KU LEUVEN

FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN

CAMPUS BRUGGE

Datacaptatie van Xsens sensoren

Projectlab Master ICT

Kristof T”JONCK

Academiejaar 2017-2018

Samenvatting

Dit jaar werd in KU Leuven campus Brugge een nieuw bewegingslabo geopend. Hier kunnen lichaamsbewegingen opgemeten en geanalyseerd worden aan de hand van sensoren. Een van de sensorpakketten is de XSens Awinda MTw. Die sensoren worden onder andere ingezet voor patiëntengroepen vlak na de plaatsing van een kunstknie. Het is juist dit moment, vroeg in de revalidatie, dat in beeld gebracht moet worden via de sensoren.

Bij de XSens sensoren is een software pakket meegeleverd, om zo de sensoren aan te kunnen sturen. Die software, genaamd de MT Manager, bevat alle tools nodig om alle sensoren volledig te configureren. Als er met de MT Manager data opgenomen wordt, kan achteraf alle data gevisualiseerd worden in grafieken en 3D modellen met die software.

Door problemen met betrekking tot het dataformaat, de instellingen en firmware wordt het moeilijk om data te capteren via de MT Manager. Sensoren verliezen ook vaak de connectie door een lege batterij. Het verlies van de connectie wordt niet weergegeven in de MT Manager. Het is door al deze problemen dat de MT Manager het moeilijk maakt om metingen uit te voeren met bruikbare data als resultaat. Er zou een nieuwe GUI gemaakt moeten worden om zo alle problemen van de MT Manager op te lossen.

XSens voorziet naast de MT Manager ook een SDK. Door gebruik te maken van die SDK kunnen eigen applicaties gemaakt worden, om zo de sensoren aan te sturen. Dit project gebruikt die SDK om de sensoren aan te sturen in een GUI applicatie. Er wordt verder gewerkt met een sample GUI applicatie vanuit die SDK in C++ met behulp van het Qt raamwerk. De sample GUI bevat al de basisfunctionaliteiten om een meting uit te voeren. Door hieraan features toe te voegen, kan een applicatie gemaakt worden naar de wensen van de gebruikers in het bewegingslabo.

Bij de applicatie worden verschillende features toegevoegd die de problemen van de MT Manager oplossen. Daarnaast werden ook nog andere features toegevoegd zoals de synchronisatie met een ander systeem. Het systeem wordt vaak gebruikt in combinatie met de Vicon Nexus. Het is dus gewenst als de data simultaan naast elkaar opgemeten wordt met dezelfde frequentie als het Vicon Nexus systeem.

Alle problemen die er met de MT Manager waren, zijn via deze applicatie opgelost. De functionaliteiten zijn ook uitvoerig getest geweest in het bewegingslabo met Ive Weygers, zodat deze zeker zouden voldoen aan de eisen om doeltreffend metingen uit te kunnen voeren.

INHOUD

Lijst met afkortingen

XDA	XSens Device API
GUI	Graphical User Interface
MTw	Motion Tracker wireless

Lijst met figuren

1 INTRODUCTIE

1.1 XSens Sensoren

In het bewegingslabo van KU Leuven campus Brugge zijn verschillende sensor- en camerasystemen aanwezig. Via die systemen wordt het mogelijk om bewegingen op te meten en te analyseren.

Een van die systemen is de MTw Awinda. Die sensoren werden aangekocht in het zogenaamde 'MTw Awinda Research Bundle' pakket (zie Figuur 1.1).

Figuur 1.1: Awinda Research Bundle [6]

Het bevat acht bewegingssensoren (MTw's) die beweging kunnen opmeten via drie sensoren:

- accelerometer,
- gyroscoop,
- magnetometer.

De sensoren kunnen op het lichaam gekleefd worden met behulp van meegeleverde velcro kleefbanden.

Naast die sensoren bevat het pakket een 'Awinda Recording and Docking Station'. Dit is de master waarmee de MTw's draadloos verbonden kunnen worden om zo data op te nemen. Via dit docking station kunnen de sensoren ook opgeladen worden door deze in te pluggen in de voorziene connectoren. [1, 6]

1.2 Probleemstelling

Bij het gebruik van MT-manager zijn er enkele wederkerende problemen:

- MT-manager genereert een '.mtb'-bestand bij het opnemen van sensordata. Een MATLAB script maakt het mogelijk om dat bestand uit te pakken, om zo de ruwe sensordata (accelerometer, gyroscoop, magnetometer) te kunnen gebruiken in MATLAB. Afhankelijk van de versie van MT Manager waarmee de data opgenomen is loopt deze extractie verkeerd, met onbruikbare data als gevolg.

- Op regelmatige basis verliezen de sensoren de connectie met MT Manager. Dit is vaak omdat de batterij van de MTw leeg is. Het is onduidelijk of alle sensoren nog steeds aan het opnemen zijn en hoe lang deze al aan het opnemen zijn.

- De sample frequentie kan ingesteld worden in MT Manager. Dit wil wel eens fout gaan. Wanneer men denkt op 100Hz te meten, ziet men achteraf andere sampling frequenties verschijnen in de conversie naar MATLAB. De data opgenomen door de sensoren, dient gesynchroniseerd te worden met andere systemen met een eigen sampling frequentie. Om interpolatie technieken te vermijden zou het gewenst zijn dat alle systemen op eenzelfde sample frequentie werken.

- De sensoren en de master hebben een eigen firmware versie. Het is zeer onduidelijk welke versie stabiel is. Verschillende sensoren staan op verschillende firmware versies. Soms worden sensoren helemaal niet meer herkend werden in MT-manager na een update van hun firmware naar de laatste versie.

De sensoren worden onder andere ingezet voor patiëntengroepen vlak na de plaatsing van een kunstknie. Het is juist dit moment, vroeg in de revalidatie, dat in beeld gebracht moet worden. Al deze problemen maken het lastig om bruikbare data te bekomen voor dergelijke toepassingen.

1.3 Doel

Een nieuwe GUI moet gemaakt worden gebruik makende van de Awinda SDK.

Hierbij moeten de problemen vanuit sectie 1.2 opgelost worden.

- De data moet geëxporteerd worden naar MATLAB. Dit door de data op te slaan naar een bestand dat meteen bruikbaar is in MATLAB.
- De status van de sensoren zou bijgehouden moeten worden waardoor het meteen duidelijk is als de verbinding van één van de sensoren verbroken is.
- De sample frequentie van alle apparaten zou gecheckt moeten worden. Dit om te voorkomen dat een van de sensoren ingesteld is met een foute update rate

Naast een oplossing voor de problemen moeten er ook enkele andere basisfeatures voorzien worden. Dit zijn de volgende features:

- Het starten en stoppen van een recording van zeven sensoren simultaan.
- Het sensorsysteem moet gesynchroniseerd worden met een externe trigger. Dit wil zeggen dat het starten en stoppen van een recording moet kunnen gebeuren via een externe puls.

Aangezien de sensoren af en toe gebruikt moeten worden in ziekenhuizen werd gevraagd om de firmware niet aan te passen, zodat de sensoren zeker zouden blijven werken in de ziekenhuizen. Hierdoor moet het probleem omtrent de firmware moet dus niet opgelost worden in dit project.

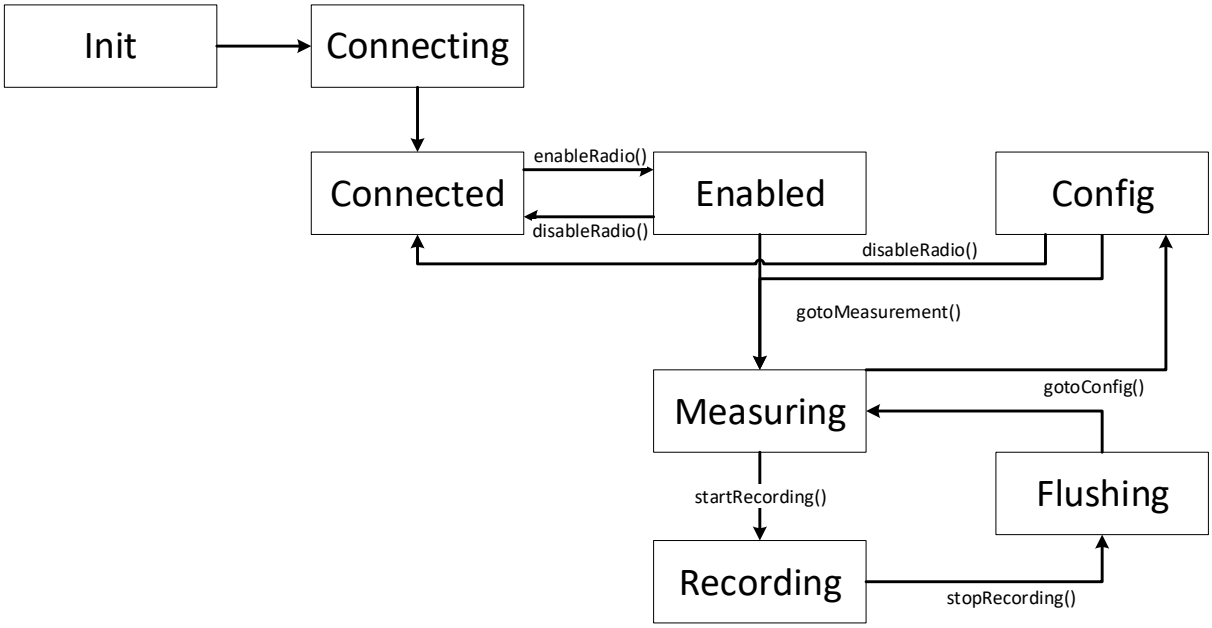
2 MTw Awinda

Het systeem - zowel de Awinda Master als de MTw – is volledig opgebouwd uit state machines. De functionaliteiten van de apparaten zijn specifiek gerelateerd aan een bepaalde toestand. De mogelijke overgangen van de ene naar de andere staat zijn beperkt. Men kan dus niet zomaar van de een naar de ander overschakelen, maar er moet een bepaalde workflow gevolgd worden. Het is dus aangeraden om de toestanden en overgangen van zowel de Awinda Master als van de MTw te begrijpen om zoveel mogelijk volgens die workflow te werken. [2] De bijgeleverde software van de MTw Awinda komt ook in dit hoofdstuk aan bod.

De sate diagramma's zijn beiden te vinden in de documentatie van de SDK. Echter zitten hierin veel fouten, zoals verkeerde acties en schrijffouten. Alle state diagramma's werden hertekend om zo alle fouten van de state diagramma's op te lossen.

2.1 MTw Master

De MTw Master ontvangt gesynchroniseerde gegevens van alle draadloos verbonden MTw's en laadt tegelijkertijd tot 6 MTw's op. Het kan draadloze gegevens ontvangen van maximaal 32 MTw's tegelijk. Het state diagram van de Awinda Master is te zien op Figuur 2.1.



Figuur 2.1: State diagram van de Awinda Master

- **Connecting**
  De Awinda master is nog niet geconnecteerd met de Xsens Device API (XDA). De XDA omvat de API van de SDK om zo de sensoren aan te sturen (zie 2.3.2). Om het apparaat te connecteren kunnen de USB poorten gescand worden via de XDA, om vervolgens de master aan te sturen. Als dit gebeurd is, dan zal de master naar de 'connected' state over gaan.

- **Connected**
  De Awinda master is gedetecteerd geweest door een computer via een USB-poort en is geconnecteerd via de XDA.
  Door de antenne te enablen via een commando zal de radio ingeschakeld worden en zal de master naar de ‘enabled’ state gaan.

- **Enabled**
  De radio antenne van de Awinda Master staat aan en zal een bericht broadcasten op een specifiek kanaal. MTw’s kunnen dit signaal detecteren en vervolgens connecteren met het netwerk.

  Als er één of meerdere MTw’s geconnecteerd zijn met de master, kan een meting gestart worden en zal deze naar de ‘measuring’ state over gaan.

- **Measuring**
  De geconnecteerde MTw’s meten sensorwaarden op en sturen die data door naar de master. De master stuurt dan elk data pakket door naar de XDA.
  Als men wil stoppen met meten kan overgegaan worden naar de ‘config (wireless)’ state.
  Er kan ook gestart worden met het recorden van de sensordata. Door dit commando gaat deze over naar de ‘recording’ state.

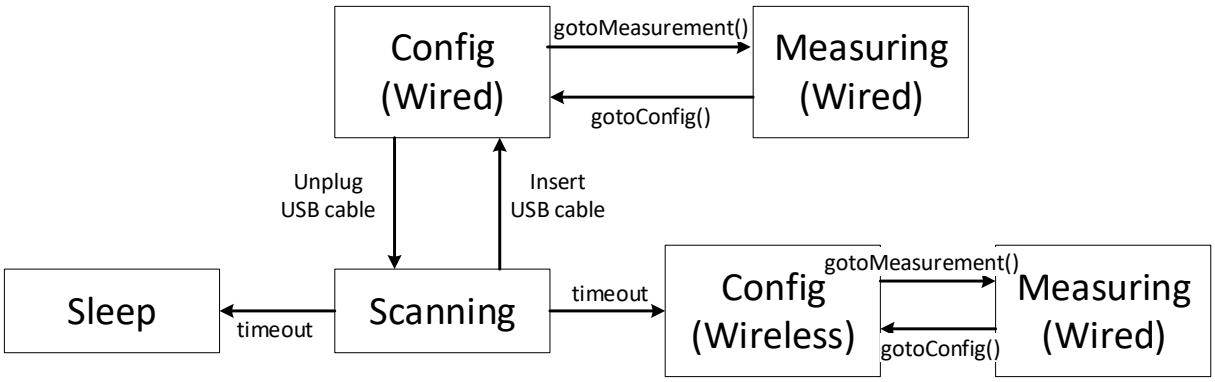
- **Config (wireless)**
  De master is operationeel. Dit wil zeggen dat er nog altijd een draadloos netwerk is, maar dat er geen nieuwe MTw’s meer kunnen connecteren. Om de MTw’s opnieuw te laten connecteren moet de radio antenne uitgezet worden en vervolgens terug aan.
  Er kan ook terug gemeten worden met de reeds geconnecteerde MTw’s waardoor deze terug in de ‘measuring’ state terecht komt.

- **Recording**
  Een recording van de MTw’s is bezig. Een nieuw ‘.mtb’-bestand wordt aangemaakt waarin de data opgeslagen wordt. Bij de maximale update rate zullen de hertransmissies niet opgeslagen worden, bij een lagere update rate zullen deze wel opgeslagen worden.
  Als men wil stoppen met recorden zal deze eerst naar de ‘flushing’ state gaan om daarna terug naar de ‘measuring’ state te gaan.

- **Flushing**
  De recording van de MTw’s is gestopt. Data die nog altijd in de buffer zit van de MTw’s zal opnieuw worden verstuurd naar de master en opgeslagen in de ‘.mtb’-bestand. Als de flushing gedaan is moet het log bestand nog gesloten worden en zal dan terug naar de measuring state over gaan.

2.2 MTw

De MTw of Motion Tracker wireless wordt draadloos verbonden met de MTw master. Ook kan die gedocked worden in de master om de batterij op te laden.



Figuur 2.2: State diagram van de MTw

Zoals te zien in Figuur 2.2 zal de MTw werken aan de hand van een state machine met 6 states.

- **Config (wired)**
  Het apparaat is gedocked in een MTw master via USB. De configuratie van de sensoren kan op dit moment aangepast worden. Dit is echter niet de bedoeling aangezien de sensoren er voor gemaakt zijn om draadloos te verbinden met een master. Die master zal op zijn beurt dan de MTw's besturen en configureren en beheren.

  De MTw kan in een state "Measuring (wired)" geplaatst worden via een functie in de SDK.

  Door de sensor te undocken (USB verbinding verbreken) uit de master zal de sensor in een "Scanning" state terecht komen.

- **Measuring (wired)**
  De sensor kan in deze toestand terecht komen door een gedockte MTw, via een functie van de SDK, te laten beginnen meten.

  De MTw zal via zijn sensoren metingen doen. Die metingen kunnen via de USB connectie opgevangen en verwerkt worden. Ook dit is niet de bedoeling aangezien de sensoren hiervoor gedocked moeten zijn en dus niet vrij rond kunnen bewegen.

  Door het stoppen van de meting zal deze terug in de "Config (wired)" state terecht komen.

- **Scanning**
  In de “Scanning” state zal de sensor continu op alle kanalen zoeken naar een actieve MTw master. Als de Master gedetecteerd wordt zal deze over gaan naar de “Config(Wireless)” state.

  De MTw komt in deze state terecht als de sensor aangezet wordt terwijl deze niet gedocked is in een master. Het opstarten kan op twee manieren aangezet worden:
  - Het docken en vervolgens undocken van de sensor
  - Het duwen op de knop aanwezig op de MTw

  Als de MTw na een tijdje geen master gevonden heeft schakelt deze automatisch over een de “Sleep” state. Om die terug te activeren kan dit op dezelfde manieren als het opstarten. Ook door het roteren van de MTw zal deze opnieuw geactiveerd worden.

- **Sleep**
  De “Sleep” state is een low power modus dat ingeschakeld wordt als de MTw een tijdje inactief is.

  Om de sensor opnieuw actief te krijgen kan de MTw geroteerd worden. Die rotatie zorgt er voor dat de sensor een signaal zal krijgen waardoor hij opnieuw naar de “Scanning” state gebracht zal worden.

- **Config (wireless)**
  In deze state wordt de MTw volledig gecontroleerd door de master. Bij het starten van een meting op de master worden alle geconnecteerde sensoren getriggerd om te starten met meten en komt terecht in de “Measuring (wireless)” state.

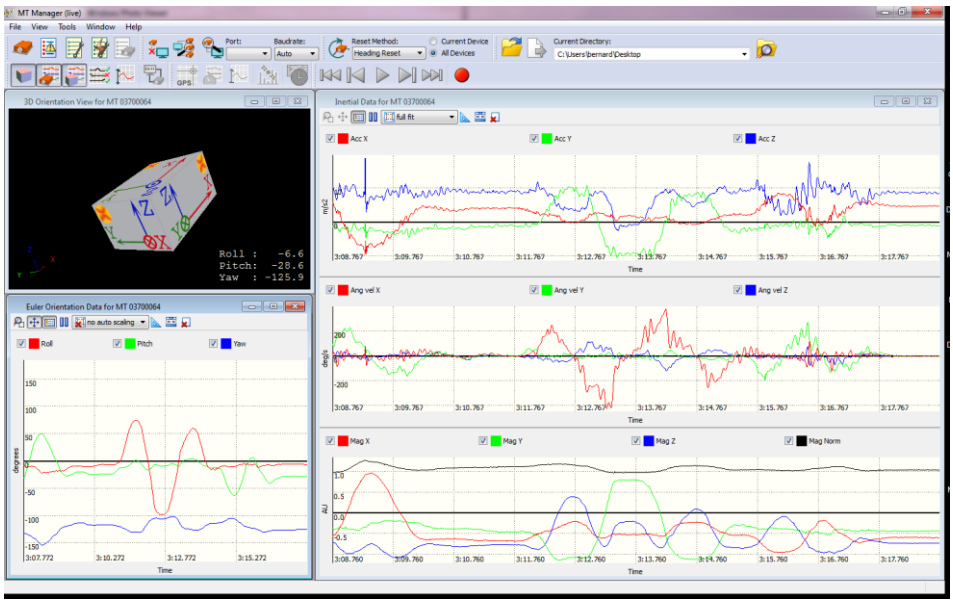
- **Measuring (wireless)**
  Als de master de trigger gegeven heeft zal de sensor beginnen meten. De meetdata van de MTw’s wordt draadloos naar de master verstuurd. Als de master vervolgens een trigger stuurt om te stoppen met sturen komt deze opnieuw terecht in de “Config (wireless) state”.

2.3 Software

De MT Manager software suite, is het officiële softwarepakket waarmee de MTw Awinda geconfigureerd kan worden. Het bevat de MT Manager alsook een MT SDK.

2.3.1 MT Manager

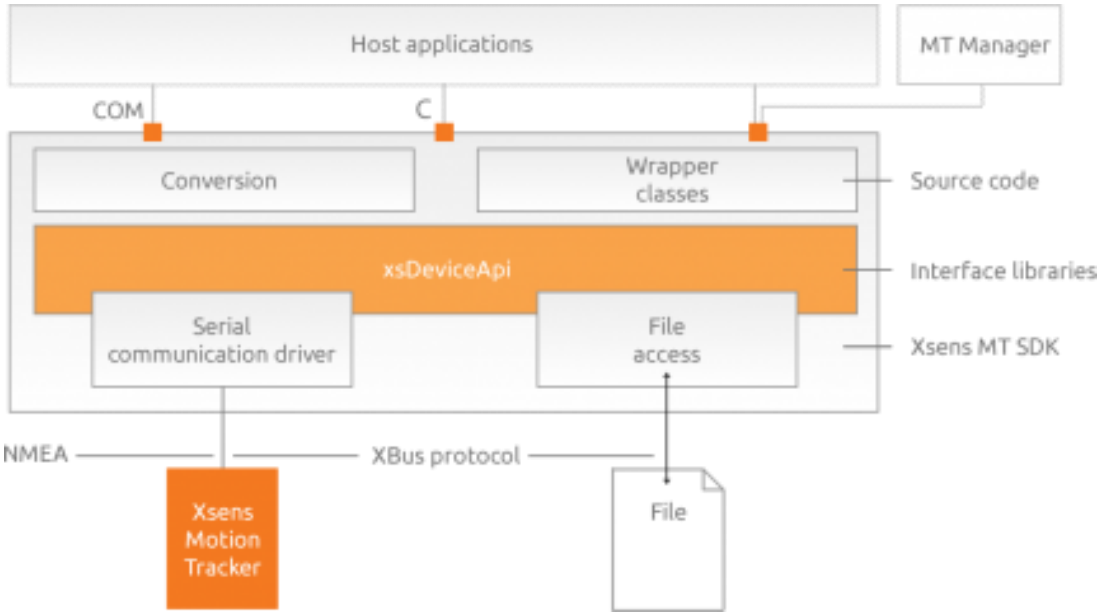
De MTw Awinda kan bestuurd en worden via de MT Manager. Alle geconnecteerde apparaten kunnen via deze tool volledig geconfigureerd worden. Via deze software kan data opgeslagen en gevisualiseerd worden. Ook het synchroniseren met andere apparaten kan hierin ingesteld worden. Zoals te zien op Figuur 2.3 bevat het ook 3D modellen en grafieken om zo de data te kunnen analyseren.



Figuur 2.3: MT Manager

2.3.2 MT SDK

De MT SDK (Software Development Kit) geeft gebruikers de mogelijkheid om de MTw's te configureren en integreren in een eigen applicatie. Het laat ook toe om de output van de motion trackers in real-time op te volgen.



Figuur 2.4: MT software suite [5]

De SDK bevat drivers voor verschillende operating systemen om zo de apparaten aan te kunnen sturen. De source code van de drivers zelf zijn ook meegeleverd, maar het is niet aangeraden om deze te gebruiken voor het ontwikkelen van applicaties.

Om die drivers aan te spreken is een Xsens Device API (XDA) voorzien in de SDK. Dit is een volledige library waarmee de apparaten volledig bestuurd kunnen worden door deze te gebruiken in een applicatie. Zoals te zien op Figuur 2.4: MT software suite zal de MT Manager ook gebruik maken van die API. Bij gebruikersinteracties met de MT Manager zal de API gebruikt worden om zo de settings van het Xsens apparaat aan te passen. Er worden libraries met DLL's voorzien voor ontwikkelaars (32- en 64-bit). Hiermee kunnen ontwikkelaars direct aan de slag gaan om een dergelijke applicatie te bouwen. [5]

Er zijn code samples voorzien voor:

- C#,
- C++,
- MATLAB.

Bij C# en C++ is er een code sample voorzien in de vorm van een console applicatie alsook een GUI applicatie. In MATLAB is er enkel een simpel script voorzien.

3 UITWERKING

Er is er voor gekozen om verder te werken met een van de sample GUI applicaties. Dit aangezien de sample GUI's de meeste basisfunctionaliteiten bevatten die geïmplementeerd moeten worden. Het volgend hoofdstuk bespreekt welke keuzes er gemaakt werden met betrekking tot de programmeertaal, hoe de sample applicatie opgebouwd is en op welke manier er extra features toegevoegd zijn aan die applicatie.

3.1 Keuze van de programmeertaal

3.1.1 C#

Oorspronkelijk werd er gekozen om gebruik te maken van de sample C# applicatie. Dit is een project opgebouwd via Visual Studio en kan ook meteen daar geopend worden via het 'sln'-bestand. Aangezien Visual Studio al reeds geïnstalleerd was op de computer was het evident om te starten met dit project.

De GUI werd opgebouwd in een form. Dit is een drag-and-drop tool waarmee op een eenvoudige manier GUI's aangemaakt en aangepast kunnen worden. Aangezien er nieuwe features toegevoegd moeten worden aan de GUI was dit ook een groot pluspunt.

Door enkele problemen werd het lastig om verder te werken met C#. Die redenen waren:

- Om de XDA te gebruiken wordt er gebruik gemaakt van een wrapper rond de C++ library. Bij het gebruik van de applicatie komen er telkens foutmeldingen naar boven. Hierbij wordt er weinig informatie weergegeven over waar het precies fout liep. Verder werken met dergelijke problemen is niet ideaal.
- Omdat de data naar MATLAB te krijgen werd er aan gedacht om de data op te slaan in een '.MAT'-bestand. MATLAB heeft hiervoor enkel een C++ library ter beschikking. Hiervoor is ook geen wrapper beschikbaar in C#. Zonder die library is het praktisch onmogelijk om die '.MAT'-bestanden te maken.

Om die redenen werd er dus gekozen om niet meer via C# verder te werken maar wel om te kiezen voor de andere sample GUI via C++.

3.1.2 C++

De GUI sample in C++ maakt gebruik van Qt. Dit is een van de meest gebruikte software ontwikkelingsframewerken in C++. Het bevat tools en libraries om zo op een snelle manier C++ programma's en grafische user interfaces te ontwikkelen.

Figuur 3.1: Logo van Qt [3]

Qt heeft een software pakket, namelijk de Qt Creator. Dit is een IDE waarin veel hulpmiddelen zitten om de Qt programma's te ontwikkelen. Verder kan men in de Qt Creator ook gemakkelijk GUI's aanmaken. Dit door deze puur in C++ aan te maken of door de drag-and-droptools te gebruiken die voorzien zijn in de IDE.

De sample applicatie heeft een GUI via drag-and-drop. Om die sample te openen moet de Qt Creator dus gedownload worden, om zo het project te kunnen aanpassen en compileren. [3]

3.2 Opbouw van de sample GUI applicatie

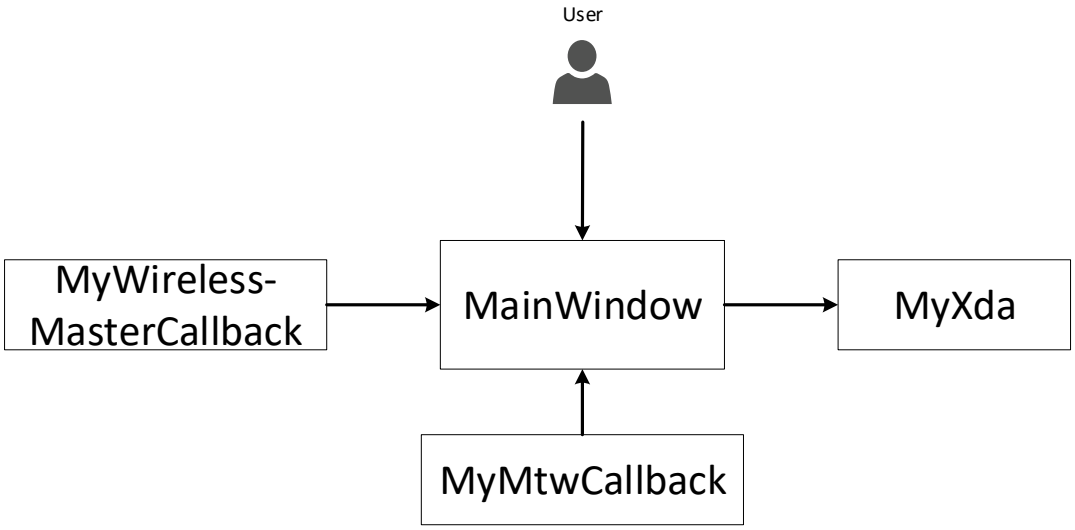
Om aanpassingen te doen en features toe te voegen is het belangrijk dat het hoofdprogramma eerst gesnapt wordt. In het sample programma zijn vier klassen belangrijk:

- MainWindow
- MyXda
- MyMtwCallback
- MyWirelessMasterCallback

De MainWindow is gelinkt aan een form en zal de basis vormen om de GUI te tonen en veranderen. Het bevat een state machine dat ongeveer overeenstemt met de states van de master. Naarmate de master veranderd zal de GUI ook veranderen door signalen van andere bronnen.

De interacties zijn afkomstig van drie bronnen (zie Figuur 3.2). Die bronnen zijn:

- de gebruiker,
- de MyWirelessCallback,
- de MyMtwCallback



Figuur 3.2: Opbouw van de applicatie

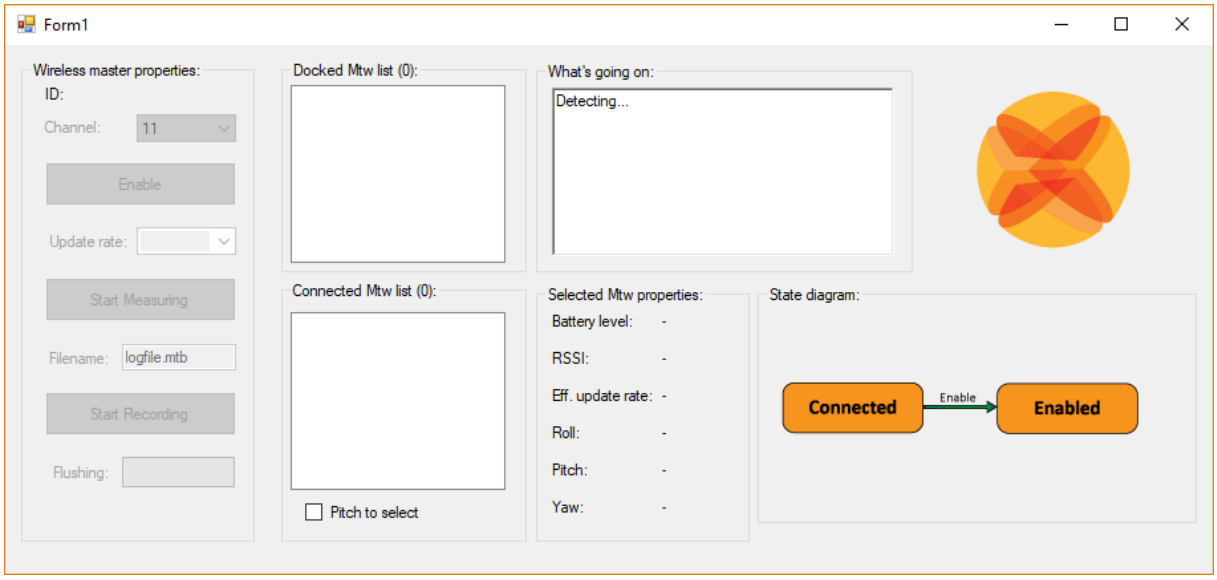
Alle interacties met de MainWindow klasse worden via een ‘connect’ methode in de MainWindow gelinkt aan een functie die uitgevoerd moet worden. De ‘connect’ methode is een functie van Qt om GUI objecten en andere signalen te linken aan bepaalde acties. De GUI en de XDA worden zoveel mogelijk gesplitst. Alle code met betrekking tot het aansturen van de MTw’s zit in de MyXda klasse. De logica van het programma zelf zit en de GUI zit in de MainWindow klasse.

Er wordt verder gewerkt met de structuur die aangemaakt geweest is in de sample GUI. Anders zou de gehele applicatie herschreven moeten worden waardoor er eventueel geen tijd zou zijn om alle features toe te voegen.

3.2.1 Gebruikersinteracties

De gebruiker zal gebruik maken van de GUI om zo de master aan te sturen. De MainWindow zal op zich dan via de Qt ‘connect’ een methode aanroepen, om zo via de XDA de master te besturen.

Een screenshot van de GUI is te zien op Figuur 3.3. De GUI ziet er voor C++ en C# exact hetzelfde uit en heeft dus bijgevolg dezelfde functionaliteiten.



Figuur 3.3: Sample GUI

De GUI bevat de basisfunctionaliteiten om zo te kunnen starten met het meten en opnemen van data. Dit door de master aan te sturen via verschillende knoppen. Een state diagram van de master zal gevisualiseerd worden in de GUI met de acties die kunnen uitgevoerd worden om zo naar een andere state te gaan. Het programma volgt dan ook de state machine van de master zoals aangegeven in sectie 2.1.

De sensoren worden ook gevisualiseerd. Als er een sensor gedocked is in de master dan zal deze terecht komen in de ‘Docked Mtw’ lijst. Als er een MTw draadloos verbonden is met de master wordt deze toegevoegd aan de ‘Connected Mtw’ lijst. Als een MTw geselecteerd wordt in de geconnecteerde lijst, dan kunnen de eigenschappen van die MTw waargenomen worden in de GUI.

Verder is er ook nog een ‘log’-venster aanwezig waar informatie over de interacties met de master en de sensoren te vinden zijn. Als er bijvoorbeeld sensoren connecteren met de master wordt het in dat venster gelogd.

3.2.2 Interacties met de callback

Als de draadloze Awinda Master geconnecteerd is wordt hier een ‘MyWirelessMasterCallback’ aan gekoppeld. Die callback klasse wordt met de master gebonden en stuurt signalen naar de GUI telkens als er bepaalde gebeurtenissen geweest zijn in de master. Als de master gestart is met recorden wordt er bijvoorbeeld een callback gestuurd genaamd ‘recordingStarted’. Die callback wordt via de ‘connect’-methode opnieuw gelinkt met een methode die voor de bijhorende callback een functie uitvoert.

Aan een bepaalde MTw kan ook een callback gekoppeld worden, namelijk de ‘MyMtwCallback’. Als er in de GUI op start measuring gedrukt wordt en men selecteert een MTw uit de ‘Connected Mtw List’ dan wordt deze callback aan de geselecteerde MTw gelinkt. Telkens data beschikbaar is van de geselecteerde MTw, wordt dit via een callback beschikbaar gesteld aan de MainWindow. Telkens die data ontvangen wordt in de MainWindow worden de ‘Selected MTw properties’ aangepast, opnieuw door een functie te linken aan die callback.

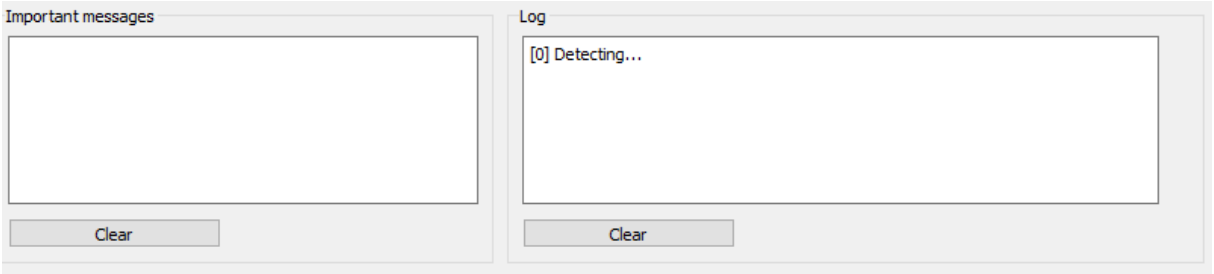
3.3 Extra toegevoegde features

3.3.1 Loggen van belangrijke fouten en errors

In de sample applicatie zit al een venster waarin de logging gebeurt van bepaalde gebeurtenissen. Bijvoorbeeld als de master van de ene state naar de ander gaat. Als er iets fout gaat in de applicatie zal dit ook in het log venster terecht komen.

Het is moeilijk om te onderscheiden welke acties puur voor debug redenen zijn in het log venster en welke een actie van de gebruiker verwachten. Om die reden is er een nieuw venster voorzien genaamd ‘Important messages’. Hierin komen alle berichten waarbij een zware fout gebeurd is en dat er een actie van de gebruiker verwacht wordt. Een ‘err’ methode is aangemaakt waarmee een bericht naar het nieuw aangemaakte venster gestuurd kan worden. Met de clear knop wordt alle tekst gewist in de vensters.

Als er bijvoorbeeld zoals in sectie 0 een fout gebeurt bij het instellen van de sample frequentie zal dit via de ‘err’ methode gemeld worden in het venster van belangrijke berichten.. Ook als er een sensor connectie verliest doordat de batterij leeg is zal er via de master callback telkens een error doorgestuurd worden. Als die error zich voordoet wordt zo een bericht geplaatst in de ‘Important messages’, om zo aan te duiden dat een van de sensoren niet meer geconnecteerd is. Hierdoor kan de gebruiker onmiddellijk ingrijpen.



Figuur 3.4: Tekst vensters voor belangrijke berichten en logging

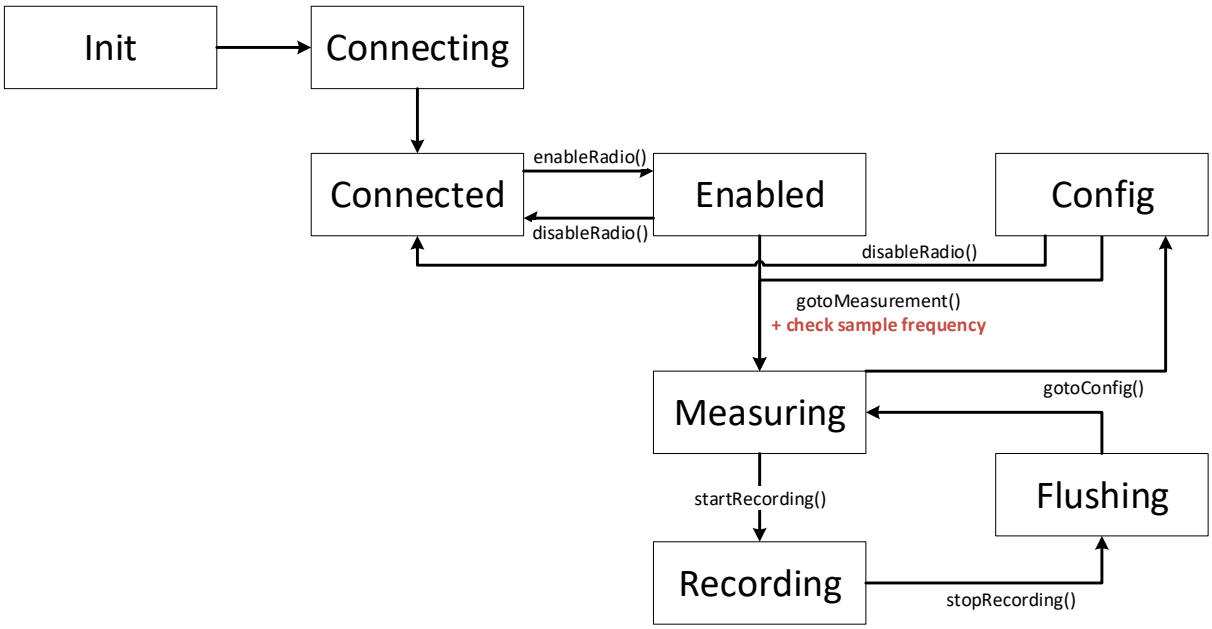
3.3.2 Detectie van foute instellingen

Aangezien de sample frequentie soms fout ingesteld wordt, zoals reeds vermeld in sectie 1.2, moet er een manier voorzien worden waarmee gecheckt wordt de sample frequentie zeker correct ingesteld is.

De sample frequentie van de master en MTw's kan enkel ingesteld worden als de master in de 'enabled' of de 'config' state zit.

Via de applicatie kan een sample frequentie gekozen worden. Door vervolgens op de 'Start measuring' knop te duwen zal de sample frequentie ingesteld worden van de MTw's via de master. Na het instellen van die sample frequentie wordt het commando verstuurd om te starten met opmeten.

Het is net tussen het instellen van de sample frequentie en het starten van de meting dat een extra check geïmplementeerd is. Op Figuur 3.5 is in het rood aangeduid waar de extra check gebeurt in de state machine van de master.



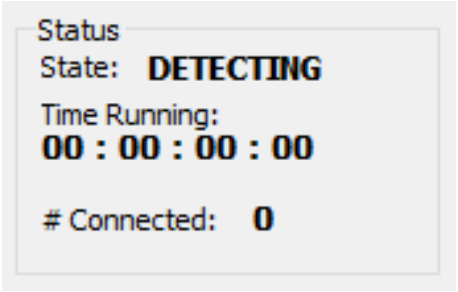
Figuur 3.5: Aanpassing van de state machine voor het checken van de sample rate

In de 'MyXda' klasse wordt een methode toegevoegd waarbij een bepaalde update rate gecheckt kan worden. In de GUI wordt, na het instellen van de sample rate, die functie gebruikt om te kijken als de gekozen waarde overeenkomt met alle ingestelde waarden. Via de XsControl van de MyXda klasse is het mogelijk om alle devices uit te lezen en aan te sturen.

Door via de XsControl de geconnecteerde apparaten uit te lezen kan de de master gevonden worden. Via de master kunnen alle child devices opgevraagd worden, dit zijn de MTw's die draadloos verbonden zijn. Van die child devices kan telkens gecheckt worden wat de update rate is en als die overeenstemt met de gewenste sample frequentie.

Als één van de MTw's een foute update rate heeft zal er in het veld van de belangrijke berichten een foutmelding komen. Anders gaat deze naar de 'Measuring' state.

3.3.3 Monitoren van de status



Figuur 3.6: Status venster

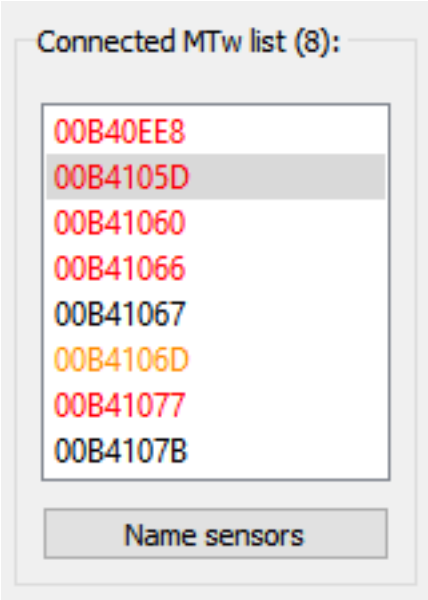
Telkens de state machine van het programma verandert, wordt de state in de GUI ook aangepast zodat men meteen kan zien in welke state de master zich bevindt.

Bij het recorden is het gewenst om te weten hoe lang er al aan het recorden is. Een timer via QTime wordt hiervoor geïnitialiseerd in het begin van het programma. Als de master via een callback aangeeft dat de recording gestart is wordt deze timer gestart. Bij het starten zal er na het duwen op de 'start recording' knop, gestart worden met meten. Door via de callback te werken zal de timer pas gestart worden als de recording werkelijk gestart is en niet wanneer er op de knop gedrukt is. De timer wordt opnieuw aangemaakt als de master via een callback aangeeft dat hij opnieuw in de measuring state terecht is gekomen vanuit de recording state, wat dus wil zeggen dat de recording gestopt is. Door de timer opnieuw aan te maken zal deze stoppen en kan opnieuw gestart worden van nul. Een stop functie voor de QTime is namelijk niet beschikbaar waardoor dit de enige mogelijkheid is.

Om de tijd op de GUI te updaten wordt er gebruik gemaakt van een timer van de QTimer klasse. Hierbij kan er een interval ingesteld worden waarbij deze telkens een timeout signaal stuurt na een interval en dan herstart deze van nul. Dit interval wordt op 100ms gezet zodat de timer om de 100ms een timeout signaal stuurt. Het timeout signaal wordt geconnecteerd met een functie die de timer van het recorden weergeeft op de GUI. Als er aan het recorden is zal de tijd dus om de 100ms veranderen. Als er geen recording bezig is zal de timer van het recorden nog niet gestart zijn waardoor de tijd op de GUI niet geüpdatet wordt. Er wordt namelijk telkens gecheckt als de QTime van het recorden actief is of niet.

Alle geconnecteerde MTw's worden opgeslagen in een lijst. Door te kijken hoe groot die lijst is kan men weten hoeveel er geconnecteerd zijn. Elke keer een MTw connecteert met de master, stuurt de master een callback. Via die callback wordt er telkens een MTw toegevoegd in de lijst. Op dat moment kan het label veranderd worden met het aantal geconnecteerde MTw's.

3.3.4 Monitoren van de batterijen



Figuur 3.7: Batterij status van geconnecteerde MTw's

De sensoren verliezen soms hun connectie doordat de batterij leeg is. Via een timer wordt het batterij level sensor per sensor opgevraagd in het sample programma. De lijst met verbonden MTw's wordt hiervoor één per één overlopen. Dit aangezien de master niet alle batterijlevels tegelijk kan opvragen. De timer in het sample programma werkt met een interval van 1000ms, hierdoor zend deze om de seconde een timeout signaal. Aan dit signaal wordt een actie gelinkt waarbij elke timeout het batterijlevel van de volgende sensor zal opvragen. Als er acht sensoren verbonden zijn zal het dus acht seconden duren vooraleer het batterijlevel van die sensor opnieuw opgevraagd wordt.

Als een measurement gestart wordt, dan zal er een nieuwe callback geregistreerd worden. Die zal inkomende data checken op berichten met betrekking tot het batterijlevel. Aangezien er om de seconde een batterijlevel opgevraagd wordt van een MTw, zal die MTw via een callback zijn batterijlevel terug sturen. Aan die callback wordt een functie gelinkt. Die functie zal de batterijstatus van de verschillende devices bijhouden in een lijst.

In de functie waarbij de batterijstatus bijgehouden wordt, zal de MTw in de lijst met geconnecteerde MTw's aangepast worden. Bij het aanpassen van de batterijstatus van een bepaalde MTw wordt er ook een kleur gegeven aan die MTw in de lijst. Dit zodat men direct visueel kan waarnemen welke batterijen bijna leeg zijn. Zoals in Figuur 3.7 wordt de MTw een kleur gegeven volgens het ontvangen batterijlevel. De volgende kleuren worden ingesteld:

- $30\% \leq batterijlevel$: $zwart$
- $15\% \leq batterijlevel < 30\%$: $oranje$
- $batterijlevel < 15\%$: $rood$

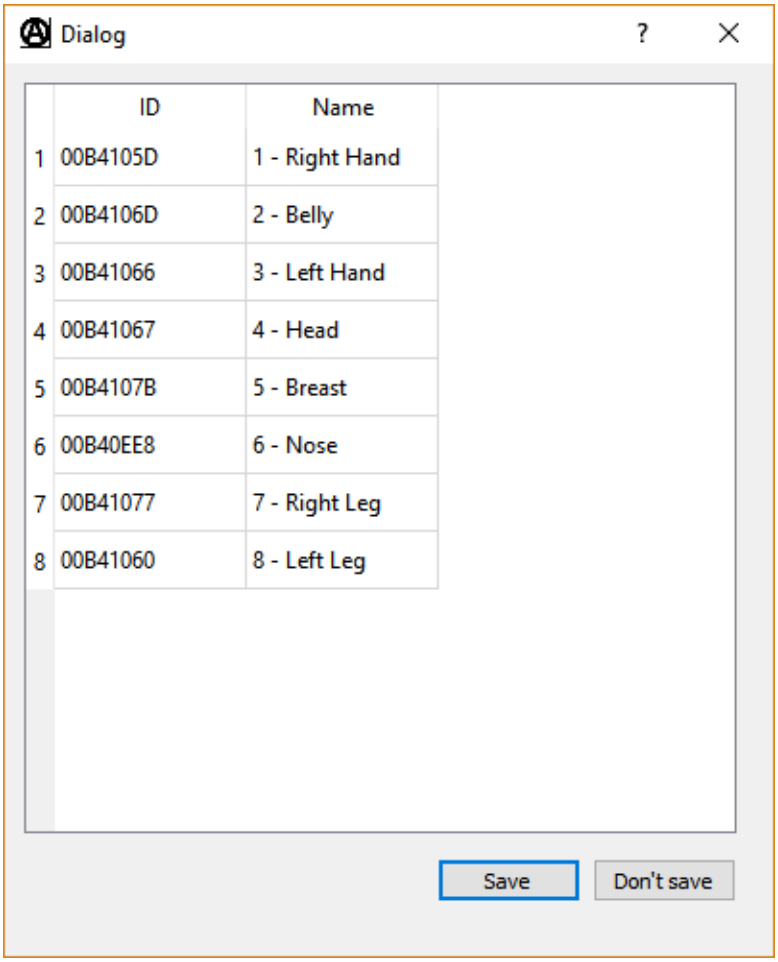
Bij de lijst van de geconnecteerde MTw's is ook een knop toegevoegd die een venster opent om de sensoren een naam te geven. Dit wordt verder uitgelegd in sectie 3.3.5.

3.3.5 Linken van een naam met een sensor

Bij het opnemen van data zou het handig zijn dat elke MTw een naam krijgt waardoor de data direct gelinkt kan worden aan een lichaamsdeel. Hiervoor is een venster voorzien dat geopend kan worden door op de 'Name sensor' knop te duwen. Het venster is te zien in Figuur 3.8. Het venster bevat een tabel met alle IDs van de MTw's. Die tabel wordt geladen volgens een QHash list vanuit de hoofdapplicatie en gesorteerd volgens de naam.

Elke keer een MTw connecteert met de master, stuurt de master een callback naar de applicatie. Bij die callback wordt gecheckt als de ID van de MTw al als key in de hashlist staat. Als dit niet zo is wordt een nieuwe entry toegevoegd in de QHash met de ID als hash key en een lege string als waarde van de hash. Als vervolgens het venster geopend wordt kan de naam veranderd worden in te tabel. Het ID kan niet veranderd worden. Elke MTw heeft een nummer op zich geplakt (van 1 tot 8). Door de naam dus telkens te beginnen met het nummer en daarna de plaats waar het gehangen werd is worden de MTw's gesorteerd volgens het nummer en kan men meteen zien welk nummer aan welk lichaamsdeel hangt.

Als er in het venster op 'Save' geduwd wordt, worden alle ID's en hun naam opgeslagen in een bestand genaamd 'config.csv'. Bij het openen van de applicatie wordt dat CSV-bestand telkens gelezen en geplaatst in de hashlist. Hierdoor kunnen de namen die vroeger ingesteld werden opnieuw aangepast worden in de tabel. Als er op 'Don't save' gedrukt wordt zal het niet opgeslagen worden in een CSV-bestand, maar worden de ingestelde waarden wel gebruikt voor de sessie.



Figuur 3.8: Linken van een apparaat ID met een naam

De namen die ingevuld zijn worden ook gebruikt in sectie 3.3.6 en 3.3.7 om een naam te geven aan de MTw's bij de conversie naar een CSV-bestand. Hierdoor kan men direct na de conversie ook zien welke MTw's met welke naam gelinkt zijn.

3.3.6 Converteren van een MTB-bestand

Aangezien het MTB-bestand niet altijd uitleesbaar is in MATLAB, zou het handig zijn om reeds bestaande MTB-bestanden te converteren.

3.3.6.1 Maken van een MAT-bestand

Via de C++ library van MATLAB is het mogelijk om een bestandformaat te maken dat direct leesbaar is in MATLAB. In het installatiepad van MATLAB is die library automatisch meegeleverd. Die moet toegevoegd worden aan het project om hiervan gebruik te kunnen maken.

De 'libmat.dll' en 'libmx.dll' bestanden bevatten de functionaliteiten nodig om de MAT-bestanden aan te maken. Arrays en datastructuren kunnen hier ook mee aangemaakt worden. Door de 'mat.h' en 'matrix.h' te importeren in een C++ programma kan er van die functionaliteiten gebruik gemaakt worden. [4]

Het probleem hiermee is dat de 'libmat.dll' en 'libmx.dll' libraries niet apart geïmporteerd kunnen worden. Die libraries hangen namelijk af van de rest van de dll's waardoor de volledige MATLAB library met alle dll's geïmporteerd moet worden in het project. Die volledige library is 2.2 GB groot waardoor het project veel te veel ruimte in beslag zou nemen.

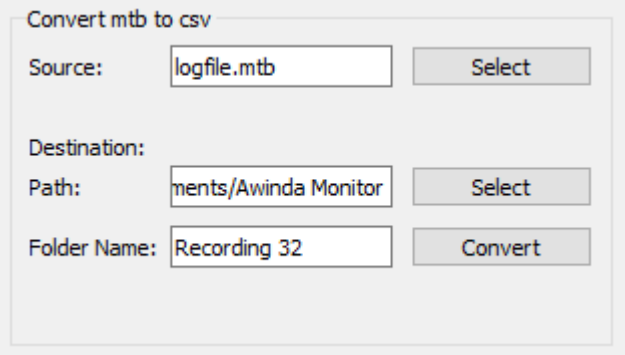
Om die reden wordt er gekozen voor een conversie naar een CSV-bestand in plaats van een MAT-bestand.

3.3.6.2 Maken van een CSV-bestand

Er werd voor een CSV-bestand gekozen omdat deze de problemen met de MAT-bestand oplost. Daarbij heeft een CSV-bestand ook nog enkele andere voordelen:

- Het is makkelijk te implementeren en te parsen.
- Men kan data onmiddellijk bekijken in programma's zoals Excel zonder die zelf te moeten parsen in een applicatie.
- In vergelijking met andere formaten zoals JSON of XML zal het minder plaats in beslag nemen.
- De data is onmiddellijk leesbaar voor het menselijk oog
- Het is compact. Het begint vaak met een kolomnaam en de rest is gevuld met data.

Het nadeel bij een CSV-bestand is dat er geen structuur kan aangebracht worden in de data. Een CSV-bestand bevat op zich dus maar één lijst met een bepaalde structuur data. Andere formaten zoals JSON hebben wel deze functionaliteiten maar moeten eerst geparsed worden om de data te kunnen bekijken. In MATLAB is het ook evidenter om CSV te parsen in plaats van JSON.



Figuur 3.9: GUI voor het converteren

In de GUI wordt een deel toegevoegd waarmee het mogelijk wordt om de conversie uit te voeren.

Een *source* bestand kan geselecteerd worden door op de 'Select' knop te drukken. Aan de knop wordt een actie gelinkt waarbij via een 'QFileDialog', een selectievenster voor het selecteren van bestanden, een bestand geselecteerd kan worden. Het geselecteerde bestand wordt vervolgens ingevoerd in het tekstvak naast de knop.

Ook het pad waar de conversie terecht moet komen kan ingesteld worden via een 'Select' knop op dezelfde manier. Daarnaast moet ook een folder naam ingevuld worden. Dit is de naam van de folder die in het pad aangemaakt zal worden om daarin alle CSV-bestanden op te slaan.

Door vervolgens op de convert knop te duwen zal een conversie van het bestand gebeuren. Hiervoor wordt in de 'MyXda' klasse een functie aangemaakt om het bestand te laden alsook een functie om de geladen data te converteren. Eerst wordt de functie aangeroepen om het bestand te laden, vervolgens de functie die de data converteert.

Het MTB-bestand bevat de configuratie van de 'XsControl'. Die 'XsControl' bevat de data van de apparaten die geconnecteerd waren toen de recording gestart werd. Een nieuwe instantie van de 'MyXda' klasse wordt gebruikt om de data te laden en converteren. Hierdoor zal er zeker geen conflict zijn met de instantie die gebruikt wordt voor het aansturen van de master. Vooraleer de conversie uitgevoerd wordt zal er eerst gecheckt worden als de folder al bestaat. Bestaat de folder nog niet dan zal de conversie meteen starten. Als de folder wel al bestaat dan kan er via een pop-up gekozen worden om die al dan niet te overschrijven. Een dergelijke pop-up venster is een feature van Qt genaamd 'QMessageBox'.

Openen van een log bestand

Via de XsControl kan een log bestand geopend worden. Die zal vervolgens de structuur van master en draadloos geconnecteerde child devices (MTw's) laden. In die master kan een optie ingesteld worden waardoor de opnamegegevens bewaard blijven. Als hierna de functie aangeroepen wordt om de data te laden in de master, zal die alle data laden in de bijhorende softwarematige device objecten (MTw's) en ze daar ook bewaren. Als de optie niet ingesteld wordt om de opnamegegevens te bewaren dan zullen die data geladen worden in de device objecten, maar zullen niet bewaard blijven in het object. Belangrijk is het feit dat er gewacht moet worden bij het laden van de log bestand in de master tot deze volledig geladen is.

Converteren van het log bestand

De functie om de conversie te doen maakt eerst een folder aan die ingesteld geweest is in de GUI. Een nieuw bestand ‘deviceinfo.csv’ wordt aangemaakt. De master wordt opnieuw gevonden door via de XsControl te zoeken naar de master. Via die master kunnen alle childs (MTw’s) opgelijst worden die verbonden waren bij het recorden van de data. Alle childs worden één voor één overlopen. De eigenschappen zoals de device ID, de naam, de sample frequentie en het aantal pakketten worden per MTw weggeschreven naar het deviceinfo bestand. De data zelf kan niet opgeslagen worden in datzelfde bestand. In een subfolder ‘/data’ wordt een bestand aangemaakt per MTw om daar de data in op te slaan. Elke child krijgt een CSV-bestand met het device ID als naam. Elk data pakket wordt lijn per lijn opgeslagen in dat bestand. Elk pakket bevat verschillende parameters gesplitst door een komma.

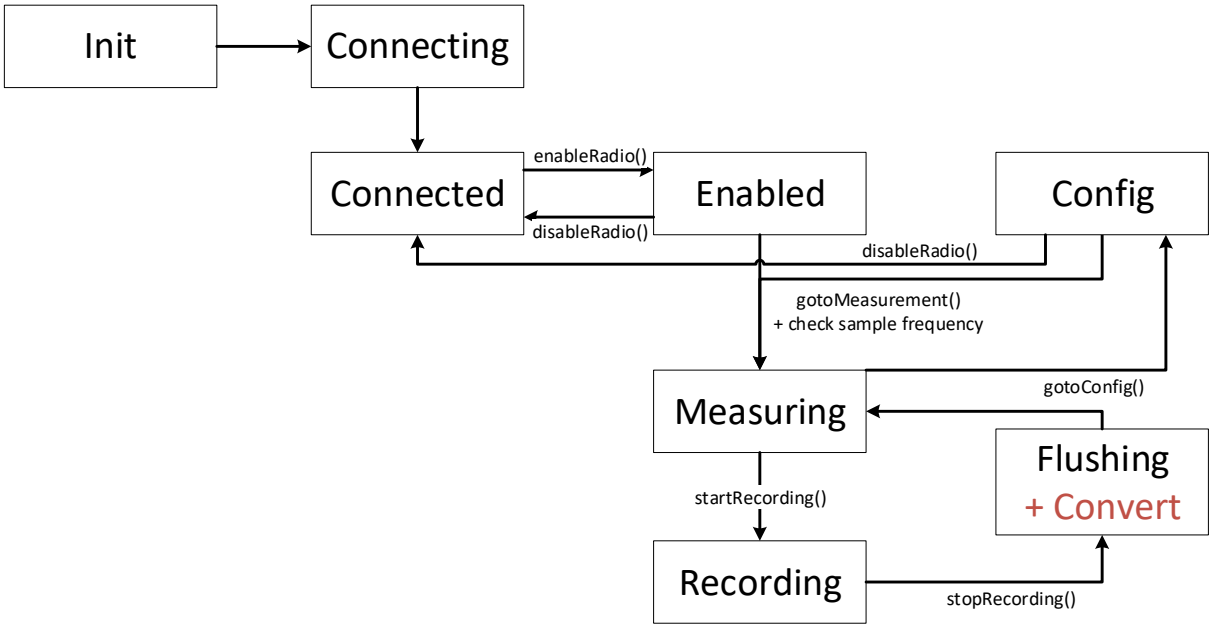
De parameters zijn:

- een pakket counter,
- de RSSI (Received Signal Strength Indicator),
- de accelerometer data (X, Y, Z waarde),
- de gyroscoop data (X, Y, Z waarde),
- de magnetometer data (X, Y, Z waarde).

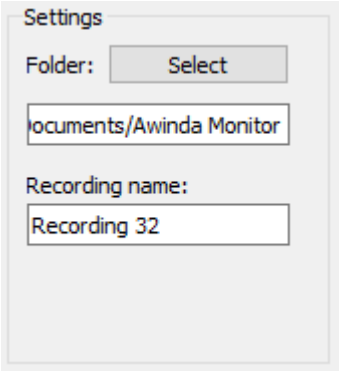
3.3.7 Recording opslaan als ander formaat

Aangezien het niet gewenst is dat de data opgeslagen wordt in het MTB-bestand zou het beter zijn dat deze onmiddellijk na het recorden opgeslagen wordt in de CSV-bestanden.

In de ‘flushing’ state worden de pakketten vanuit de buffers van de MTw nog verstuurd en opgeslagen in het log bestand. Hierna wordt het log bestand gesloten. Het is net in die ‘flushing’ state na het sluiten van het log bestand dat de data geconverteerd kan worden. In Figuur 3.10 is dit aangeduid in het rood.



Figuur 3.10: Aanpassing van de state machine voor het converteren naar een ander formaat



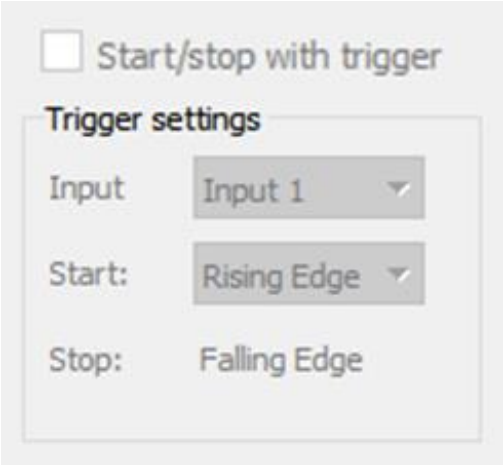
Figuur 3.11: Settings voor het opslaan bij het recorden

Opnieuw kan een folder pad geselecteerd worden met de hulp van een ‘Select’ knop. Aan die knop wordt opnieuw actie gelinkt waarbij er via een selectievenster een folder geselecteerd kan worden. Hierbij zal het tekstveld aangepast worden volgens de geselecteerde folder. Na het selecteren van de folder wordt in het geselecteerde pad gecheckt wat de laatste recording is. Als er in het pad bijvoorbeeld 31 recordings zijn zal de laatste recording folder, “Recording 31” zijn. Bij het checken van het pad, zal de het tekstveld automatisch naar de naam “Recording 32” veranderen. De naam van de recording kan ook manueel ingesteld worden als er een andere naam gewenst is. Dit is de foldernaam die aangemaakt zal worden om de CSV bestanden in op te slaan.

Om de conversie uit te voeren wordt het MTB-bestand opgeslagen in de aangemaakte folder. Nadat de ‘flushing’ gebeurd is en het log bestand gesloten is, wordt er soortgelijk tewerk gegaan als in sectie 3.3.6. Het log bestand wordt geladen in een nieuwe XDA en op dezelfde manier geconverteerd met dezelfde functies.

Na het recorden, flushen en converteren wordt in het geselecteerde pad opnieuw gezocht wat de laatste recording folder is. Op die manier zal de naam van de recording telkens met één verhogen zodat er na het recorden meteen een nieuwe recording kan gestart worden zonder een nieuwe naam te moeten invoeren.

3.3.8 Recorden via externe trigger



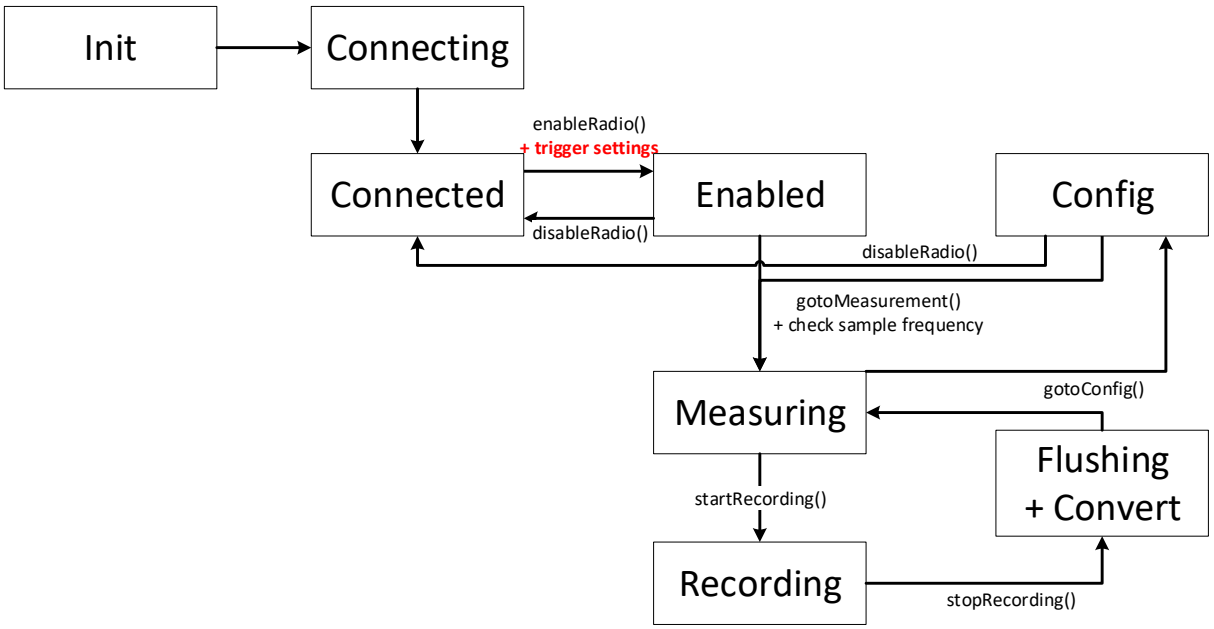
Figuur 3.12: Trigger settings

Om het systeem te laten synchroniseren met een ander systeem zijn er aan de master twee input connectoren voorzien. De master kan ingesteld worden om een actie uit te voeren als deze een trigger krijgt van een connector.

De externe trigger kan enkel ingesteld worden als de master in de ‘connected’ state zit. In die state van de GUI wordt het mogelijk gemaakt om de checkbox voor het starten en stoppen met een trigger aan te vinken. Bij het veranderen van de checkbox wordt een functie aangeroepen die de ‘Trigger settings’ activeert of deactiveert als de checkbox aan- of uitgevinkt is.

Bij de trigger settings (zie Figuur 3.12) kan een input geselecteerd worden, dit is één van de twee mogelijke inputs van de master. Er kan een op de stijgende, dalende of beide flanken gewerkt worden per instelling. Aangezien het enkel gewenst is om op één input te werken wordt er gewerkt met een flank voor het starten en een andere flank voor het stoppen. Voor het starten kan er gekozen worden om op de stijgende of dalende flank te werken. Als een flank geselecteerd wordt dan zal het label bij het stoppen automatisch overschakelen naar de tegenovergestelde flank. Hierdoor kan men direct zien bij het starten en stoppen welke flank er gebruikt wordt.

De trigger settings kunnen zoals reeds gezegd enkel ingeschakeld worden bij de ‘connected’ state van de master. Zoals in het rood afgebeeld op Figuur 3.13 wordt er voor gekozen om de trigger settings in te stellen net voor het enablen van de radio antenne. Dit is dus als men op de ‘Enable’-knop duwt.



Figuur 3.13: Aanpassing van de state machine voor het instellen van een trigger

Om de sync settings in te stellen moet er een ‘XsSyncSettingArray’ gemaakt worden waarin alle sync settings toegevoegd moeten worden. Als de checkbox niet aangevinkt is dan wordt een lege sync array toegevoegd aan de master. Hierdoor worden alle sync settings uit de master verwijderd. Als de checkbox wel aangevinkt is worden de sync settings ingesteld volgens de geselecteerde waarden in de trigger settings. Voor het starten en stoppen wordt er een setting aangemaakt met de geselecteerde input en de geselecteerde flank voor het starten alsook de andere flank voor het stoppen.

Naast de input en flank worden ook andere parameters ingesteld in de software, zoals hoe groot de puls breedte is, een offset om de actie uit te voeren, een optie om de actie maar één maal uit te voeren, … De puls breedte maakt niet zo veel uit, aangezien er enkel gewerkt wordt met een trigger op een flank. De andere parameters moeten allemaal op 0 geplaatst worden zodat de meting direct start en stopt bij de bijhorende trigger. De settings voor het stoppen en starten worden beiden toegevoegd aan de sync array waarna die array toegevoegd wordt aan de settings van de master.

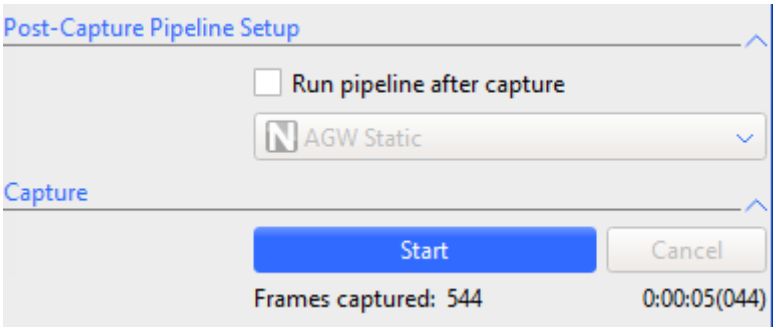
Het systeem wordt gesynchroniseerd met de Nexus Vicon. Dit is ook een systeem om bewegingen op te meten en analyseren. Op dit systeem kunnen pulsen uitgestuurd worden bij het starten en stoppen van een meting. Er wordt ingesteld dat er bij het starten van een meting, een stijgende flank verstuurd wordt. Bij het stoppen van de meting zal een dalende flank uitgestuurd worden. In onze applicatie wordt een stijgende flank voor het starten en

dalende flank voor het stoppen ingesteld. Als er op start recording geduwd wordt in onze applicatie dan zal deze wachten op een trigger om zo te starten met meten.

Deze synchronisatie werd ook uitvoerig getest. Aangezien het de bedoeling is dat de data van beiden op dezelfde sample frequentie werken werden de beide systemen op 100Hz ingesteld. Als de data van ons systeem (Figuur 3.14) vergeleken wordt met de recording van het Vicon systeem (Figuur 3.15) kan men zien dat het aantal pakketten/frames gelijk zijn. Met andere woorden was de meting dus volledig gelijklopend.

A	B	C	D	E
Identifier	Name	SampleFre	Packetcount	
B40EE8		100	544	
B41066		100	544	
B41067		100	544	
B4106D		100	544	
B41077		100	544	

Figuur 3.14: CSV van de gesynchroniseerde recording



Figuur 3.15: Gesynchroniseerde recording met de Nexus Vicon

3.4 Deployen van de applicatie

Bij het compileren van de source code worden alle libraries dynamisch gelinkt aan het project. Als het '.exe'-bestand uitgevoerd wordt zonder de Qt creator zullen die libraries niet meer dynamisch gelinkt worden. Hierdoor moeten alle gebruikte DLL's in de folder geplaatst worden van de executable. Dit zijn DLL's van zowel Qt als van de XSens API. Op de GitHub pagina worden al die DLL's meegegeven in een 'zip'-bestand. Het is belangrijk dat de structuur van de folder bewaard blijft als deze gekopieerd wordt naast de executable file.

3.5 MATLAB script

De geconverteerde CSV-bestanden moeten nog geïmporteerd worden naar MATLAB. Hiervoor worden de CSV-bestanden uitgelezen via een MATLAB script.

Een functie wordt hiervoor aangemaakt dat gebruikt kan worden in een ander MATLAB script. Als argument kan een folder meegegeven worden waar de functie zal zoeken achter data. Als er geen folder meegegeven is als argument zal er automatisch gevraagd worden om een folder te selecteren.

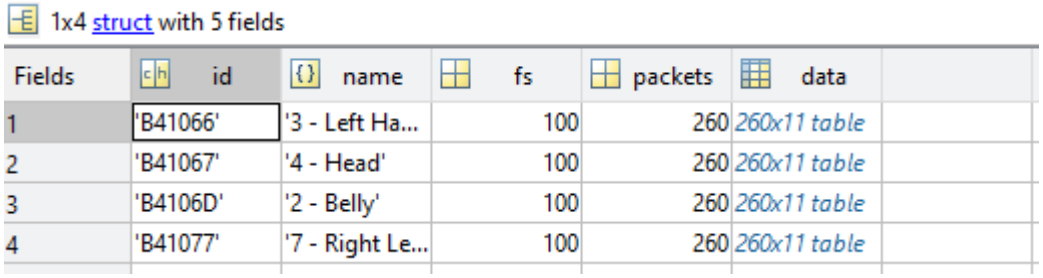
Het eerste dat het script doet is het uitlezen van de ‘deviceinfo.csv’. Hierin is een oplijsting van de eigenschappen van alle MTw’s die data verzameld hebben.

Die eigenschappen zijn:

- het device ID,
- de naam,
- de sample frequentie,
- het aantal verstuurde pakketten.

Alle MTw’s die in het ‘deviceinfo.csv’ bestand vermeld zijn hebben ook een CSV-bestand in de ‘/data’ folder. De naam van dat CSV-bestand komt overeen met het ID van MTw.

De data van de MTw’s worden een voor een uitgelezen en samen met de eigenschappen geplaatst als een veld van een MATLAB struct. Uiteindelijk wordt elke struct geplaatst in een array (zie Figuur 3.16). Die array wordt teruggegeven door de functie.


1x4 struct with 5 fields

Fields	id	name	fs	packets	data
1	'B41066'	'3 - Left Ha...	100	260	260x11 table
2	'B41067'	'4 - Head'	100	260	260x11 table
3	'B4106D'	'2 - Belly'	100	260	260x11 table
4	'B41077'	'7 - Right Le...	100	260	260x11 table

Figuur 3.16: MATLAB struct array

Door die functie te gebruiken in een eigen MATLAB script, kan de data van elk device gebruikt worden en kan men ook direct zien wat de eigenschappen van MTw’s zijn. Het dataformaat komt overeen met hoe het vroegere MATLAB script de data uitpakte. Hierdoor kunnen scripts de geschreven zijn aan de hand van dergelijk formaat nog altijd gebruikt worden.

4 CONCLUSIE

Alle problemen die er waren met de MT Manager zijn opgelost in de nieuwe GUI.

- Het probleem met betrekking tot het gegenereerde MTB-bestand is opgelost door een conversie toe te voegen in de GUI waarmee een MTB-bestand geconverteerd wordt in CSV-bestanden. Ook na het recorden wordt het MTB-bestand automatisch geconverteerd naar CSV bestanden waardoor er geen MTB-bestand meer uitgelezen moet worden in MATLAB. In de plaats daarvan is er een MATLAB script voorzien die de data van de CSV-bestanden inleest. Die data kan meteen gebruikt worden voor verwerking en analyse.

- De sensoren verliezen soms de connectie met de master doordat de batterij leeg is. Dit probleem is opgelost door telkens er zich een error voordoet met een van de MTw's (verliezen van connectie) die error in een venster te plaatsen waar alle belangrijke berichten in komen. Hierdoor weet de gebruiker dat de batterij van één van de sensoren leeg is en dat de meting gestopt moet worden. Het batterijlevel van elke geconnecteerde MTw wordt ook opgevraagd. Via een kleurenindicatie kan waargenomen worden als de batterij bijna leeg is.

- De status van het recorden kan bijgehouden worden via een statusvenster waarin de state van de master weergegeven wordt, samen met een timer voor het recorden en het aantal geconnecteerde MTw's.

- Een check wordt uitgevoerd op de sample frequentie. Door via de master te kijken welke MTw's verbonden zijn en bij elke MTw de sample frequentie te checken kan men er nu zeker van zijn dat alle MTw's met de juiste sample frequentie opmeten.

Een dergelijke applicatie maken is niet altijd evident. Aangezien er verder gewerkt werd met een sample applicatie moet men eerst die volledige applicatie begrijpen worden. Anders wordt het moeilijk om er op verder te bouwen. De structuur die gebruikt wordt in de sample applicatie is bijgevolg ook de structuur die gebruikt moet worden bij het verder ontwikkelen van de applicatie. Als de structuur niet gevolgd wordt zou de applicatie weer van begin af aan opgebouwd moeten worden. Als er niet verder gewerkt was geweest op de sample applicatie gingen er nooit zoveel functionaliteiten inbegrepen zijn tegen de einddatum.

Omdat veel van de functionaliteiten interactie vereisen met de Awinda master en MTw's moeten die ook telkens getest worden met de hardware zelf. Vaak werden er thuis dan ook functies geschreven. Die werden dan vervolgens getest werden in de labo sessies of buiten de lesuren in het bewegingslabo. Het is enkel door deze in de praktijk te testen en debuggen dat men er zeker van kan zijn dat de functionaliteiten werken. De functionaliteiten werden dan ook uitvoerig getest in het bewegingslabo.

Het ontwikkelen van een grafische user interface in een onbekende tool kan soms frustrerend zijn. Er werd in het project bijvoorbeeld begonnen in C# waarna na een tijdje problemen de kop op staken. Hierdoor was het reeds verwezenlijkte werk in C# waardeloos. Ook bij het importeren van de MATLAB library in het project waren er veel problemen om deze te importeren. Als deze uiteindelijk geïmporteerd werden, werd pas duidelijk dat de volledige library moest gebruikt worden in het project waardoor ook al dat werk ook niet meer nuttig was.

Desalniettemin heb ik in dit project veel geleerd over het ontwikkelen van software via een SDK. Vooral over Qt heb ik veel geleerd. Die kennis kan zeker ook nog van pas komen in het latere bedrijfsleven aangezien dit in de industrie vaak gebruikt wordt om c++ applicaties te ontwikkelen en GUI’s te maken.

5 FUTURE WORK

Vaak moeten verschillende recordings na elkaar gebeuren via het Vicon Nexus systeem. De GUI gemaakt met dit project zorgt er voor dat de master ingesteld kan worden om met dat systeem te synchroniseren. Er moet telkens op ‘Start recording’ geduwd worden om te kunnen starten met meten. Hierbij wacht deze dan op een trigger van de Vicon om te starten met opnemen. Als de recording gestopt wordt met een dalende flank moet er opnieuw geduwd worden op de knop om terug te starten met meten. Het zou dus handig zijn als er een manier was waarbij er na het stoppen van de recording via een dalende flank, direct gewacht wordt op een stijgende flank om weer op te nemen. Op die manier zou men enkel nog op de gebruikersinterface van de Vicon Nexus moeten letten en zal de data van de Awinda automatisch opgeslagen worden in nieuwe folders. Dit was een idee dat opgekomen was tijdens het testen van de synchronisatie, de laatste week van het project. Een implementatie hiervan is reeds aangebracht in de GUI en source code. Die implementatie is echter niet meer getest geweest.

De verschillende velden met betrekking tot de trigger en de naam van de map waar de geconverteerde data opgeslagen moet worden zou bij elk project eventueel hetzelfde kunnen zijn. Een mogelijkheid om alle verschillende velden van de GUI op te slaan en te laden bij het openen van de GUI zou mogelijks een extra feature kunnen zijn van deze applicatie.

Referenties

[1] Xsens Technologies B.V., “MTw User Manual,” no. November, pp. 1–86, 2013.

[2] Xsens Technologies B.V., “XDA Documentation,” 2016.

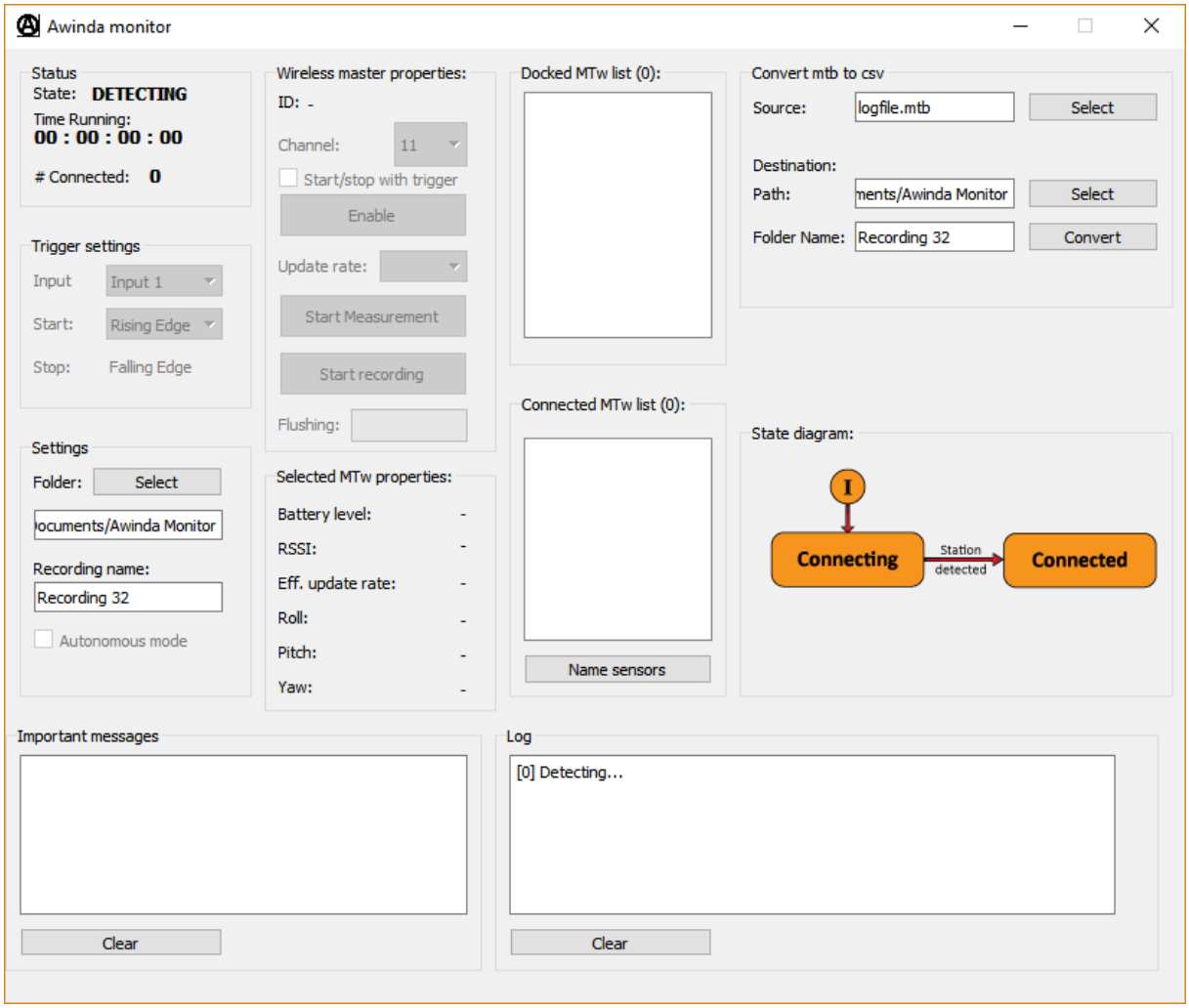
[3] Qt, “Qt | Cross-platform software development for embedded & desktop,” 2016. [Online]. Available: https://www.qt.io/. [Accessed: 28-May-2018].

[4] “MAT-File Library and Include Files - MATLAB & Simulink.” [Online]. Available: https://www.mathworks.com/help/matlab/matlab_external/mat-file-li

[5] Xsens, “MTw Awinda - Products - Xsens 3D motion tracking,” 2016. [Online]. Available: https://www.xsens.com/products/mtw-awinda/. [Accessed: 25-May-2018

[6] “Xsens : 3D Motion Tracking - Xsens Online Shop,” *Xsens.Com*. [Online]. Available: https://shop.xsens.com/shop/xsens-mvn/mtw-awinda-wireless-3dof-

Bijlage A AFGEWERKTE GUI



Figuur A.1: Volledig afgewerkte GUI

FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN
CAMPUS BRUGGE
Spoorwegstraat 12
8200 BRUGGE, België
tel. + 32 50 66 48 00
iiw.brugge@kuleuven.be
www.iiw.kuleuven.be