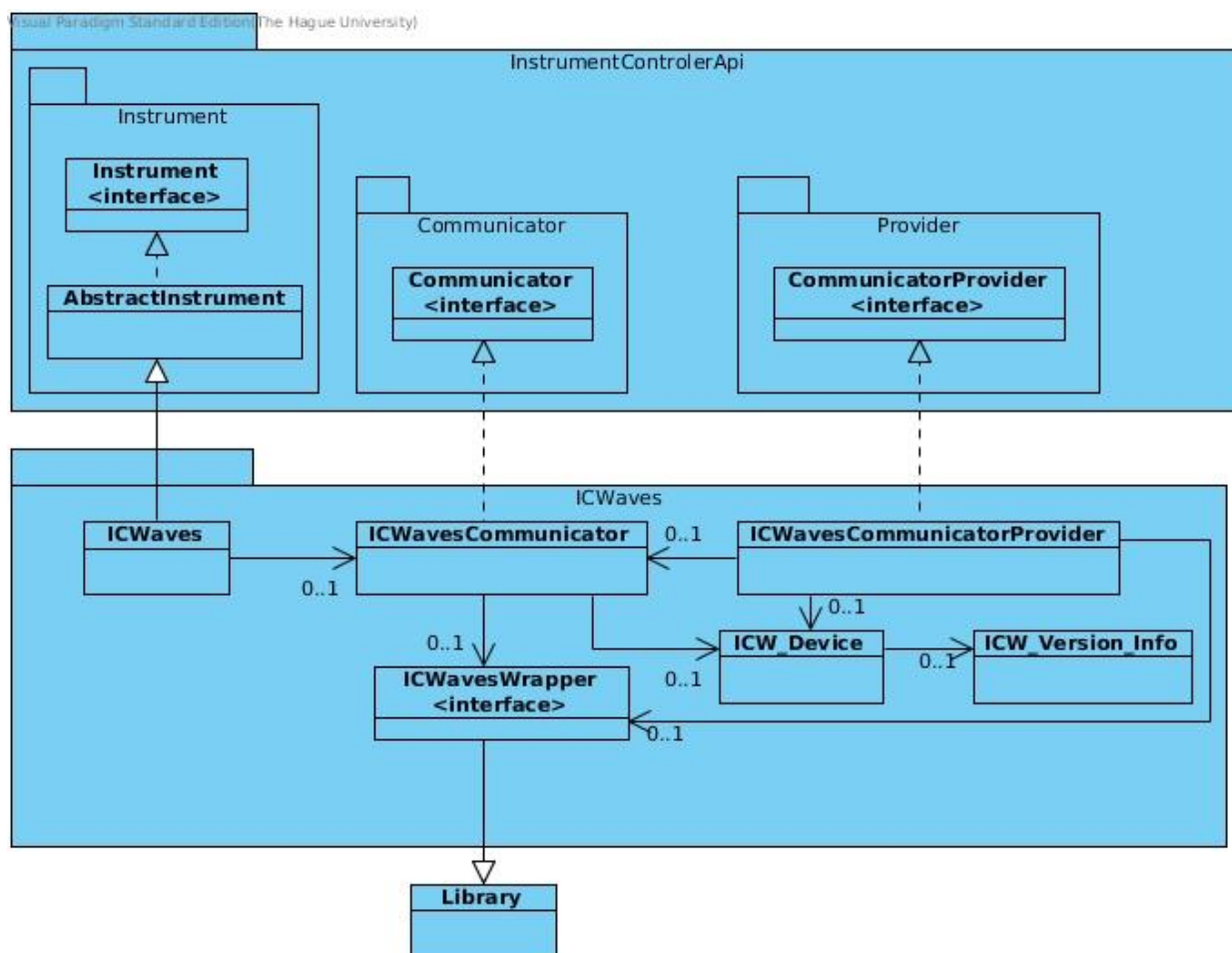
De ICWavesWrapper is een interface die alle methodes van de library beschrijft en maakt gebruik van een library klasse, omdat dit nodig is voor het gebruik van JNA.

3. UML-diagrammen implementatie eerste versie Matrix module

In dit hoofdstuk zijn de UML-diagrammen te zien die zijn gemaakt tijdens de eerste sprint van het afstudeerproject. Het klassendiagram is uitgebreid tijdens deze sprint en er is een sequentiediagram gemaakt.

3.1. Klassendiagram versie 1:

Tijdens het uitvoeren van de eerste sprint zijn er twee klassendiagrammen gemaakt, het eerste klassendiagram bevat geen methodes en attributen en het tweede diagram bevat deze wel. Dit is gedaan, zodat er een beter overzicht is van de klassen.

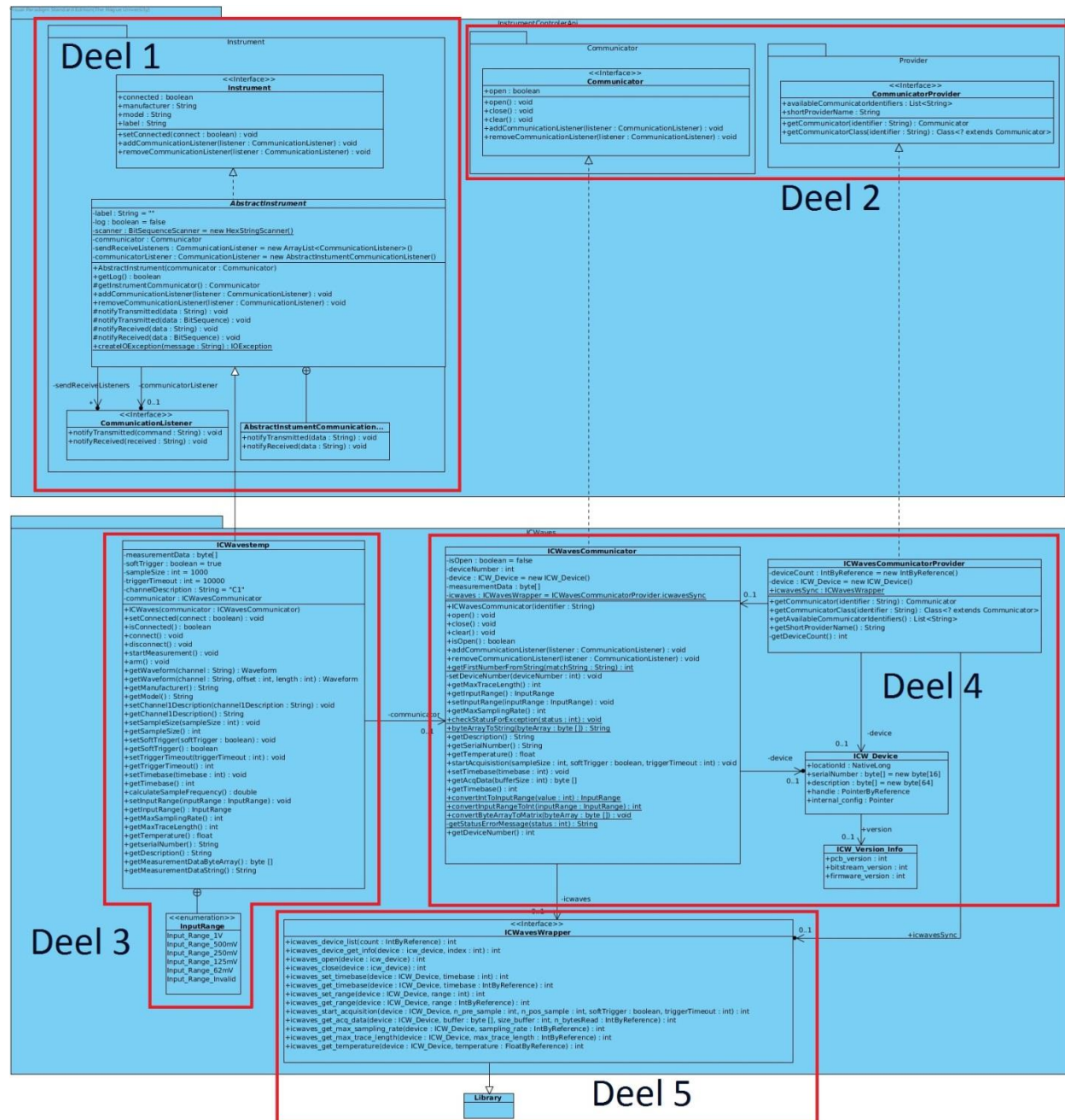


Afbeelding 4 Klassendiagram versie 1

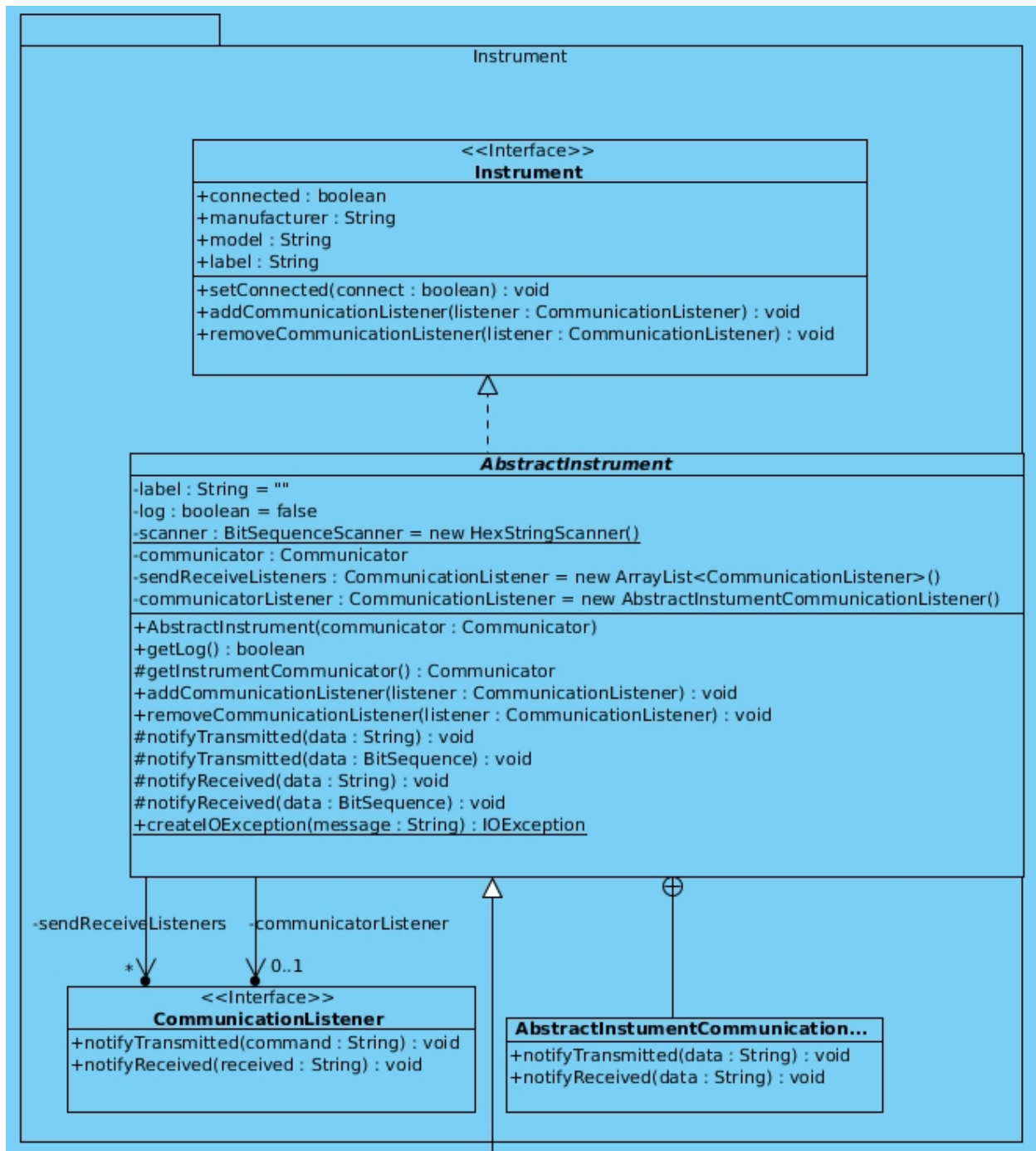
In Afbeelding 4 is het klassendiagram te zien van de eerste sprint. In dit diagram zijn er twee klassen bijgekomen, namelijk de **ICW_Device** en de **ICW_Version_info**. Deze twee klassen zijn nodig om het apparaat aan te geven waarmee gecommuniceerd moet worden. Bij dit ontwerp wordt de communicatie naar de library geregeld door de **ICWavesCommunicator** klasse in plaats van de **ICWaves** klasse.

3.2. Ingevuld klassendiagram versie 1

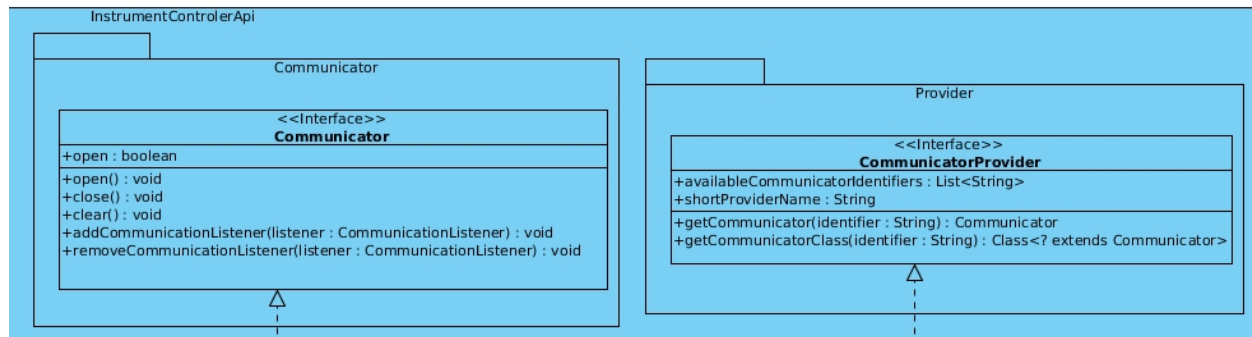
Hieronder is het klassendiagram te zien dat hoort bij de eerste versie van de module. Dit is hetzelfde diagram als bij Hoofdstuk 3.1 alleen zijn hier wel de methodes en attributen te zien. Dit diagram is in vijf delen te zien op de volgende pagina's.



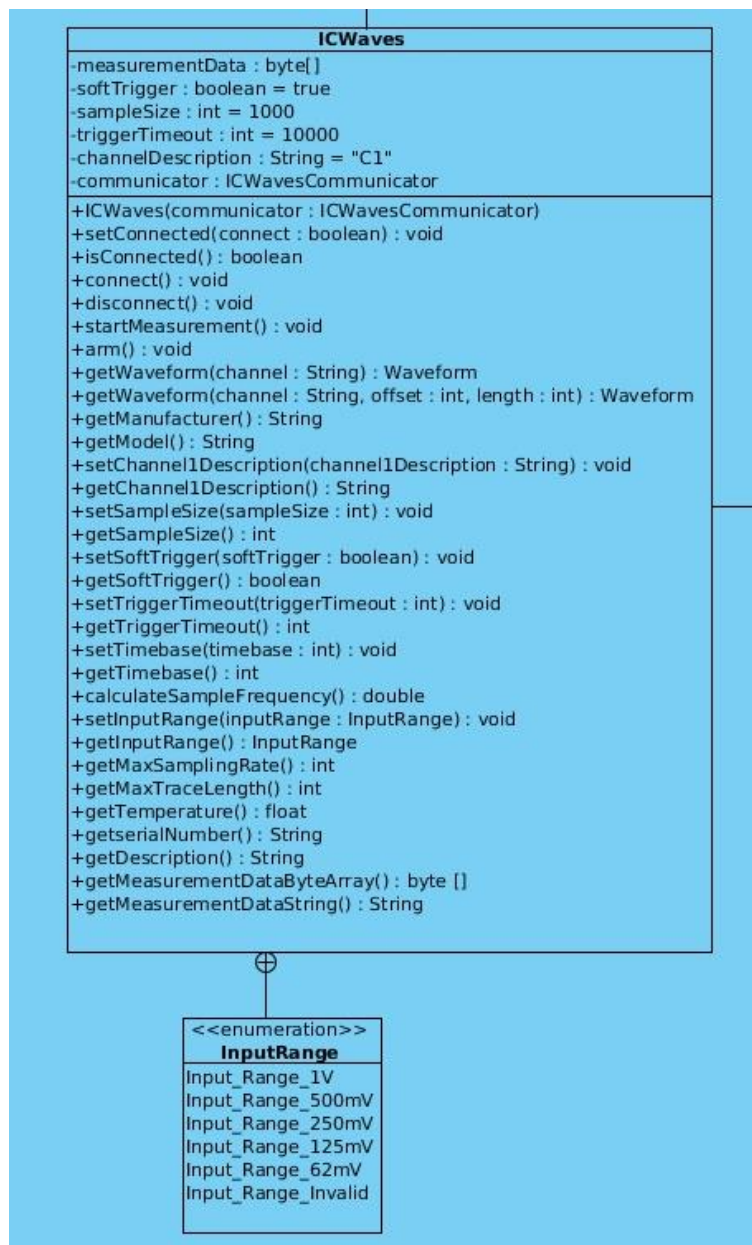
Afbeelding 5 Klassendiagram versie 1 ingevuld



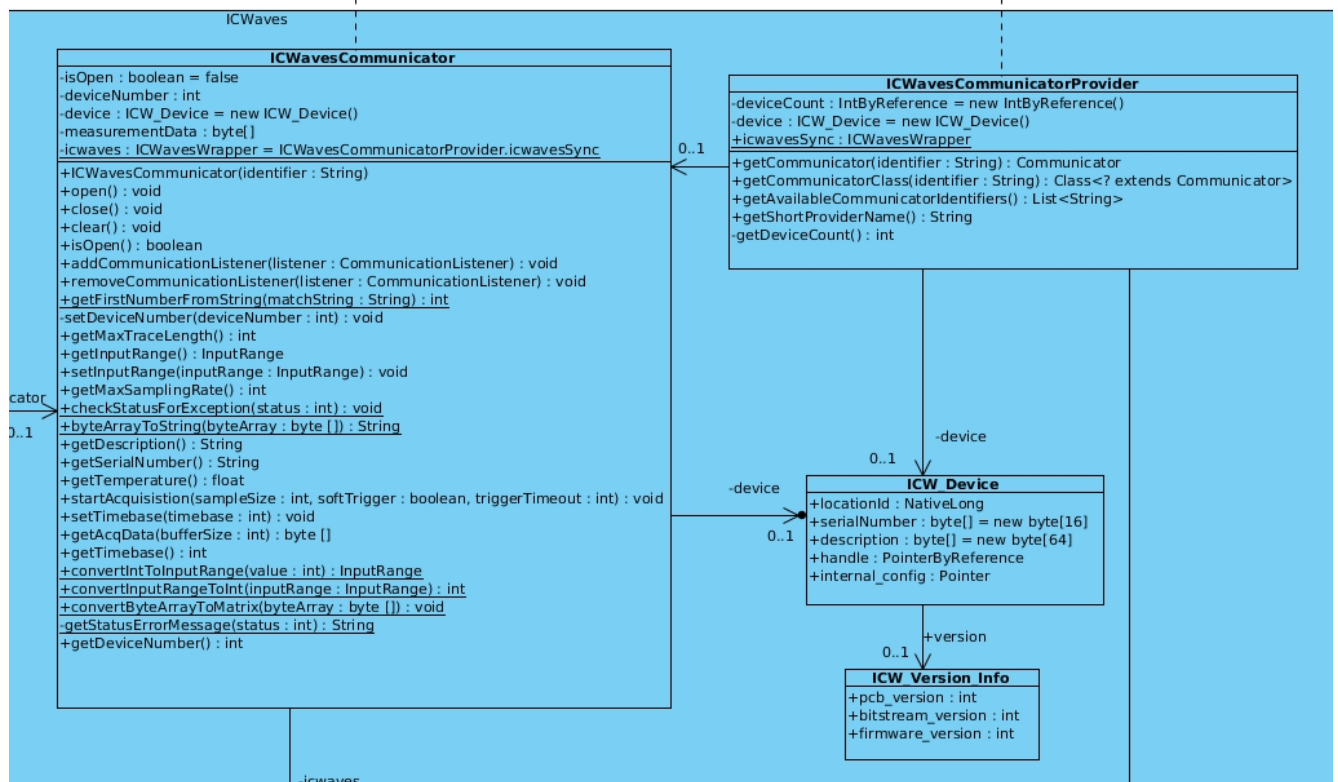
Afbeelding 6 Klassendiagram versie 1(deel 1)



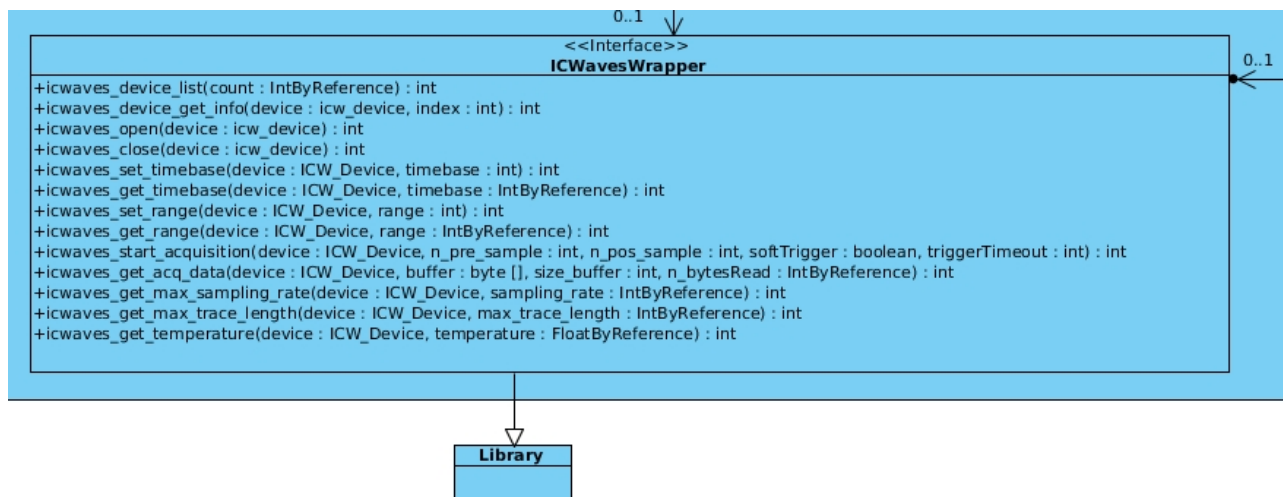
Afbeelding 7 Klassendiagram versie 1(deel 2)



Afbeelding 8 Klassendiagram versie 1(deel 3)



Afbeelding 10 Klassendiagram versie 1(deel 4)



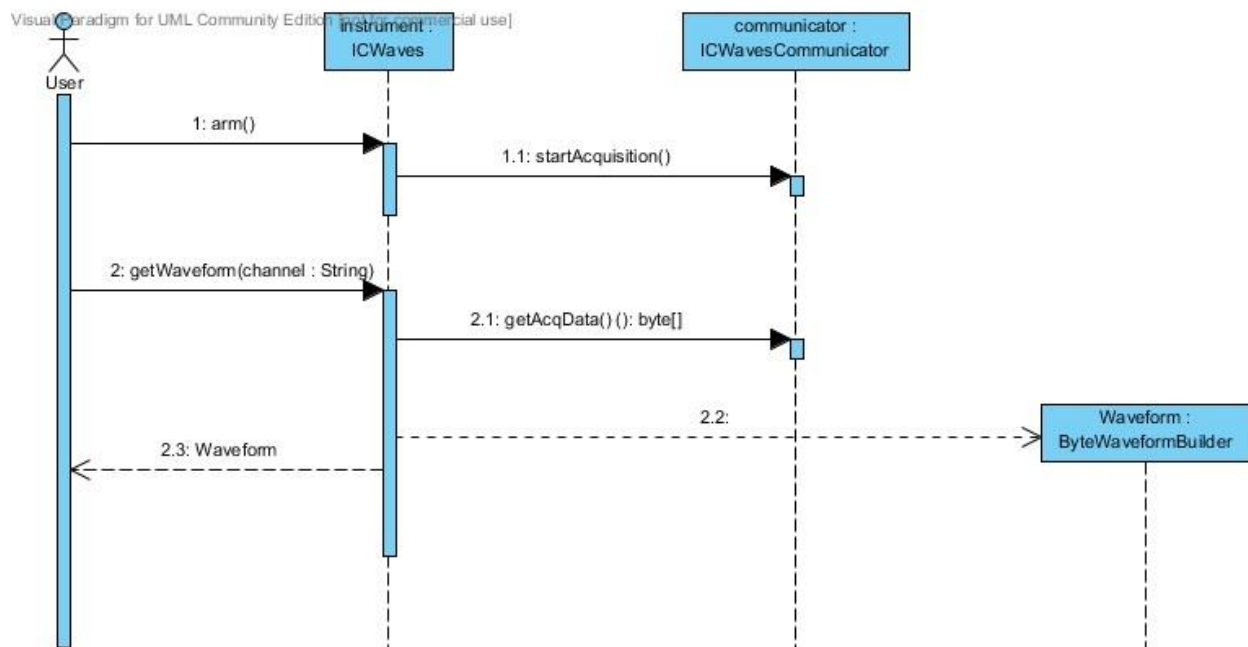
Afbeelding 9 Klassendiagram versie 1(deel 5)

3.3. Sequentiediagram oscilloscoop werking simpel:

In het onderstaande sequentiediagram is de werking van de icWaves trigger box als oscilloscoop te zien. De communicatie naar de library is niet weergegeven in dit diagram.

In het diagram is te zien dat een arm methode wordt gebruikt van de ICWaves. Vervolgens wordt de startAcquisition van de ICWavesCommunicator aangeroepen. Deze communiceert dit vervolgens naar het echte apparaat door middel van de library. De startAcquisition zorgt ervoor dat een meting gestart wordt.

De tweede stap is het opvragen van de opgenomen data, dit gebeurt met getWaveform. Hiermee wordt aan de ICWavesCommunicator de opgenomen data opgevraagd. De verkregen data wordt vervolgens naar een Waveform klasse omgezet en teruggegeven aan de gebruiker.

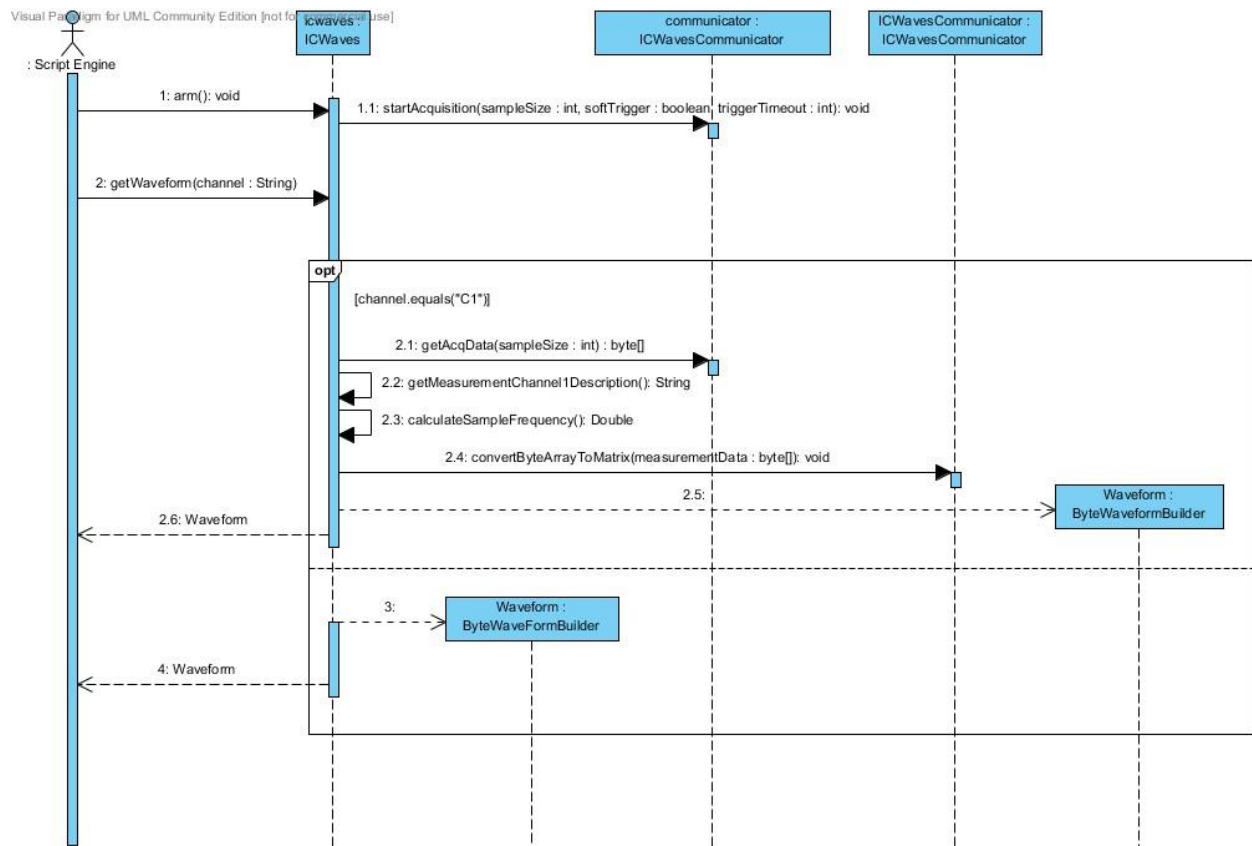


Afbeelding 11 Sequentiediagram oscilloscoop werking simpel

3.4. Sequentiedagram oscilloscoop werking:

In het onderstaande sequentiedagram is opnieuw de werking van de icWaves trigger box als oscilloscoop te zien. Ook in dit diagram is de communicatie met de library niet weergegeven.

In dit diagram is in meer detail de werking van de getWaveform methode weergegeven. Zo is te zien dat als C1 als string wordt meegegeven de opgenomen data wordt opgevraagd, de channel beschrijving en de bemonsteringssnelheid worden berekend. Vervolgens wordt de opgenomen data omgezet naar een het goede formaat en wordt er een Waveform van gemaakt. Wanneer iets anders wordt meegegeven dan C1 wordt een lege Waveform teruggegeven.



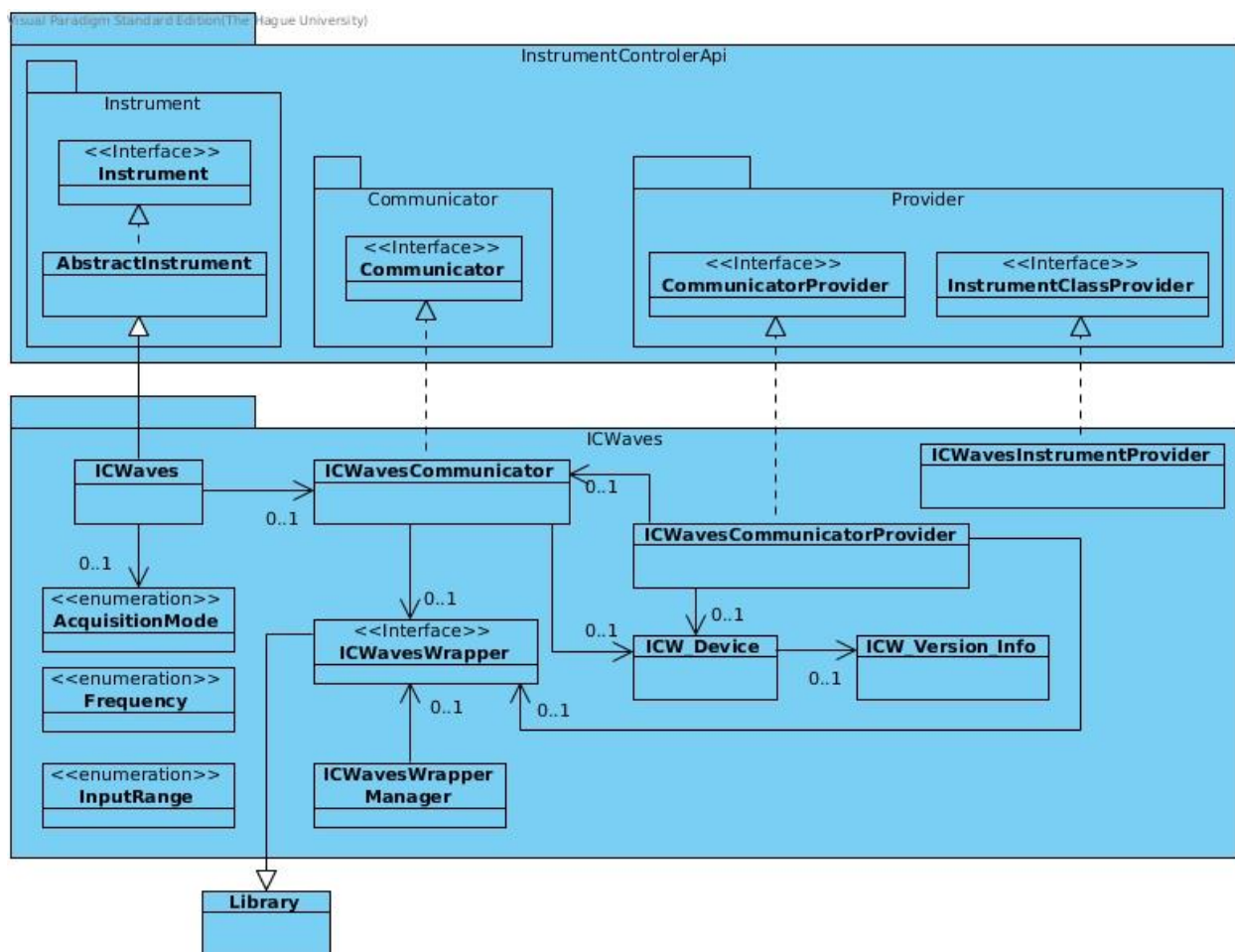
Afbeelding 12 Sequentiedagram oscilloscoop werking

4. UML-diagrammen implementatie tweede versie Matrix module

Tijdens de tweede sprint is alleen het klassendiagram uitgebreid.

4.1. Klasse diagram versie 2:

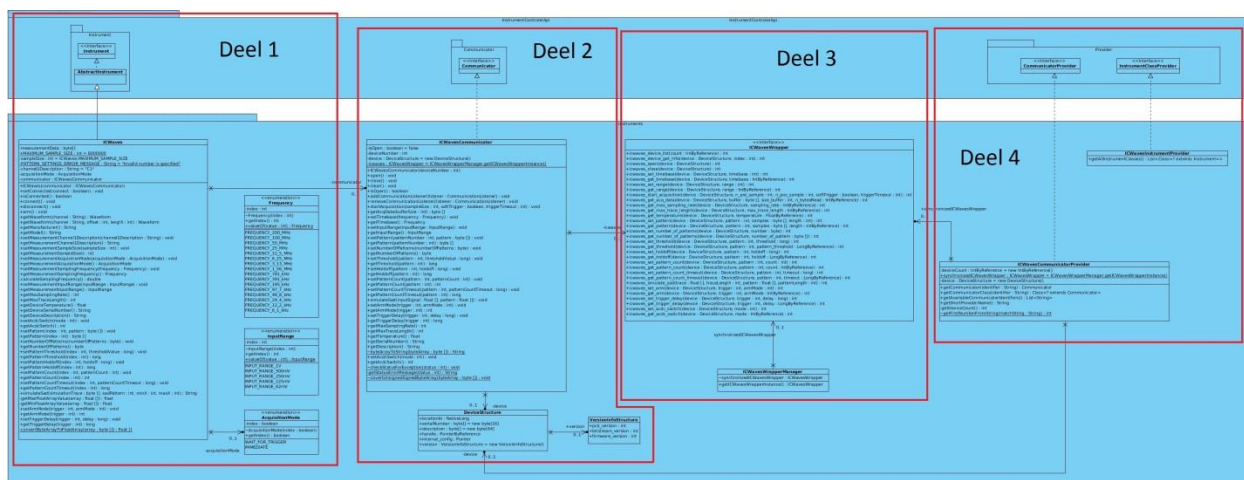
In de onderstaande afbeelding is het klassendiagram te zien die is gemaakt tijdens de tweede sprint van het afstudeerproject. In dit diagram zijn drie enumeraties. Dit zijn enumeraties die gebruikt worden om de manier van opnemen, de bemonsteringssnelheid en het bereik van de AD-converter aan te geven. In het diagram is ook een klasse bijgekomen die de ICWaves klasse registreert als instrument, dit is de ICWavesInstrumentProvider. Zonder deze klasse wordt de icWaves niet herkend als een instrument binnen de Matrix.



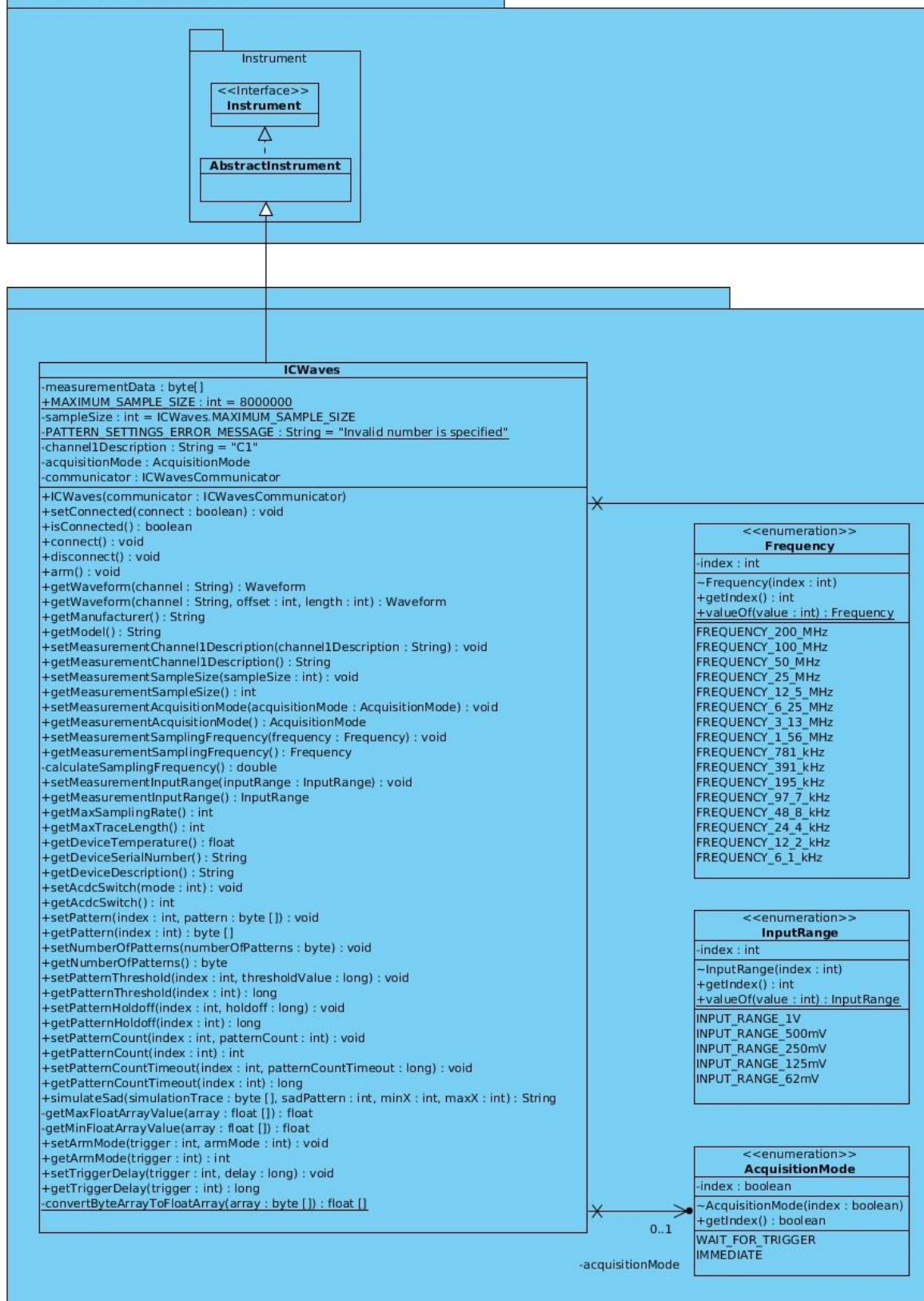
Afbeelding 13 Klassendiagram versie 2

4.2. Ingevuld klassendiagram versie 2:

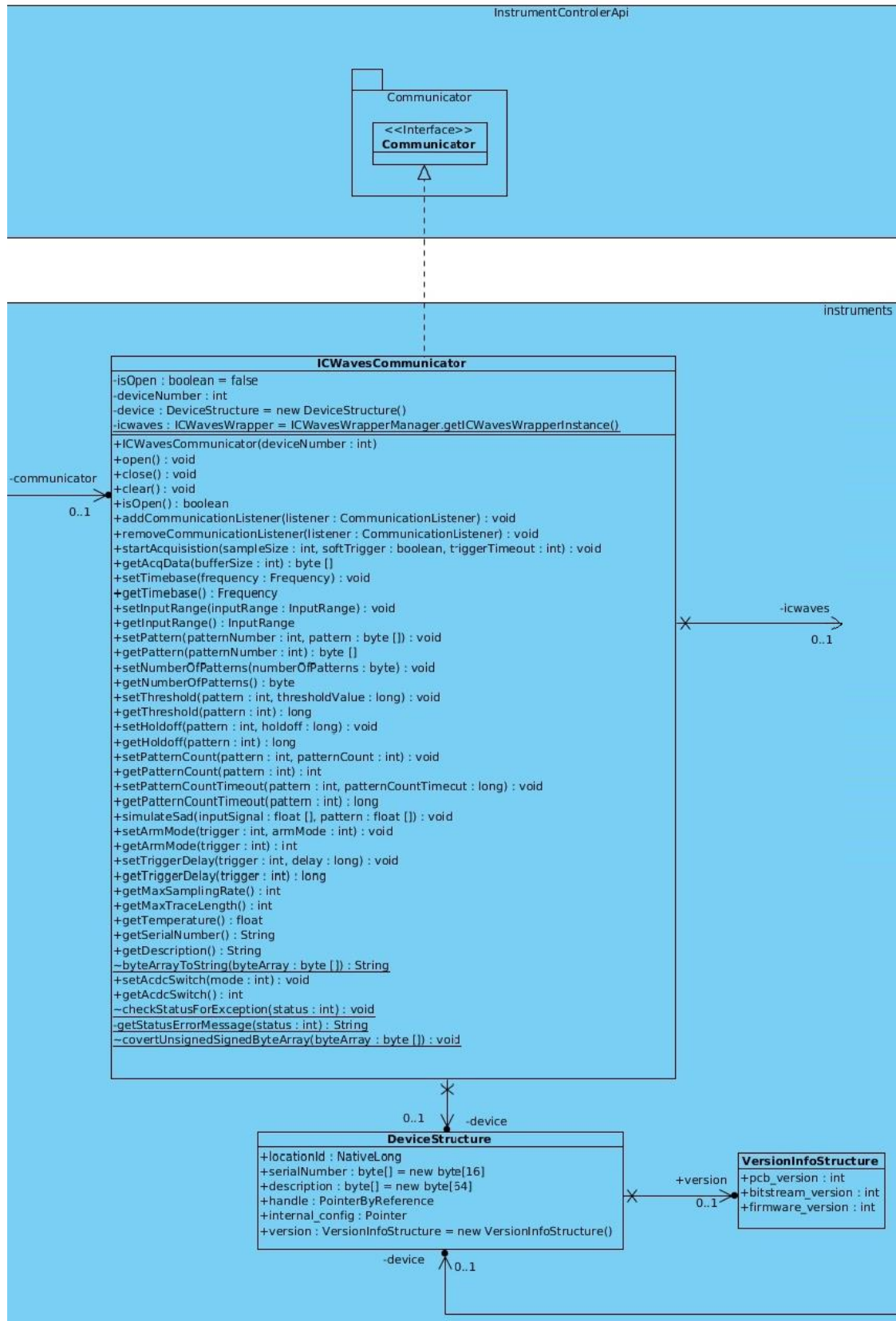
Hieronder is het klassendiagram te zien dat hoort bij de tweede versie van de module. Dit is hetzelfde diagram als bij Hoofdstuk 4.1, alleen zijn hier wel de methodes en attributen te zien. In dit diagram zijn de methodes en attributen van de InstrumentControlerApi package weggelaten, omdat het diagram alleen maar groter hiervan wordt en er geen veranderingen aan gemaakt zijn. Hetzelfde is gedaan voor de library klasse. Het diagram is opgedeeld in vier onderdelen die op de volgende pagina's te zien zijn.



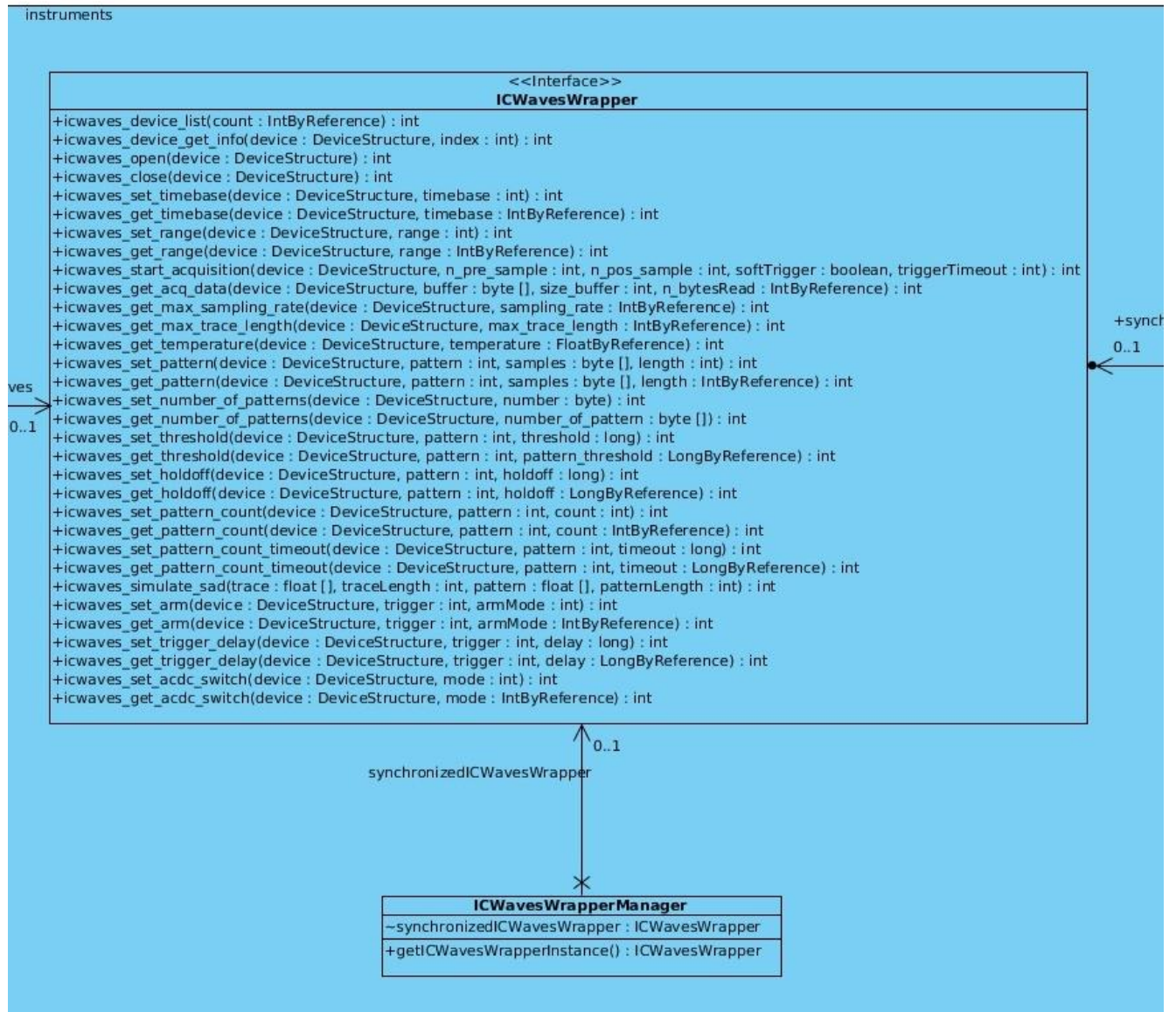
Afbeelding 14 Klassendiagram versie 2 ingevuld



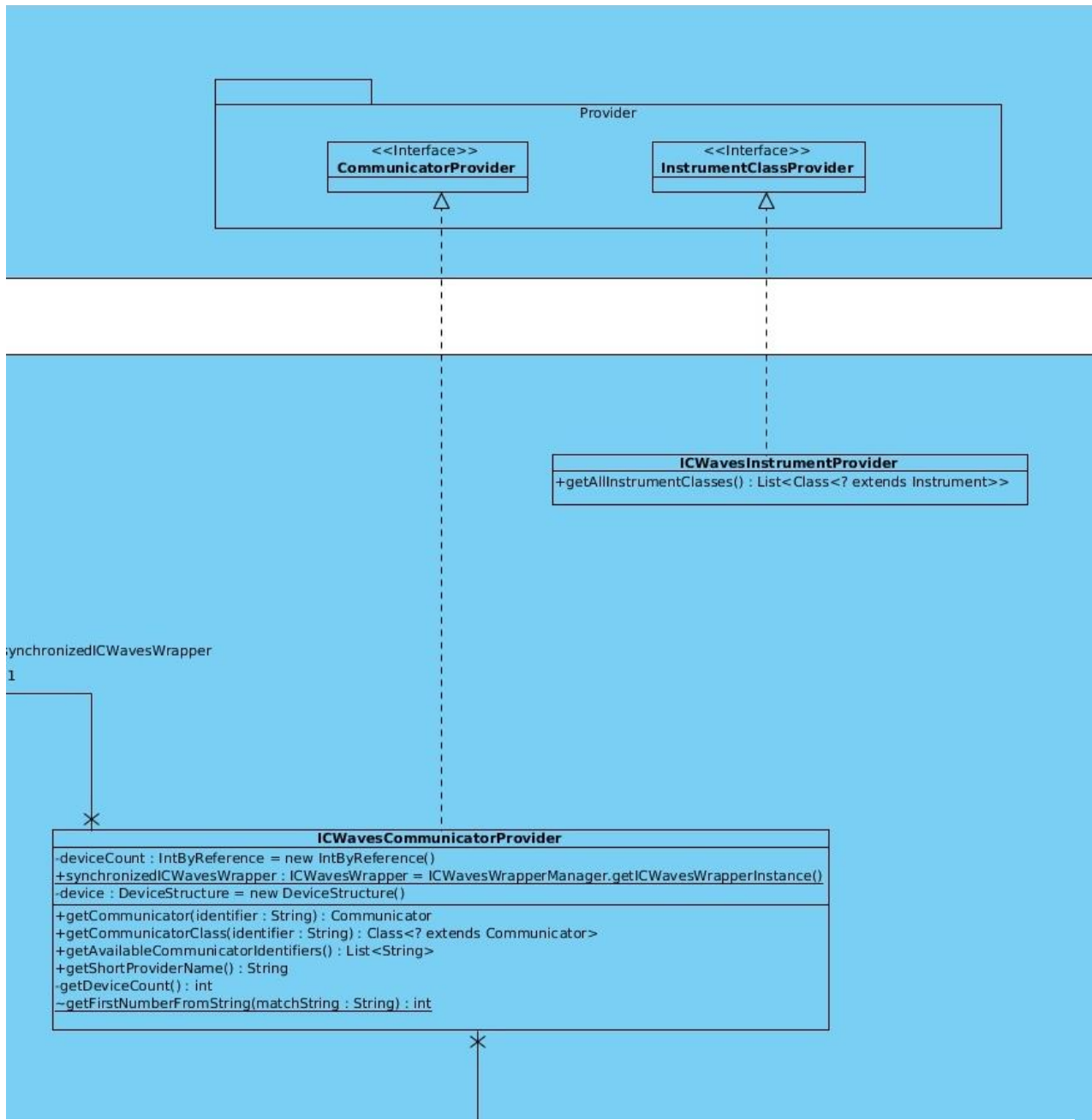
Afbeelding 15 Klassendiagram versie 2 (deel 1)



Afbeelding 16 Klassendiagram versie 2 (deel 2)



Afbeelding 17 Klassendiagram versie 2 (deel 3)



Afbeelding 18 Klassendiagram versie 2 (deel 4)