buddyAssist

Onderwater Navigatie- en Opsporingssysteem

versie 0.9, Augustus 2016

J. Tigchelaar

Hogeschool Rotterdam

SCUBAssist

buddyAssist

Onderwater Navigatie- en Opsporingssysteem

versie 0.9, Augustus 2016

door

J. Tigchelaar

in overeenstemming met de vereisten voor het verkrijgen van de titel van

Bachelor
of Information and Communication Technology

aan de Hogeschool Rotterdam,
te verdedigen op maandag 29 augustus 2016 om 16:00 uur.

Student nummer: 0529543
Supervisor: ir. T. Ubert
ir. G. Maggiore

Op dit verslag is geheimhouding van toepassing tot en met 31 december 2016.

Een elektronische versie van dit verslag is beschikbaar op
http://www.scubassist.com/scriptie.
Het project is beschikbaar op https://github.com/scubAssist/.

Voorwoord

Dit project heeft, inclusief het schrijven van deze scriptie, een groot deel van mijn leven beheerst de afgelopen jaren. En ik heb genoten van (bijna) elk moment.

Graag bedank ik iedereen die meegedacht heeft en me op alle mogelijke manieren hebben ondersteund. In het bijzonder bedank ik mijn vrouw Jolanda, voor haar eindeloze geduld tijdens het aanhoren van idëeen en voor het meedenken en helpen tijdens het realiseren van de prototypes.

Mijn familie wil ik bedanken voor de steun en aanmoediging. Niet alleen voor het project, maar ook tijdens de andere aspecten van de studie. Ik ben blij en trots dat ik aan mijn ouders kan laten zien dat ik, zij het met enige vertraging, toch een HBO studie heb kunnen afronden.

Daarnaast wil ik mijn studiegenoten Sietse, Mark en William bedanken voor de onderlinge steun en motivatie. Mijn collega 's Maarten Hersbach en Willem de Voogd wil ik bedanken voor hun hulp tijdens het schrijven van deze scriptie en bij het realiseren van het project.

Speciale dank gaat uit naar mijn goede vriend Barry Breur, bedankt voor de mooie afbeeldingen en het vertrouwen in het project.

Een woord van dank richting mijn werkgever Voogd & Voogd kan zeker niet uitblijven, het feit dat ze mij toegestaan hebben de buddyAssist naast mijn werk te realiseren in mijn eigen bedrijf is iets unieks, waarvoor ik erg dankbaar ben. Met name mijn manager John Borsboom verdient een woord van dank voor zijn interesse, hulp en begeleiding.

J. Tigchelaar
Stellendam, juli 2016

Abstract

Uittreksel

Lange tijd is er geen echte innovatie geweest op het gebied van (sport)duiken. Het is een veilige sport, maar als er iets misgaat, is het vaak direct dodelijk omdat het moeilijk is een duiker te vinden en hulp te bieden. Daarnaast hebben duikers, sinds het ontstaan van de aqua long in 1952 door Jacques Yves Cousteau, weinig echte innovaties meegemaakt. De apparatuur is kleiner en gebruiksvriendelijker geworden, maar de hoofdzaken zijn vrijwel gelijk gebleven.

Na een aantal analoge voorlopers kwam er in 1983 (*Entwicklung Tauchcomputer*, 2012) de eerste digitale duikcomputer. Duiken werd plotsklaps veel veiliger omdat mensen geholpen werden bij het bepalen van de decompressie tijden en limieten. Innovatie van dit niveau is daarna vrijwel uitgebleven.

Dit document beschrijft een systeem wat eenzelfde uitwerking kan hebben op de veiligheid van sport- en beroeps duikers als de introductie van de duikcomputer. Het systeem stelt duikers in staat om in 3 dimensies hun positie onderwater te bepalen en het stelt hulpverleners in staat om de exacte positie van een duiker onder water te zien in het geval van een noodsituatie.

Abstract

For a long time there has been no real innovation regarding (scuba)diving. The sport is relatively safe, but when an accident does occur it tends to be fatal due to the difficulty in locating and assisting divers. Besides this, diving has had no real innovation after Jacques Yves Cousteau invented the aqualung in 1952. The equipment has become smaller and more user-friendly, but by and large it is the same equipment since 1952.

After some analog predecessors the first digital dive computer was developed in 1983 (*Entwicklung Tauchcomputer*, 2012). Immediately diving became safer because divers were being assisted in dealing with decompression times and limits. Innovation of this level has not happened since.

This document describes a system that can have the same effect on the safety of divers as the invention of the dive computer. It allows divers to see their location in three dimensions while also allowing first responders to see where a diver is located in the event of an emergency.

Inhoudsopgave

1 Inleiding **1**
- 1.1 Ontstaan van het idee . . . 1
- 1.2 Onderzoeksvraag . . . 2
  - 1.2.1 Deelvragen . . . 2
- 1.3 Gebruikte methoden . . . 2
  - 1.3.1 Kwalitatief . . . 2
  - 1.3.2 Kwantitatief . . . 2
- 1.4 Stakeholders . . . 2

2 Planning en project methodiek **5**
- 2.1 Project methodiek . . . 5
- 2.2 Planning . . . 6
  - 2.2.1 Sprint 1 (WEEK 1 – WEEK 3) . . . 6
  - 2.2.2 Sprint 2 (WEEK 4 – WEEK 5) . . . 6
  - 2.2.3 Sprint 3 (WEEK 6 – WEEK 8) . . . 6
  - 2.2.4 Sprint 4 (WEEK 9 – WEEK 11) . . . 6
  - 2.2.5 Sprint 5 (WEEK 12 – WEEK 14) . . . 6
  - 2.2.6 Sprint 6 (WEEK 15 – WEEK 16) . . . 6
  - 2.2.7 Sprint 7 (WEEK 17 – WEEK 18) . . . 7
  - 2.2.8 Sprint 8 (WEEK 19 – WEEK 20) . . . 7

3 Onderzoek **9**
- 3.1 Budget bepaling . . . 9
- 3.2 Bestaande technieken en realiteit van het positiebepaling concept . . . 9
- 3.3 Betaalbaar akoestisch modem . . . 11
- 3.4 Overige hardware . . . 12
- 3.5 Backend . . . 12
- 3.6 Wetgeving . . . 12

4 Uitwerking **13**
- 4.1 Het globale systeem . . . 14
  - 4.1.1 Duiker systeem . . . 14
  - 4.1.2 Het boei systeem . . . 15
  - 4.1.3 Het centrale systeem . . . 16
  - 4.1.4 Het ad hoc systeem . . . 16
- 4.2 Systeem architectuur . . . 16
  - 4.2.1 REST API . . . 16
  - 4.2.2 Flightradar en Monitoring Website . . . 18
  - 4.2.3 Community Website . . . 18
  - 4.2.4 Security . . . 18
- 4.3 Positiebepaling . . . 18
  - 4.3.1 TULIP . . . 18
  - 4.3.2 Box-grid schatting . . . 20
  - 4.3.3 BAPE algoritme . . . 20
  - 4.3.4 Resultaten en vergelijking . . . 21
  - 4.3.5 Praktisch . . . 23
- 4.4 Modem . . . 25
  - 4.4.1 Ultrasoon circuit . . . 26
  - 4.4.2 Protocol . . . 29

5 Conclusie en discussie **31**
5.1 Conclusie . . . 31
5.2 Discussie . . . 32

6 Vervolg **33**
6.1 Doorontwikkeling . . . 33
6.2 Commerciële exploitatie . . . 33

7 Reflectie **35**
7.1 Reflectie . . . 35

Bibliografie **37**

A Appendix Visuals **39**

B Appendix Hardware Schema's **43**

C Appendix Code **45**
C.0.1 Modem . . . 45
C.0.2 Simulatie . . . 47

Afkortingen

BAPE Box-grid Accoustic Position Estimation. 20, 21, 22, 25, 47

CHRRB Customisable Hydrophonic Rescue and Recovery Badge. 12, 26

GPS Global Positioning System. 9, 15, 25

HRO Hogeschool Rotterdam en Omstreken. 12

MIT Massachusetts Institute of Technology. xi, 18

OOK On-Off Keying. 26

POI Points Of Interest. 15

RDM De Rotterdamsche Droogdok Maatschappij. 12, 28

RTC Real Time Clock. 23, 24

WHOI Woods Hole Oceanographic Institution. 25

Begrippenlijst

Arduino Arduino is een opensource-computerplatform dat is opgebouwd rond de ATmega168 microcontroller van Atmel en het software ontwikkelplatform Processing. Dit platform is bedoeld voor hobbyisten, artiesten, kunstenaars en iedereen die geïnteresseerd is in het maken en ontwerpen van slimme en creatieve objecten die kunnen reageren op hun omgeving. 15, 25, 26, 45

Open Source beschrijft de praktijk die in productie en ontwikkeling vrije toegang geeft tot de bronmaterialen (de source) van het eindproduct. 10, 11, 12, 15, 25, 31, 32, 33, 36

REST REST staat voor REpresentational State Transfer en is een software architectuur waarbij de nadruk ligt op het manipuleren van objecten en niet op implementatie details. [1]. 16, 17, 18

RF Radiogolven, ook radiofrequente straling genoemd, zijn golven in de vorm van elektromagnetische straling met golflengten uiteenlopend van ruwweg duizend kilometer tot een millimeter, dus in het frequentiegebied van enkele honderden Hz tot enkele honderden GHz, die gebruikt worden in de communicatietechniek om informatie over te brengen van een radiozender naar een of meer ontvangers.. 12, 13, 14, 22, 23, 24, 26

triangulatie Een techniek waarbij de positie bepaald wordt op basis van de gemeten afstand tot 3 vaste objecten. xi, 18

TULIP Een triangulatie-algoritme ontwikkel door het Massachusetts Institute of Technology (MIT). 10, 18, 19, 20, 21, 22

WiFi Wi-Fi is een certificatielabel ('logo') voor producten voor draadloze datanetwerken, die werken volgens de internationale standaard *IEEE 802.11* (draadloos ethernet of wifi). Producten die volgens deze standaard werken maken gebruik van radiofrequenties in de 2,4GHz- en/of 5,0GHz-band die onder voorwaarden zonder licentie gebruikt mogen worden.. 13, 14, 15, 25

[1] https://en.wikipedia.org/wiki/Representational_state_transfer

1

Inleiding

In dit document worden de concepten en de uiteindelijke implementatie van het *'buddyAssist'* systeem uiteengezet. Dit document is bedoeld als afstudeerscriptie, maar ook als achtergrond voor mensen die meer informatie willen over het ontstaan van het systeem en de concrete implementatie.

1.1. Ontstaan van het idee

Ik ben zelf woonachtig in Stellendam, een klein dorpje net boven Zeeland en gelegen op het eerste eiland grenzend aan de Grevelingen. In de zomer is dit het Mekka voor duikers, uit heel Europa komen duikers naar de dorpen rondom de Grevelingen en de Zeeuwse wateren.

Duiken is in de basis een veilige sport, de marge voor fouten is echter erg klein en ogenschijnlijke kleine gezondheidsproblemen kunnen onderwater snel tot gevaarlijke situaties leiden.

Elk jaar zijn er dan ook weer enkele ongelukken te betreuren, vaak met dodelijke afloop. Het overlijden van de duikers is erg, het feit dat nabestaanden vaak weken moeten wachten voor het lichaam gevonden is, maakt de situatie nog erger.

Ik ben gaan zoeken naar systemen die het terugvinden van duikers die onderwater onwel worden eenvoudiger maakt. Er zijn een aantal systemen, voornamelijk gericht op het uit kunnen peilen van verdronken personen. [1]

Het kunnen opsporen van reeds verdronken personen is vooral vanuit de (overheids) instanties die nu betrokken zijn bij het opsporen van deze mensen een behoefte. Ik wil zelf proberen te voorkomen dat iemand verdrinkt, of dan toch een zeer snelle hulpverlenings-reactie bewerkstelligen.

De bestaande systemen zijn moeilijk aan de man te brengen omdat ze geen direct voordeel bieden aan de duikers en toch vrij prijzig kunnen zijn. Veel duikers willen direct nut hebben van een systeem en vinden een systeem wat alleen toegevoegde waarde heeft voor het terugvinden na overlijden niet een grote investering waard.

Omdat ik zelf ook duik ben ik gaan nadenken over een systeem wat toegevoegde waarde heeft voor de duiker *en* kan dienen als opsporingssysteem. Na veel onderzoek, overleg en brainstormen is het idee ontstaan om een onderwater **positie bepaling systeem** te ontwikkelen, waarbij de positie van de duikers ook terug gecommuniceerd wordt naar een centraal systeem om zo bij calamiteiten een duiker snel te kunnen terugvinden.

Het voordeel van dit systeem is dat het direct toegevoegde waarde biedt aan de gebruiker en dus interessanter is om aan te schaffen, dan alleen een systeem waarmee een duiker teruggevonden kan worden. Het systeem wat in dit document besproken wordt zou de bestaande duikcomputers geheel kunnen vervangen en biedt dezelfde functionaliteiten met daaraan toegevoegd positie bepaling, dus eigenlijk een 'onderwater gps' systeem.

[1] CHRRB: Verdronken personen detectie

1.2. Onderzoeksvraag

De hoofdvraag van dit project is: *Is het mogelijk om voor een realistisch bedrag een onderwater positiebepaling systeem te ontwikkelen waar sportduikers gebruik van willen maken?*

1.2.1. Deelvragen

- Wat is een realistisch bedrag voor de gemiddelde sportduiker?
- Welke technieken zijn reeds beschikbaar?
- Kunnen we betaalbare communicatie hard- en software ontwikkelen?
- Werkt het positiebepaling concept?
- Is bi-directionele communicatie mogelijk?
- Hebben we te maken met conflicterende wetgeving?
- Welke opslag- en verwerkingstechnieken kunnen we gebruiken in het backend platform?
- Hoe zetten we het systeem in de markt?
- Hoe zorgen we voor een solide bedrijfsmodel?
- Willen duikers gebruik maken van het systeem?

1.3. Gebruikte methoden

Het project zelf mag betiteld worden als een exploratief onderzoek. Grote delen van het systeem bestaan nog niet en dit onderzoek zal antwoord moeten geven op de diverse vraagstukken om het systeem te laten werken. In het onderzoek is gebruik gemaakt van zowel kwantitatieve als kwalitatieve methoden.

1.3.1. Kwalitatief

- *Literatuuronderzoek*: Ik maak gebruik van diverse onderzoeken en rapporten van universiteiten over heel de wereld. Daarnaast raadpleeg ik veel bronnen over de bestaande hard- en software technieken op het gebied van onderwater communicatie. Tevens heb ik contact gehad met een aantal bedrijven en instanties waarbij ik inzage heb gekregen in de gebruikte hard- en software.

1.3.2. Kwantitatief

- *Secundaire analyse*: Ik heb om de validiteit van het concept te bepalen gebruik gemaakt van de beschikbare data uit de papers die gebruikt zijn tijdens de ontwikkeling.
- *Experimenteel onderzoek*: Tijdens het gehele ontwikkeltraject word de hard- en software experimenteel getest. De resultaten hiervan worden gebruikt om het systeem te valideren en te verfijnen.

1.4. Stakeholders

De afstudeer situatie is enigszins vreemd daar ik afstudeer in mijn eigen bedrijf. Zoals aangegeven in mijn mandaat zijn er genoeg stakeholders om toch te kunnen voldoen aan de eisen die aan een afstudeeropdracht gesteld worden. Daarnaast heb ik mijn manager bij mijn werkgever Voogd & Voogd bereid gevonden om als stagebegeleider te fungeren zodat ook deze rol is ingevuld.

Sinds het starten van dit project ben ik gaan samenwerken met een ander project binnen de Hogeschool Rotterdam, te weten het CHRRB: Verdronken Personen Detectie project. Er is veel overlap tussen de technische behoefte van beide projecten en een samenwerking was daarom voor de hand liggend.

Deze samenwerking heeft ook gezorgd voor een toename in het aantal stakeholders. Voor dit project heb ik echter alleen de directe stakeholders van het buddyAssist systeem genoemd.

KLPD dienst opsporing drenkelingen	Jitteke Struik
KNRM	Ad van der Plas
Nederlandse onderwatersportbond	Ron Lemmens
	Desmond van Santen
Kenniscentrum RDM	Dr. ir. Ing. Kees Pieters
Rijkswaterstaat	Marcel Hintzen
	Gerius van Woudenberg
Gemeente Schouwen-Duiveland	Marcel van den Berge

2

Planning en project methodiek

Omdat het een vrij omvangrijk project betreft, was een goede planning noodzakelijk en een project methodiek die zich leent voor snel veranderende inzichten. De auteur heeft ruime praktijk ervaring met Agile ontwikkel methodieken, de keuze om dit project in te delen in diverse sprints volgens de Agile SCRUM methodiek, was dan ook voor de hand liggend.

2.1. Project methodiek

Zoals aangegeven was het, voordat in ernst aan het project begonnen werd, reeds duidelijk dat de inzichten gedurende het proces zouden wijzigen. Een planning en project methodiek die daar mee om konden gaan was dan ook erg belangrijk. Door eerdere ervaringen met Agile SCRUM en de daarbij behorende manier van werken is de keuze hierop gevallen.

Aan het begin van het project is er gekeken naar de zaken die onderzocht moesten worden en deze zijn in zogeheten *timeboxes* ingedeeld. Daarna is er een aantal sprints benoemd met de daarbij behorende artefacten die opgeleverd moesten worden.

Figuur 2.1: Scrum board bij het Voogd & Voogd InnovatieLab

2.2. Planning

Onderstaand de planning die opgezet is aan het begin van het project. Het project is opgedeeld in 8 sprints van 2 weken. Binnen elk van die sprints wordt een deel van het onderzoek uitgevoerd en wordt direct een deel van de scriptie gevuld. De planning was om medio juni 2016 klaar te zijn, de geboorte van de dochter van de auteur kwam onverwacht 4 weken eerder dan gepland en heeft gezorgd voor het tijdelijk opschorten van de werkzaamheden aan de scriptie. Gelukkig is het mogelijk gebleken dit schooljaar toch het project af te ronden.

2.2.1. Sprint 1 (WEEK 1 – WEEK 3)

In deze iteratie wordt het vooronderzoek wat reeds gedaan is, uitgewerkt in een verslag. Hierin wordt aangegeven welke systemen er reeds bestaan, waarom ze niet voldoen en welke elementen we specifiek gaan onderzoeken.

Artefact: Eerste deel van het onderzoeksverslag met afbakening

2.2.2. Sprint 2 (WEEK 4 – WEEK 5)

In deze iteratie word de basis van de techniek uitgewerkt in een simulatie model. Het gaat dan om de positiebepaling algoritmes en de te verwachten afwijkingen en oplossingen daarvoor.

Artefact: Tweede deel van het onderzoeksverslag met de resultaten en aanbevelingen uit de simulaties.

2.2.3. Sprint 3 (WEEK 6 – WEEK 8)

In deze iteratie wordt het basisdeel van de te ontwikkelen hardware gerealiseerd. Alhoewel het niet nodig is voor de scriptie, is dit een essentiëel onderdeel van de rest van het onderzoek. Het belang is om de simulaties te kunnen staven aan de werkelijkheid en eventueel nog correcties door te voeren.

Artefact: Derde deel van het onderzoeksverslag met de praktijk metingen. Prototype van het onderwater modem met beperkte capaciteit.

2.2.4. Sprint 4 (WEEK 9 – WEEK 11)

In deze iteratie wordt de communicatie tussen de verschillende hardware componenten ontwikkeld. Het gaat dan om de ontwikkeling en het testen van het onderwater protocollen en de synchronisatie van de duiker en het systeem boven water.

Artefact: Uitbreiding van het onderzoeksverslag met de uitwerking van de protocollen.

2.2.5. Sprint 5 (WEEK 12 – WEEK 14)

Het uitvoeren van de synchronisatie en de daaropvolgende tests om de positiebepaling in een echte situatie uit te voeren. Tevens bepalen welke data vanaf de duikers aangeboden gaat worden bij de boeien.

Artefact: Uitbreiding van het onderzoeksverslag met de uitslagen van de echte positiebepaling testen.

2.2.6. Sprint 6 (WEEK 15 – WEEK 16)

Onderzoeken van de opslagmogelijkheden van de data die door het systeem gegenereerd wordt. Tevens kijken naar het platform om de data beschikbaar te stellen aan de eindgebruikers en geïnteresseerde partijen.

Artefact: Uitbreiding van het onderzoeksverslag met de resultaten van het onderzoek naar het backend platform.

2.2.7. Sprint 7 (WEEK 17 – WEEK 18)
Het opstellen van een business model waarmee het systeem in de markt geplaatst kan worden.

Artefact: Uitbreiding van het onderzoeksverslag met het Businessmodel, voorzien van business model canvas.

2.2.8. Sprint 8 (WEEK 19 – WEEK 20)
Afronden van het onderzoeksverslag en voorbereiden op de verdediging. Scriptie laten drukken als boek.

Artefact: Presentatie en gedrukte scriptie

3

Onderzoek

Voor dit project zijn diverse kleine en grote onderzoeken uitgevoerd via de eerder genoemde onderzoeksmethoden. In dit hoofdstuk worden de onderzoeken en de resultaten daarvan doorlopen. In de rest van het verslag wordt gerefereerd aan de in dit hoofdstuk beschreven onderzoeken en uitkomsten.

3.1. Budget bepaling

Een belangrijke vraag voor dit gehele project was het bepalen van de maximale kosten van het systeem. Het systeem kan perfect functioneren, maar dermate duur uitvallen dat er geen markt voor te realiseren is. Om in de rest van de onderzoeken sturing te kunnen geven aan de te gebruiken oplossingen en componenten is er eerst een opsomming gemaakt van de meest populaire, op dit moment beschikbare, duikcomputers en de gemiddelde prijs daarvan. Het buddyAssist systeem biedt veel meer functionaliteiten dan een duikcomputer, maar een beter referentiepunt is niet beschikbaar.

Omschrijving	Url	Prijs
iGills iPhone duikcomputer	http://www.igills.com/igills-5x-housing-for-iphone-5s-5c-and-5/	€ 269,00
DivePhone	http://innovasub.com/pages/DivePhone.aspx	€ 390,00
Suunto DX Black Titanium + USB	http://www.lucasdivestore.com/nl/divegear/comp-watches.html	€ 1.199,00
Scubapro Mantis 2.0 With Transmitter	http://www.lucasdivestore.com/nl/divegear/comp-watches.html	€ 1.099,00
Suunto EON STEEL	http://www.lucasdivestore.com/nl/divegear/comp-watches.html	€ 899,00
Shearwater Perdix	http://www.lucasdivestore.com/nl/divegear/comp-watches.html	€ 875,00
Aqualung i750T Color wrist dive computer w/usb	http://www.lucasdivestore.com/nl/divegear/comp-watches.html	€ 799,00
Mares Icon HD 4.0 Black Edition + Transmitter	http://www.lucasdivestore.com/nl/divegear/comp-watches.html	€ 699,00
Mares Icon HD 4.0 Black Edition	http://www.lucasdivestore.com/nl/divegear/comp-watches.html	€ 499,00
Mares Nemo Wide	http://www.lucasdivestore.com/nl/divegear/comp-watches.html	€ 269,00
Suunto Helo2 + USB	http://www.lucasdivestore.com/nl/divegear/comp-watches.html	€ 459,00
	Gemiddeld	€ 745,60

Figuur 3.1: Overzicht prijzen populaire duikcomputers

In bovenstaande tabel is een opsomming van de systemen opgenomen, daar kwam een gemiddelde prijs uit van **745 Euro**.

3.2. Bestaande technieken en realiteit van het positiebepaling concept

Na het bepalen van een mogelijk budget was het van belang om vast te stellen hoe we de positie willen bepalen. Hierbij is voornamelijk gekeken naar het onderzoek van Wouter A.P. van Kleunen op de Universiteit Twente. In zijn paper (van Kleunen, 2014) werkt hij in hoofdstuk 5 en 6 de theorie en praktijk uit van een aantal methoden om onderwater de positie van een 'blind node' te bepalen. In ons geval is de blind node de duiker waarvan de positie onbekend is.

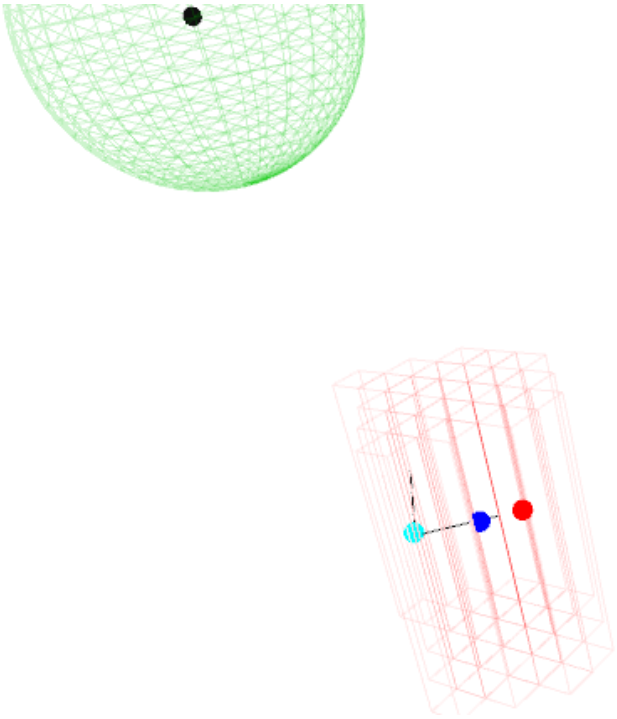
Het idee achter deze methode van positiebepaling is vergelijkbaar met die van Global Positioning System (GPS): Er zijn satellieten die radiosignalen uitsturen. Een GPS ontvanger ontvangt deze signalen en kan dan na het ontvangen van een aantal signalen de eigen positie bepalen ten opzichte van deze satellieten. Onder water is het principe gelijk, echter worden de radiosignalen vervangen door geluidssignalen, omdat radiosignalen onder water geen groot bereik hebben, geluidssignalen doen het onderwater juist weer beter dan boven water. De satellieten worden vervangen door boeien welke op een vaste positie verankerd worden. Hiermee is de werking van het systeem dus goed te vergelijken met die van het bestaande GPS systeem.
Op basis van de paper (van Kleunen, 2014) is er contact gezocht met Wouter en heeft er een bezoek plaatsgevonden op de Universiteit Twente, daar is met de *'Pervasive Systems'* groep, waar Wouter deel van uitmaakt, gesproken over de mogelijkheden om zijn onderzoek om te zetten naar een praktisch inzetbaar systeem. Het was een leerzame periode, maar helaas is de samenwerking met de Universiteit Twente na enkele maanden stil gevallen. De door hen gebruikte systemen waren erg duur en er was geen academische reden om dit verder te ontwikkelen.

Dat betekende zeker niet het einde van het project, de gesprekken die gevoerd zijn bevestigden de validiteit van het idee. De concrete uitvoering moest echter wel anders ingevuld worden om binnen het gestelde budget te kunnen blijven. De *Seastar* akoestische modems die door de Universiteit Twente gebruikt worden overschrijden dat budget (zie 3.1) al met enkele factoren. Daarom is besloten om een Open Source akoestisch onderwater modem te ontwikkelen of te vinden. Hiervoor is een apart onderzoek uitgevoerd (zie 3.3).

Er vanuit gaande dat dit modem gerealiseerd kon worden, is er een onderzoek gedaan naar de concrete implementatie van de positie bepaling. Zoals ook beschreven in de eerder genoemde paper (van Kleunen, 2014, p 78-88) is het van belang de afstand te bepalen vanaf een aantal bekende punten om zo de positie van het onbekende punt te kunnen bepalen. Dit is zeker geen nieuwe techniek en een zeer mooie en robuuste uitwerking hiervan is te vinden in het TULIP Algoritme (Cottrell, 2012). Dit algoritme is in staat om zonder exact kruispunt van de gemeten afstanden toch een positie te bepalen. Later is er een ander algoritme geïmplementeerd, maar het vinden en testen van TULIP bevestigde de haalbaarheid van het concept.

De concepten van het systeem zijn geïmplementeerd in een simulatie programma[1] waarbij verschillende technieken en variabelen kunnen worden toegepast om de technieken te testen. De uitvoer van deze tests is te vinden in hoofdstuk 4.3.4.

[1] De simulatie software is te vinden op http://www.github.com/scubAssist/



Figuur 3.2: Simulatie van de positiebepaling systemen

De basisingrediënten voor de positiebepaling waren nu aanwezig. In hoofdstuk 4.3 wordt concreet invulling gegeven aan de gebruikte methodieken.

3.3. Betaalbaar akoestisch modem

Omdat er al enig vooronderzoek gedaan was en vrijwel elke toepassing gebruik maakt van ultrasone communicatie was het duidelijk dat dit de meest optimale methode moest zijn. Er is wel een onderzoek gedaan naar alternatieve methoden, echter werd in het literatuuronderzoek alleen maar bevestigd dat ultrasoon geluid de meest optimale techniek was (Lanbo, Shengli & Jun-Hong, 2008).

In de eerste opzet zou er gebruik gemaakt worden van een bestaand modem, maar er bleek geen commerciëel beschikbaar modem te bestaan in een prijsklasse die interessant zou zijn voor het project. Neem onderstaande tabel uit (Benson, 2010, p 36).

	Data rate	Transmission distance	Transmit & Receive power	Cost	Firmware and software design
Teledyne Benthos	2400 bps	2-6 km	12 W 0.4 W	$10,000	Proprietary
LinkQuest	9600 bps	1500 m	4 W 0.8 W	$8,000	Proprietary
WHOI Micro-Modem	80 bps (FH-FSK) 300-5400 (PSK)	1-10 km	10-100 W 200 mW – 2W	$8,000	All design information is available online.
UCSD Modem	200 bps	2 km	1 – 40 W 1W	$600	All design information will be available online.

Figuur 3.3: Overzicht bestaande modems

Zelfs het UCSD modem is nog te duur voor het budget met $600. Op basis van bovenstaande is er besloten te onderzoeken of het mogelijk was om zelf een bi-directioneel ultrasoon akoestisch modem te ontwikkelen voor een bedrag wat binnen het totaal budget zou passen. Geïnspireerd door de paper

van het Oceans 10 modem (Benson, 2010) is er begonnen aan het zelf ontwikkelen van een Open Source modem. Meer informatie hierover is te vinden in hoofdstuk 4.4.

3.4. Overige hardware

Nadat vastgesteld was hoe de communicatie hardware zou gaan functioneren was het tijd om te kijken naar de overige hardware die nodig is voor het project. Hierbij valt te denken aan de apparatuur die door de duiker meegenomen wordt, maar ook aan de boeien die nodig zijn voor de communicatie en de positie bepaling.

In lijn met de rest van het project is er zoveel mogelijk gebruik gemaakt van bestaande Open Source hard- en software. Dit maakt het voor andere partijen mogelijk om zonder onnodige kosten dit systeem te implementeren. In hoofdstuk 4.1 wordt aangegeven welke hard- en software er gebruikt wordt om het totale systeem te laten functioneren.

3.5. Backend

Nadat alle niet standaard componenten en technieken bekend waren, is er begonnen met het kiezen van de juiste architectuur en software oplossingen om de infrastructuur van het systeem vorm te geven.

Zoals aangegeven gaat ook hier de voorkeur uit naar Open Source oplossingen. Binnen het project was er primair ervaring met een aantal front-end technieken zoals *AngularJs* en *Typescript*. Voor de serverimplementatie was deze ervaring vooral aanwezig op het gebied van Microsoft .Net oplossingen. Om vast te blijven houden aan de lijn van gebruikte technieken is er besloten het Backend te implementeren in Node.js, een breed gebruikte oplossing voor server implementaties in Open Source systemen. De concrete uitwerking van de architectuur is vastgelegd in hoofdstuk 4.1.

3.6. Wetgeving

Toen aan het project begonnen werd, is er zoveel mogelijk onderzoek gedaan naar de mogelijke wetgeving die van toepassing is op de gebruikte technieken. Naast de Telecommunicatiewet (*Telecommunicatiewet*, 2016) die van toepassing is op de boven water gebruikte RF technieken, werd geen wetgeving ontdekt die van toepassing is op onderwater communicatie.

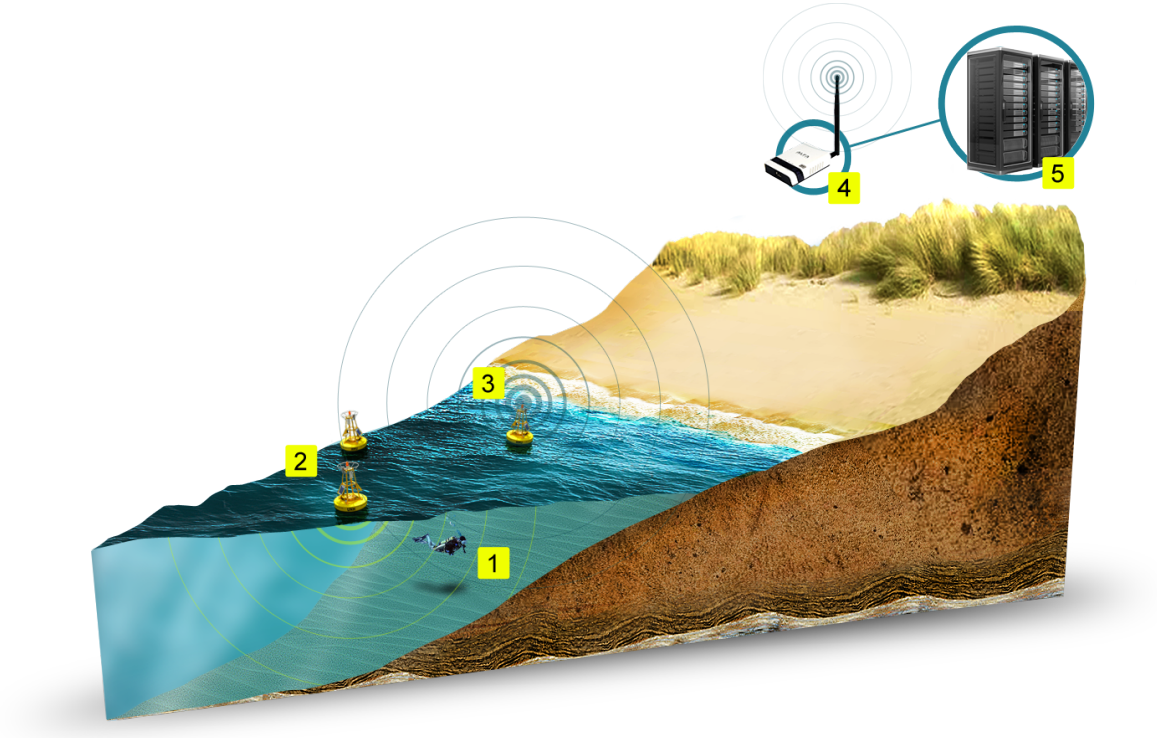
In April 2016 is het project team samen gaan werken met de Customisable Hydrophonic Rescue and Recovery Badge (CHRRB): Verdronken Personen Detectie groep. Deze groep werkt vanuit de De Rotterdamsche Droogdok Maatschappij (RDM) locatie van de Hogeschool Rotterdam en Omstreken (HRO) aan een systeem om drenkelingen op de binnenvaart te lokaliseren. Er is veel overlap tussen beide projecten en een samenwerking was dan ook een logische stap. Binnen de CHRRB groep wordt gewerkt aan een voorstel om onderwater communicatie te reguleren. Het is de intentie om het buddyAssist systeem te laten voldoen aan dit voorstel.

Naast de Telecommunicatiewet (*Telecommunicatiewet*, 2016) en het voorstel uit de CHRRB groep is het ook niet toegestaan om zomaar boeien in het water te plaatsen. Rijkswaterstaat treedt op als beheerder van de Nederlandse wateren en zal toestemming moeten verlenen voor het plaatsen van de boeien. Er is contact met Rijkswaterstaat om te bepalen waar de boeien aan moeten voldoen en hoe de procedure om ze te plaatsen er uit gaat zien. Vanuit de CHRRB groep is ook een vertegenwoordiging van Rijkswaterstaat aangesloten waarmee deze zaken besproken worden. Er is op dit moment nog geen uitsluitsel over deze zaken, wel is er toestemming verleend om op 1 duik locatie de bestaande boeien van Rijkswaterstaat te gebruiken voor het buddyAssist systeem.

4

Uitwerking

In de voorgaande hoofdstukken is het ontstaan van het systeem en de uitslagen van de onderzoeken besproken. In dit hoofdstuk wordt invulling gegeven aan de concrete uitwerking en implementatie van het systeem als geheel. Speciale aandacht wordt gegeven aan de positie bepaling en de werking van het modem.



Figuur 4.1: Overzicht van het gehele systeem

In bovenstaande afbeelding is schematisch weergegeven welke onderdelen deel uitmaken van het systeem. In de navolgende paragrafen wordt gedetailleerd invulling gegeven aan deze onderdelen. Voor een globaal inzicht in het systeem is het handig om vooraf de onderdelen te benoemen met een korte beschrijving.

1. **De duiker** maakt gebruik van het buddyAssist duiker systeem. Dit biedt de duiker inzicht in zijn of haar locatie en communiceert deze locatie met het oppervlaktesysteem.
2. **De boei** is verantwoordelijk voor het communiceren van de ontvangen signalen van duikers met de hoofd boei en verstuurt op reguliere intervallen een signaal onder water waardoor het duiker systeem in staat is om de positie te bepalen.

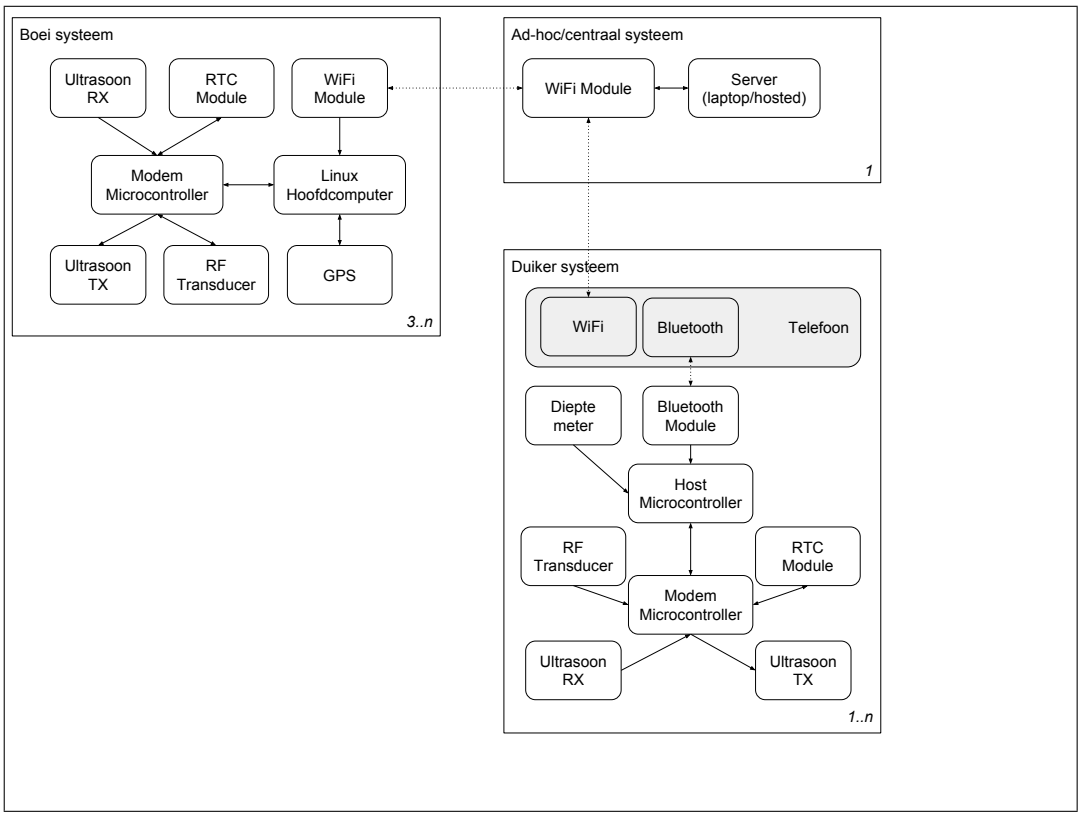
3. **De hoofd boei** is verantwoordelijk voor het communiceren van de ontvangen signalen van duikers met het centrale systeem. Dit kan gebeuren door middel van WiFi of met long-range RF oplossingen. Tevens geeft de hoofd boei een tijdssignaal af waarmee alle componenten in het systeem gesynchroniseerd worden. Daarnaast vervult de hoofd boei dezelfde taken als een normale boei.

4. **Het Access Point** is de toegang van het systeem op de duik locatie naar het centrale systeem. Dit kan in een ad hoc situatie (zie 4.1.4) een WiFi router zijn, en in de permanente systemen een RF of WiFi long-range toegangspunt.

5. **Het Centrale Systeem** is de verzameling van de servers en services die zorgen voor het opslaan, verwerken en tonen van de gegevens van de gebruikers.

4.1. Het globale systeem

In de vorige afbeelding is globaal getoond uit welke onderdelen het systeem bestaat. Onderstaand diagram kijkt gedetailleerder naar de componenten waaruit die onderdelen bestaan.



Figuur 4.2: Overzicht van de systeemcomponenten

4.1.1. Duiker systeem

Het systeem wat de duiker meeneemt onder water bestaat uit twee hoofdonderdelen, een microcontroller welke het modem en de overige hardware componenten aanstuurt en een telefoon. Op de telefoon draait de buddyAssist applicatie. Deze applicatie vervult twee functies, enerzijds is het de toegang tot het buddyAssist backend systeem en stelt het de gebruiker in staat om duik locaties te zoeken en te communiceren met de andere gebruikers, anderzijds is het de duikcomputer.

Qua behuizingen is het systeem ook opgedeeld in twee stukken, de centrale computer met het modem bevindt zich op de duikfles, dit is de meest optimale positie voor het ontvangen van signalen van de boeien. Daarnaast is er een tweede behuizing die een kleine microcontroller bevat en een Bluetooth module waarmee met de telefoon gecommuniceerd wordt. De telefoon wordt in deze tweede behuizing

geplaatst, welke waterdicht kan worden afgesloten.

Als de app constateert dat het systeem onder water gaat, schakelt het over naar de duikcomputer en wordt deze modus vergrendeld totdat de duiker weer aan de oppervlakte is. Tevens kan de duiker deze modus zelf inschakelen aan de oppervlakte. In deze modus worden de standaard informatie blokken getoond welke relevant zijn voor een duiker. Daarnaast toont het systeem de positie van de duiker en eventuele Points Of Interest (POI), objecten of locaties die interessant kunnen zijn voor de duiker.

De relevante informatie voor een duiker bestaat onder andere uit de volgende informatie:

- Huidige diepte
- Maximaal bereikte diepte tijdens deze duik
- Maximale tijd op deze diepte tot aan decompressie duik
- Totale duiktijd van deze duik
- Stijg- en daalsnelheid
- Hoeveelheid aanwezige zuurstof

Op de behuizing zijn 4 hardware knoppen aanwezig welke onder water te bedienen zijn. Deze knoppen geven de duiker de mogelijkheid om zelf POI te markeren en om instellingen uit te voeren en keuzes te maken bij eventuele alarmen.

De app welke op de telefoon draait, wordt ontwikkeld middels het Apache Cordova/PhoneGap framework. Dit framework maakt het mogelijk om applicaties te ontwikkelen in HTML en Javascript welke vervolgens gecompileerd kunnen worden om als een app op de verschillende mobiele besturingssystemen te draaien. Door gebruik te maken van dit framework hoeft de applicatie niet voor elk besturingssysteem afzonderlijk ontwikkeld te worden en besparen we ontwikkeltijd. Tevens is dit framework erg populair binnen de Open Source gemeenschap. De gebruikte technieken zijn bekend bij veel ontwikkelaars, wat er voor zorgt dat het makkelijker is voor derden om verbeteringen toe te voegen aan het systeem.

4.1.2. Het boei systeem

Het systeem wat op de boeien komt te draaien heeft meerdere verantwoordelijkheden. Enerzijds moet elke boei op exact het juiste moment een puls afgeven op basis waarvan de duiker systemen de positie kunnen bepalen, anderzijds moeten ze in de tussentijd luisteren naar berichten van de duikers. De berichten moeten vervolgens aangeboden worden bij het centrale systeem.

Omdat de functionaliteiten van de boei groter zijn dan wat de gemiddelde microcontroller aan kan, maar het wel extreem belangrijk is dat de pulsen met een milliseconden specifieke interval verstuurd worden, is dit systeem gesplitst in twee onderdelen. De microcontroller welke het modem aanstuurt en gesynchroniseerd is tot op de milliseconde en de computer die de data verzamelt en communiceert met het centrale systeem. Voor deze computer was initieel gekozen voor een Raspberry Pi omdat dit het goedkoopste moederbord was met volledige Linux support en aansluitmogelijkheden. Tijdens de ontwikkeling bleek echter dat er nog teveel 'nutteloze' aanluitingen aanwezig zijn en de leverbaarheid van de 'kale' versie, de Raspberry Pi Zero erg te wensen over laat.

Het nodige speurwerk leverde een alternatief op in de vorm van een Linkit Smart 7688 (Due) [1], een Linux compatibele computer met een erg klein formaat en een lage spanningsbehoefte. Dit systeem is voorzien van twee USB poorten en alle overige aansluitingen zijn uitgebroken naar pennen waar vervolgens, al naar gelang de behoefte binnen het systeem, zelf invulling aan gegeven kan worden. Tevens is de Duo variant voorzien van een geïtegreerde Arduino compatibele microcontroller. De prijs en functionaliteit van deze borden maken ze zeer geschikt voor het gebruik in het systeem.

[1] https://www.seeedstudio.com/item_detail.html?p_id=2574

Deze computer wordt voorzien van de nodige software welke verantwoordelijk is voor het doorgeven van de data van de duikers en de eigen positie vanuit de GPS module. Tevens kunnen de boeien op afstand benaderd worden via het WiFi netwerk om aanpassingen en updates door te voeren.

4.1.3. Het centrale systeem

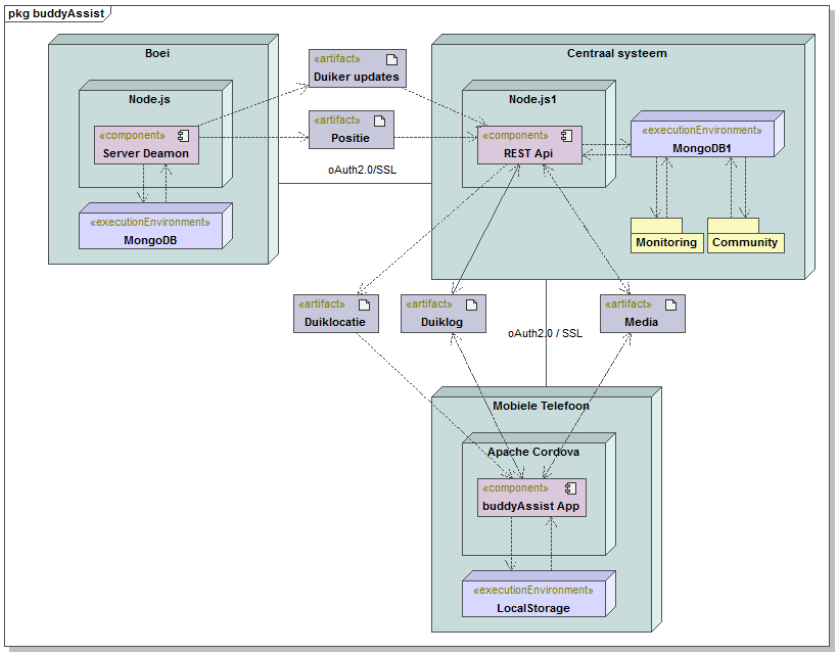
Het centrale systeem is in feite het samenbrengen van alle data van diverse duik locaties op 1 plek (zie 4.2.1). Deze duik locaties zijn dan permanent voorzien van buddyAssist boeien en maken via internet verbinding met de centrale server. De duikers kunnen zelf ook inloggen op dit centrale systeem om hier hun duiken bij te houden en deze te delen met andere gebruikers van het systeem. In hoofdstuk 4.2.3 wordt verder ingegaan op de mogelijkheden van dit systeem en de voorgestelde functionaliteiten.

4.1.4. Het ad hoc systeem

Het ad hoc systeem is bedoeld voor hulpverleningsinstanties zoals de brandweer wanneer er in een gebied gezocht moet worden naar personen of objecten. Dit systeem is in grove lijnen gelijk aan het globale systeem, alleen staat de server niet op internet maar draait deze op een machine die fysiek op de locatie aanwezig is. De overige hard- en software is identiek aan het centrale systeem. Alleen de community website ontbreekt in deze opzet.

4.2. Systeem architectuur

Het systeem bestaat qua software uit een aantal verschillende componenten welke zijn weergegeven in onderstaand diagram. Hierin is ook aangegeven hoe de beveiliging bij de verschillende berichtenstromen is geïmplementeerd. In 4.2.4 wordt verder ingegaan op de gebruikte oplossingen en worden deze getoetst aan de bestaande voorschriften.



Figuur 4.3: Overzicht van de software architectuur

4.2.1. REST API

De data communicatie met het systeem vindt plaats door middel van een zogeheten REST api. Binnen REST api's wordt gesproken van *resources* waarop bewerkingen worden uitgevoerd. In de implementatie van dit systeem bestaan onder andere de volgende resources:

- **Duiklocatie** Een locatie waar gedoken wordt, kan ad hoc of permanent zijn
- **Boei** Een boei welke zich op een duik locatie bevindt

- **Duiker** Een duiker welke zich onder water op een duik locatie bevindt
- **Gebruiker** Een duiker welke zich niet onder water bevindt
- **Duik log** Het logboek van een afgeronde duik door een duiker op een duiklocatie

Binnen REST zijn de objecten *hiërarchisch* georganiseerd. In bovenstaande opsomming is die hiërarchie terug te vinden in de wijze waarop de resources besproken zijn, een **Boei** bevind zich *op* een **Duiklocatie**. De aanroep naar deze resource ziet er dan als volgt uit: */duiklocaties/1/boeien/14*. Hierbij wordt gezegd dat op de duiklocatie met id 1 de boei met id 14 gemanipuleerd gaat worden.

Het gebruik van de API valt ruwweg in 3 fasen uiteen: de duiker bevindtzich aan de oppervlakte op een duiklocatie en vraagt de gegevens op over deze locatie met de mobiele applicatie; De duiker gaat duiken en de boei(en) rapporteren de locatie en status van de duiker; De duiker komt weer boven water en kan, al dan niet direct, het log van de duik met veel meer details uploaden naar het centrale systeem. Dit log kan voorzien worden van foto 's en commentaren.

Om de REST api te implementeren is gekozen voor Node.js [2], een Javascript runtime welke op diverse besturingssystemen kan draaien. In tegenstelling tot de reguliere Javascript welke primair bedoeld is om bij gebruikers op een systeem te draaien, is het doel van Node.js om op een server ingezet te worden. De keuze voor Node.js is gemaakt op basis van de beschikbare uitleg over het gebruik, mijn bekendheid met Javascript als ontwikkel taal en de platform-onafhankelijkheid. Binnen Node.js wordt gebruik gemaakt van diverse *modules* welke extra functionaliteiten implementeren. Om het platform te implementeren wordt gebruik gemaakt van onder andere de volgende modules:

- **Express** Om de http(s) REST interface te definiëren
- **Mongoose** ORM systeem om eenvoudiger met de MongoDB te communiceren
- **Passport** Om de diverse beveiligingsoplossingen te implementeren
- **Swagger-node** Om de REST api te kunnen modelleren en om de te kunnen publiceren aan externe partijen

Alle bewerkingen op de resources worden opgeslagen in de Document Database MongoDB. Dit type database slaat *documenten* op in plaats van *records*. Voor de data die gegenereerd wordt, is het efficiënter om deze als document op te slaan dan als records. Het systeem stelt de ontwikkelaar in staat om zonder veel code de eigenschappen van de resources te manipuleren en op te halen.

Fase 1 - Duiker aan de oppervlakte

De duiker is aangekomen op de duik locatie en bevindt zich nog aan de oppervlakte. De duiker start de buddyAssist applicatie op de mobiele telefoon en maakt met zijn of haar identificatie verbinding met het centrale systeem. De gegevens over de huidige duik locatie worden opgehaald, belangrijk hierbij zijn vooral de identificatie en de locaties van de op deze locatie aanwezige boeien. Na het ophalen van deze gegevens is de duiker klaar om door te gaan naar fase 2.

Fase 2 - Duiker gaat onder water

Gedurende de daadwerkelijke duik is er geen rechtstreekse verbinding met het centrale netwerk mogelijk, de buddyAssist applicatie slaat op de mobiele telefoon een gedetailleerd log op van de gehele duik. Op gezette intervallen (zie 4.4.2) stuurt de apparatuur een bericht wat door 1 of meerdere boeien ontvangen wordt. Deze boeien zorgen dan voor het doorgeven van deze gegevens aan het centrale systeem middels de REST api. Het interval van deze gegevens is per duiker vele malen groter dan de data welke opgeslagen wordt op de mobiele telefoon. Gemiddeld zou deze informatie 1 keer per totale interval per duiker bijgewerkt moeten worden. Dat is dus in het geval van 16 boeien (praktisch maximum) 1 maal in de 16 seconden per duiker. Na een succesvolle opstijging zal de duiker het water weer verlaten.

[2] https://nodejs.org/en/

Fase 3 - Duiker is weer aan land

Na het omkleden en demonteren van de apparatuur begint de belangrijkste fase van het duiken: het napraten. Tijdens deze fase kan door middel van de buddyAssist mobiele applicatie het gedetailleerde duik log geüpload worden naar de centrale server. Dit log is direct gedeeld met de andere gebruikers van het systeem waarvan de gebruiker heeft aangegeven dat ze dit mogen zien. Tevens krijgt de gebruiker de logs binnen van mensen die aangegeven hebben dit met hem of haar te willen delen. Traditioneel is het zo dat in de papieren logboeken de duikpartner (buddy) een handtekening zet bij het log van de partner en vice versa, dit kan middels de buddyAssist applicatie digitaal.

4.2.2. Flightradar en Monitoring Website

De flightradar en monitoring website dankt zijn naam aan het concept van de flightradar [3] website. Het doel van deze website is het *per duik locatie* visualiseren van de onder water aanwezige duikers en hun status. Een duiker kan vanwege privacy redenen aangeven dit niet wenselijk te vinden waarna deze duiker niet meer te zien zal zijn op het overzicht. Deze website wordt ook gebruikt in het ad hoc systeem (zie 4.1.4) door de divemaster om de duikers op een tijdelijke locatie in beeld te brengen. De gegevens op deze website zijn kortstondig van aard. Het is mogelijk om tot 24 uur data terug te kijken. Data punten ouder dan 24 uur worden uit het systeem verwijderd.

4.2.3. Community Website

De community website kan het best omschreven worden als Facebook voor duikers. De website dient als portaal voor de gebruikers van het buddyAssist systeem. Op de website kan gezocht worden naar duik locaties, gebruikers kunnen beoordeling geven aan deze locaties en het duiklog van duikers wordt hier digitaal bijgehouden. Alle elementen kunnen voorzien worden van commentaar door de gebruikers en het is mogelijk om bij de logs en locaties foto 's en video 's te plaatsen. Daarnaast is het mogelijk om vrienden te worden met andere gebruikers (in de duiksport worden dit buddies genoemd) zodat duiklogs en berichten van die gebruikers getoond worden aan de gebruiker. Daarnaast is het mogelijk voor bedrijven die direct verbonden zijn aan de duiksport te adverteren op de website. Dit gebeurt op een gecontroleerde manier, duikwinkels en vulstations kunnen aangeven dat ze getoond willen worden bij bepaalde duik locaties. Buiten deze nuttige informatie om, worden de gebruikers niet geconfronteerd met advertenties.

4.2.4. Security

Binnen het gehele systeem wordt gebruikt gemaakt van industrie-standaard beveiligingsoplossingen. De REST api en de websites worden door middel van oAuth2.0/OpenID afgeschermd. Dit is het protocol wat ook gebruikt wordt door de grotere bedrijven zoals Google en Facebook. Tijdens de implementatie wordt de werking zoveel mogelijk getoetst aan het OWASP project met betrekking tot Web Applicaties [4].

De communicatie via het ultrasone modem (zie 4.4) is niet voorzien van beveiliging en vormt dus in de huidige implementatie een risico. Het is praktisch erg eenvoudig om communicatie van een boei of een duiker te onderscheppen of te vervangen. Voor nu wordt dit risico geaccepteerd. Als het model verder ontwikkeld wordt en een hogere doorvoersnelheid toestaat kunnen de gegevens worden versleuteld. Nu zou de versleuteling te veel overhead op de data opleveren.

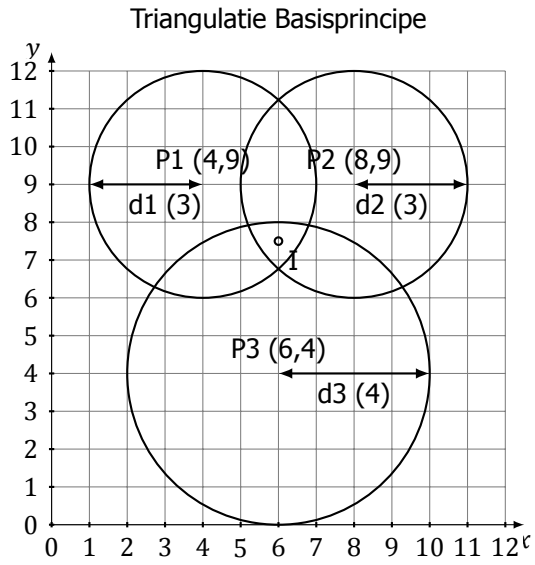
4.3. Positiebepaling

Het eerste deel van de concrete implementatie is gericht op het accuraat en snel bepalen van de positie op basis van de gemeten afstanden. Deze techniek heet triangulatie. Na het onderzoeken van diverse technieken leek het door MIT ontwikkelde TULIP algoritme het meest veelbelovend.

4.3.1. TULIP

Net als bij alle triangulatie methoden gaat TULIP uit van de afstand gemeten tot 3 punten waarvan de locatie bekend is.

[3] https://www.flightradar24.com/
[4] https://www.owasp.org/index.php/Web_Application_Security_Testing_Cheat_Sheet



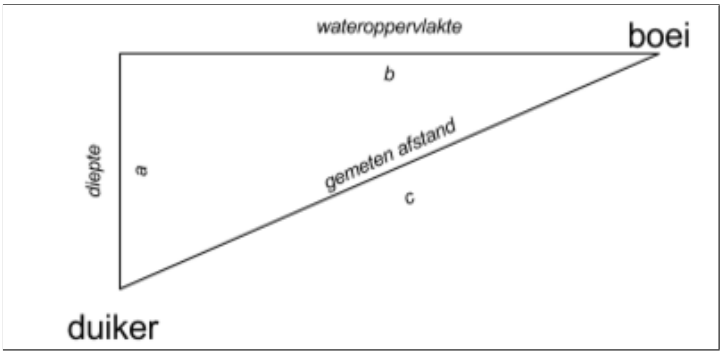
Figuur 4.4: Triangulatie basisprincipe schematisch

TULIP geeft een redelijk eenvoudige vergelijking om de x en y coördinaat van de gewenste intersectie I te vinden. Deze vergelijking wordt opgebouwd met de onderstaande variabelen:

$$i1 = P1.x$$
$$i2 = P2.x$$
$$i3 = P3.x$$
$$j1 = P1.y$$
$$j2 = P2.y$$
$$j3 = P3.y$$

Figuur 4.5: De variabelen voor TULIP

In principe is het mogelijk om deze logica te gebruiken om in 3 dimensies de positie te bepalen, echter is het in deze specifieke toepassing niet nodig om de 3e coördinaat te berekenen, de diepte is namelijk met hoge resolutie bekend door het gebruik van een diepte sensor in het systeem van de duiker. De afstand die gemeten wordt vanaf de boei is echter wel inclusief diepte component. Om dit op te lossen wordt dit component er uit gehaald, gebruik makend van de *Pythagoras* formule.



Figuur 4.6: Pythagoras oplossing voor diepte component

We gebruiken dan de standaard $a^2 + b^2 = c^2$ formule om de gemeten afstand *c* te vertalen naar de oppervlakte afstand *b*. De waarden worden gebruikt in de verdere TULIP berekening.

De daadwerkelijke berekening van de 2D *x* en *y* coördinaten is dan volgens onderstaande formule.

$$
\begin{aligned}
I.x = (((&\\
&(d1^2 - d2^2) + (i2^2 - i1^2) + (j2^2 - j1^2)\\
&) * (2 * j3 - 2 * j2) - (\\
&(d2^2 - d3^2) + (i3^2 - i2^2) + (j3^2 - j2^2)\\
&) * (2 * j2 - 2 * j1)\\
&)/(\\
&(2 * i2 - 2 * i3)(2 * j2 - 2 * j1) - (2 * i1 - 2 * i2)(2 * j3 - 2 * j2)\\
&)
\end{aligned}
$$

$$
\begin{aligned}
I.y = ((d1^2 - d2^2) + (i2^2 - i1^2) + (j2^2 - j1^2) +&\\
x * (2 * i1 - 2 * i2))/(2 * j2 - 2 * j1)&
\end{aligned}
$$

Figuur 4.7: TULIP Algoritme

In het vooronderzoek was reeds onderzocht welke variabelen van invloed zijn op de positiebepaling en welke technieken gebruikt kunnen worden om de meting te middelen. In de paper van Wouter (van Kleunen, 2014, p 105) is reeds bepaald dat er een afwijking kan worden verwacht van tussen 8 en 10 meter.

4.3.2. Box-grid schatting

Tijdens het ontwikkelen van de simulatie voor het TULIP algoritme is nagedacht over mogelijke verbeteringen. Samen met collega, medestudent en project lid Willem de Voogd is een eigen algoritme bedacht wat betere resultaten oplevert dan TULIP en als toevoeging ook een indicatie kan geven van de nauwkeurigheid van de bepaling. Deze indicatie kan worden gebruikt om naar de gebruikers van het systeem aan te geven hoe 'zeker' de positie is.

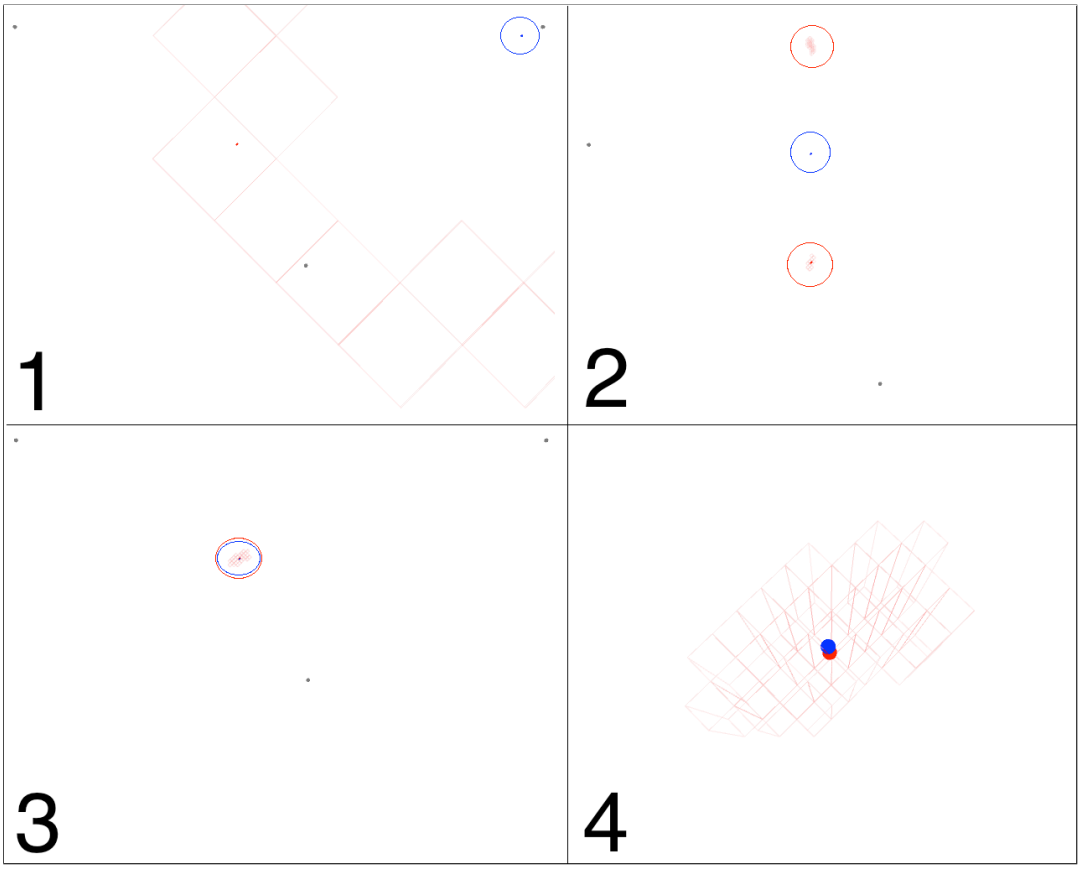
Bij gebrek aan een officiële benaming heeft het algoritme de naam Box-grid Accoustic Position Estimation (BAPE) gekregen. Deze calculatie is toegevoegd aan de simulatie (zie 3.2) en tijdens het testen is door beide algoritmes tegelijk de positie berekend. De resultaten hiervan zijn opgeslagen zodat deze gevisualiseerd en vergeleken kunnen worden.

4.3.3. BAPE algoritme

Het BAPE algoritme werkt door het duikgebied op te delen in *n* kubussen op de diepte van de duiker, dit is immers een vast gegeven. Vervolgens wordt gekeken welke kubussen op de straal van de ontvangen afstanden van de boeien liggen. Deze kubussen worden gemarkeerd als valide en opgedeeld in weer kleinere kubussen totdat het aantal mogelijke kubussen zeer snel toeneemt. Als dat punt bereikt is dan is de bepaling zo nauwkeurig als mogelijk met de metingen die er zijn.

Het grote verschil vergeleken met TULIP is dat de gemeten afstanden tot de boeien voorzien zijn van een tijds component, hoe ouder de meting hoe 'dikker' de lijn waarin de kubussen kunnen vallen. Hiermee wordt voorzien in de beweging van de duiker. De uitersten van de mogelijke kubussen fungeren als een indicatie van de onnauwkeurigheid van de bepaling. Als de kubussen verspreid zijn over een grote oppervlakte is het algoritme minder zeker van de positie dan wanneer de kubussen zich op een relatief kleine oppervlakte bevinden.

In het volgende figuur worden de fasen van BAPE getoond zoals deze in de simulatie zichtbaar zijn. De rode cirkels zijn de clusters kubussen, de blauwe cirkel is de berekende positie. De kleine rode stip is de daadwerkelijke positie van de duiker, de grijze stippen zijn de boeien.



Figuur 4.8: De fasen van het BAPE algoritme

Fase 1 - afstand tot 1 boei

Er is in deze fase slechts een signaal van 1 boei ontvangen, het gebied van mogelijke kubussen vormt een ruwe cirkel rondom deze boei. Het gevonden middelpunt is vrijwel gelijk aan de locatie van die boei.

Fase 2 - afstand tot 2 boeien

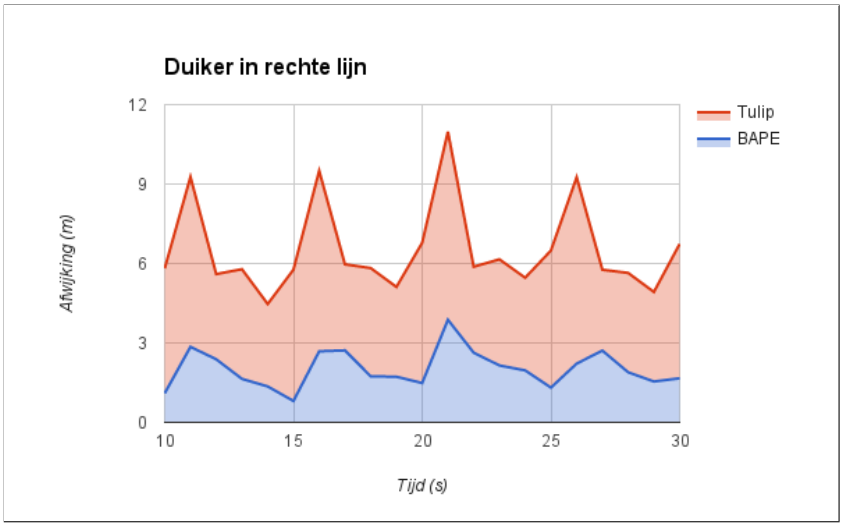
In deze fase zijn er 2 afstanden bepaald, er zijn twee mogelijke concentraties kubussen, het berekende middelpunt bevindt zich in het midden van de clusters. De daadwerkelijke positie van de duiker bevindt zich in het onderste cluster.

Fase 3 - afstand tot 3 boeien

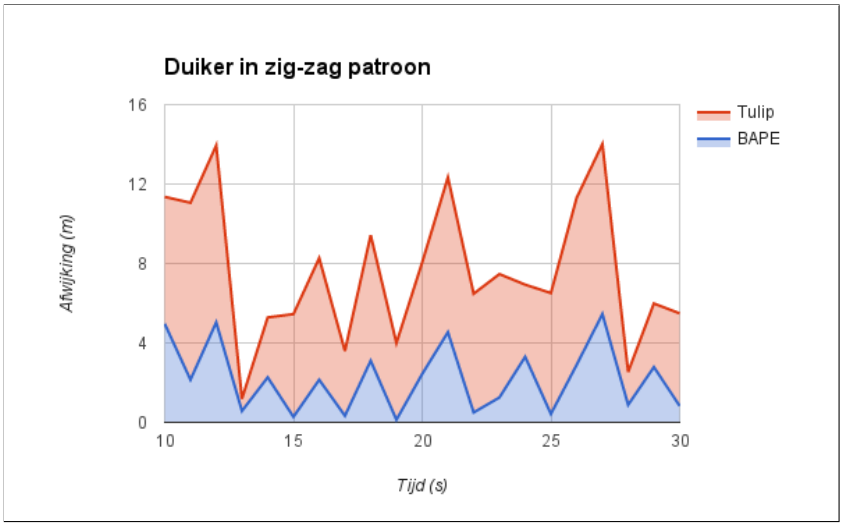
Er is een afstand ontvangen van 3 boeien, de kubussen zijn geclusterd rondom de daadwerkelijke locatie van de duiker. In stap 4 is dit in detail ingezoomd, te zien is hoe de blauwe berekende positie samenvalt met de rode daadwerkelijke positie.

4.3.4. Resultaten en vergelijking

Als de beide algoritmes gevisualiseerd worden in een grafiek waarbij de afwijking in meters ten opzichte van de echte positie geplot word dan is er een duidelijk verschil zichtbaar tussen BAPE en TULIP. Er zijn twee bewegingspatronen gebruikt in de simulatie, een rechte lijn en een zig-zag patroon. In onderstaande figuren worden de uitkomsten van de simulatie getoond.



Figuur 4.9: Simulatie resultaten bij rechte lijn



Figuur 4.10: Simulatie resultaten bij zig-zag

Er is duidelijk te zien dat BAPE minder gevoelig is voor veranderingen in richting en het interval waarmee de signalen van de boeien ontvangen worden. Wanneer in een tabel het gemiddelde en de mediaan van de simulaties uitgerekend wordt, is zichtbaar dat BAPE significant beter presteert als er geen verdere verbeteringen aan het TULIP algoritme worden doorgevoerd.

Resultaten BAPE en TULIP

Straight	*Average*	*Median*	Zig-zag	*Average*	*Median*
BAPE	2.021340286	1.8922431	BAPE	2.203146839	2.1459386
TULIP	4.511043529	4.0866985	TULIP	5.44834009	5.9782495

Figuur 4.11: Resultaten BAPE en TULIP

Op basis van bovenstaande metingen is besloten BAPE te gebruiken bij de echte implementatie van het systeem. De voordelen in reactiesnelheid en de beschikbaarheid van een indicatie van de nauwkeurigheid zijn waardevolle toevoegingen aan het systeem.

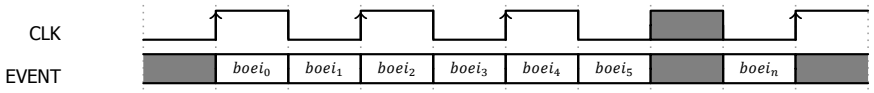
4.3.5. Praktisch

Het algoritme om de positiebepaling uit te voeren is nu bekend. De praktische implementatie is nog iets complexer dan de simulatie, maar niet veel. Het belangrijkste is dat alle betrokken systemen dezelfde '0' tijd hanteren om zo te kunnen bepalen hoe lang een signaal er over deed om aan te komen, daarom moeten deze gesynchroniseerd worden. Dit wordt bereikt door van 1 boei (de dichtste bij de wal) de zogeheten **hoofd** boei te maken, deze boei is de centrale spil in het netwerk van boeien. Deze boei geeft ook de '0' tijd aan door middel van een RF module welke over lange afstand kan communiceren. De boei stuurt op de door hem bepaalde '0' tijd een puls. Deze wordt vrijwel direct ontvangen door alle ontvangers in de buurt van deze boei. Zowel de boeien als de duikers hebben een ontvanger in het systeem zitten. Bij het ontvangen van de puls zetten deze systemen dat moment als '0' tijd in het systeem.

Vanaf het moment dat een duiker onder water gaat, zal de RF ontvanger geen synchronisatie puls meer ontvangen, het is dan erg belangrijk om het verschil ten opzichte van de laatste '0' tijd zo accuraat mogelijk te meten. De gebruikte micro controllers hebben een ingebouwde klok, maar omdat de tijd synchronisatie zo 'n belangrijk component in het systeem is worden de micro controllers voorzien van een zogeheten Real Time Clock (RTC) waarmee een verloop van enkele seconden op een jaar te bereiken is. De specifieke RTC die in het systeem gebruikt wordt, is de DS3231 van Maxim[5]. Alle tijdkritische systemen worden voorzien van een dergelijke chip.

De tijd synchronisatie is ondervangen met bovenstaande oplossing. Daarnaast is het belangrijk om zo vaak mogelijk, in een zo klein mogelijke periode, signalen te ontvangen van zoveel mogelijk boeien in het duikgebied. Bij een verplaatsing van de duiker moet het niet te lang duren voordat er weer nieuwe metingen zijn. De grootste beperking hierin is de ultrasone communicatie zelf. Zoals in detail uitgelegd wordt in hoofdstuk 4.4, duurt het versturen van een signaal meer dan een halve seconde in de huidige implementatie. Het is belangrijk om tussen de signalen enige tijd geen signaal te versturen zodat eventuele weerkaatsing of ruis geen factor is bij het volgende signaal.

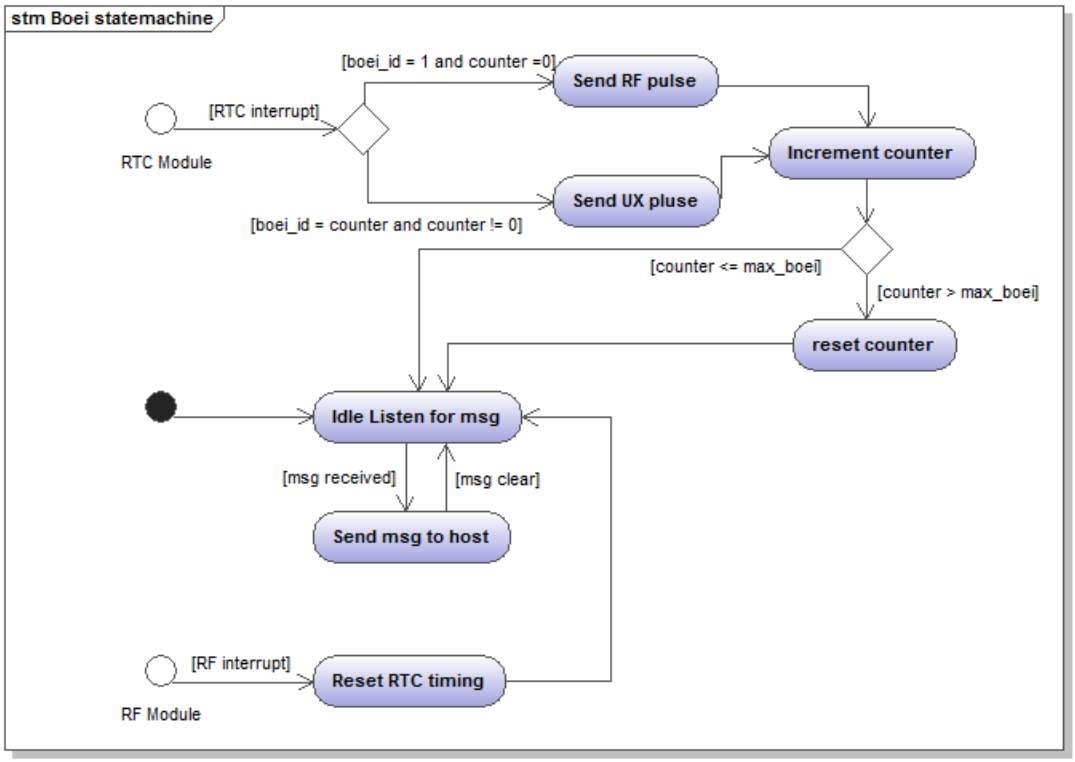
Om dit op te lossen is er gekozen voor een zogeheten 'timeslot' oplossing. Alle communicatie wordt verdeeld in 'slots' van hele seconden. Deze keuze is gemaakt omdat de gebruikte RTC de optie biedt om op bepaalde intervallen een signaal af te geven, de kleinst mogelijke interval is een seconde. De RTC is dermate accuraat dat dit de beste referentie is naar de eerder vastgestelde '0' tijd.



Figuur 4.12: Timeslots voorbeeld

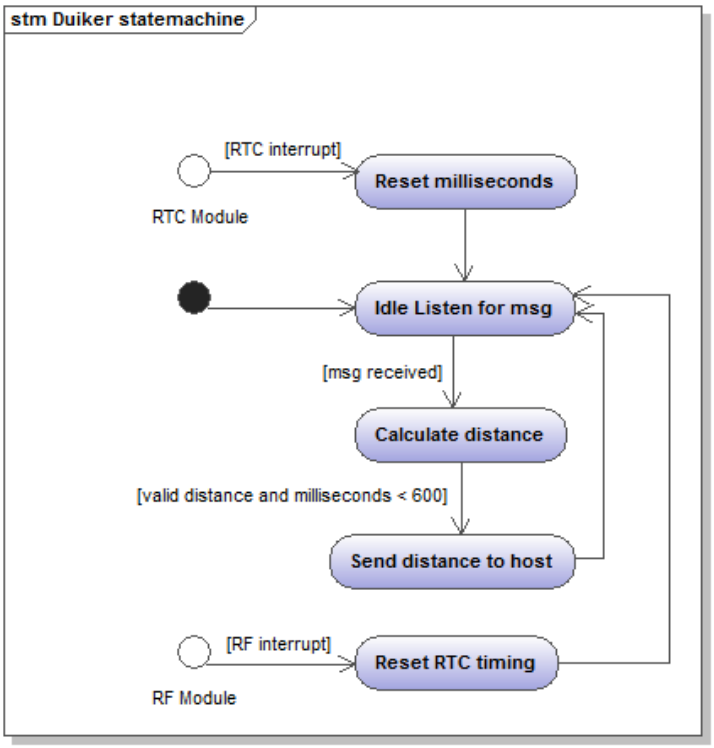
De **hoofd** boei heeft twee mogelijke acties op het moment dat de gehele seconde bereikt is volgens de RTC. Als de 'counter' 0 is dan stuurt hij een RF signaal om de andere systemen te laten weten dat dit de '0' tijd is. Is de 'counter' 1 (de hoofd boei heeft id 1) dan stuurt hij een ultrasoon bericht naar de duikers onder water. De overige boeien luisteren naar de RF interrupt om de '0' tijd te synchroniseren en implementeren dezelfde RTC interrupt om een ultrasoon signaal te versturen als de 'counter' gelijk is aan hun id.

[5] https://www.maximintegrated.com/en/products/digital/real-time-clocks/DS3231.html



Figuur 4.13: State diagram van de boeien

Ook de systemen van de duikers luisteren naar de RF synchronisatie puls. Zoals aangegeven zal de duiker de RF puls onderwater niet meer ontvangen. Elke keer dat de RTC een signaal afgeeft, wordt de interne milliseconden teller gereset naar 0. Op het moment dat er een signaal ontvangen is via de ultrasone ontvanger, wordt het aantal milliseconden waarop dat signaal ontvangen is, gebruikt om te bepalen hoe lang dat geleden is ten opzichte van de laatste 0 tijd. Dit verschil minus de tijd die het kost om een bericht te ontvangen (zie 4.4.2) is de duur dat het signaal erover gedaan heeft om de duiker te bereiken. Door de watertemperatuur in ogenschouw te nemen en of het gaat om zout- of zoet water, kan vrij accuraat bepaald worden wat de afstand tot de boei is. De wijze waarop dit geïmplementeerd is op de micro controller, is een zogeheten *interrupt*, dit garandeert de directe uitvoer van een stuk code als het signaal van de RTC ontvangen word.



Figuur 4.14: State diagram van de ontvanger eenheid van de duiker

De locatie en het id van de boeien in het duikgebied worden boven water gesynchroniseerd met de ontvangers van de duikers. Het is vervolgens mogelijk om met 3 gemeten afstanden de positie ten opzichte van deze boeien te bepalen door middel van het eerder uitgewerkte BAPE algoritme (zie 4.3). De locaties van de boeien worden door middel van GPS door de boeien zelf bepaald en doorgegeven aan het backend systeem. De systemen die gebruikt worden door de duikers halen middels WiFi de locaties van de boeien voor de huidige duik locatie op (zie 4.1 voor meer informatie).

4.4. Modem

In het vooronderzoek is gebleken dat de commerciëel beschikbare modems het budget ruim zouden overschrijden. Daarnaast zijn de specificaties vaak veel beter dan wat nodig is voor de doelstelling van dit project. Er zijn meer (Academische) projecten geweest om een goedkoper alternatief te maken zoals het Oceans10 (Benson, 2010) modem en Keelan Burns (Burns, 2013). Al deze modems refereren aan het 'oer' modem van Lee Freitag van het Woods Hole Oceanographic Institution (WHOI) (Freitag et al., 2005).

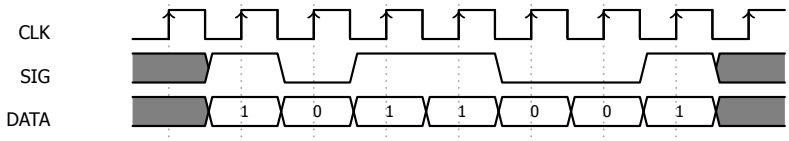
De zoektocht om een modem te ontwikkelen begon dan ook met een zeer positief Skype gesprek met Lee Freitag zelf. Met hem zijn de mogelijkheden besproken en is gevraagd of hij mee wilde werken aan de ontwikkeling. Helaas is zijn expertise te duur om 'zomaar' een project naast zijn reguliere werkzaamheden te doen, maar hij heeft wel alle specificaties gedeeld van het originele WHOI modem. Daarnaast is er gedurende de ontwikkeling meerdere malen contact geweest om op specifieke punten advies in te winnen.

Omdat er binnen het project veel gebruik gemaakt wordt van Open Source projecten voor zowel de hard- als software, moet het modem ook onder dezelfde licentie beschikbaar gesteld worden. Er wordt gebruik gemaakt van de Open Source Arduino compatibele micro controller voor het coderen en decoderen van de gegevens. Omdat voor deze micro controller reeds een grote gebruikersgroep bestaat, is de drempel voor ontwikkelaars om modificaties aan het modem te doen kleiner. Dit zal op termijn leiden tot verbeteringen aan het systeem door gebruikers rond de wereld.

4.4.1. Ultrasoon circuit

De basis van het modem is ultrasoon geluid. Dat betekent dat de informatie die verstuurd wordt, moet worden 'verpakt' in ultrasone geluidsgolven. Op basis van de eerder genoemde research papers en de gesprekken met Lee Freitag is gekozen voor een frequentie van 40 kHz. Voor ultrasoon geluid onder water geldt dat de afstand waarover gecommuniceerd kan worden groter wordt naarmate de frequentie lager wordt. 40 kHz ligt nog buiten het gehoor bereik van mensen en de meeste onderwater fauna. Een aantal experts binnen de CHRRB groep stellen het 'veilige' bereik op 32-40 kHz.

De eerste stap bestond uit het bepalen van de te gebruiken modulatie techniek, de wijze waarop de informatie omgezet word naar ultrasoon geluid. Er zijn veel technieken waarmee dat kan, maar omdat dit project weinig data wil sturen met zoveel mogelijk zekerheid is er gekozen voor een robuuste, eenvoudige techniek die weinig data toestaat maar wel tegen de nodige verstoring kan. Deze techniek heet On-Off Keying (OOK), een techniek waarbij informatie in pulsen word verzonden waarbij een hoog signaal een logische '1' weergeeft en geen signaal een logische '0'.

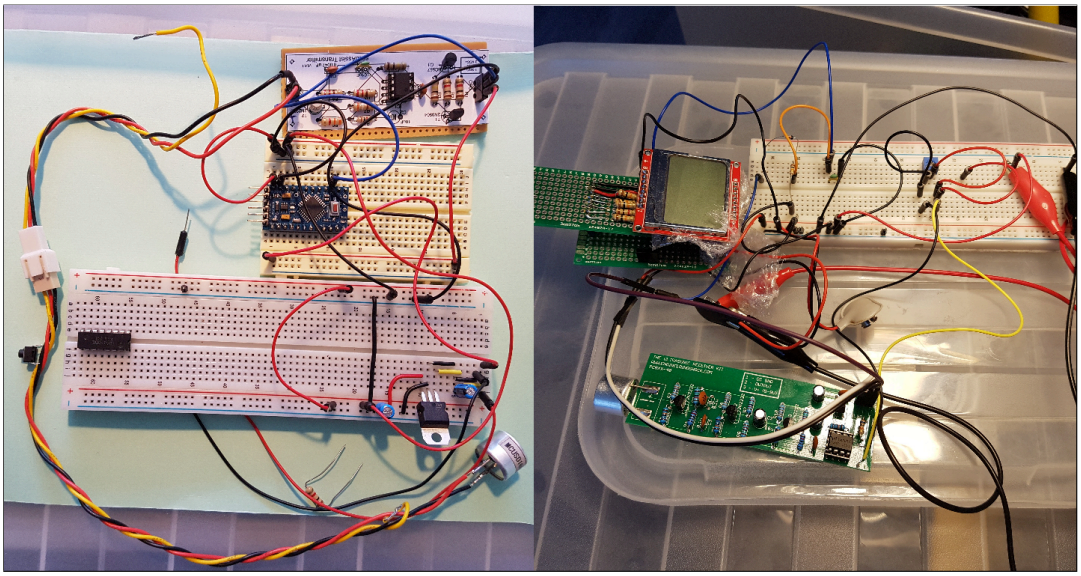


Figuur 4.15: Voorbeeld van OOK modulatie

Om deze modulatie te implementeren op de Arduino micro controller wordt gebruik gemaakt van de RadioHead (McCauley, 2008) library. Deze library implementeert een aantal modulatie protocollen met onder andere checksum controle en fout correctie. Deze library is van origine bedoeld voor het gebruik met goedkope RF zenders en ontvangers, maar door de wijze waarop de aansturing plaatsvindt is deze ook geschikt voor dit project.

De data die verstuurd moet worden verschilt voor de verschillende onderdelen van het systeem, maar het is belangrijk om vast te stellen dat de data niet langer mag zijn dan 60 bytes. Dit is een beperking opgelegd door het gebruik van de RadioHead (McCauley, 2008) library. Bij de implementatie van de protocollen is deze beperking meegenomen.

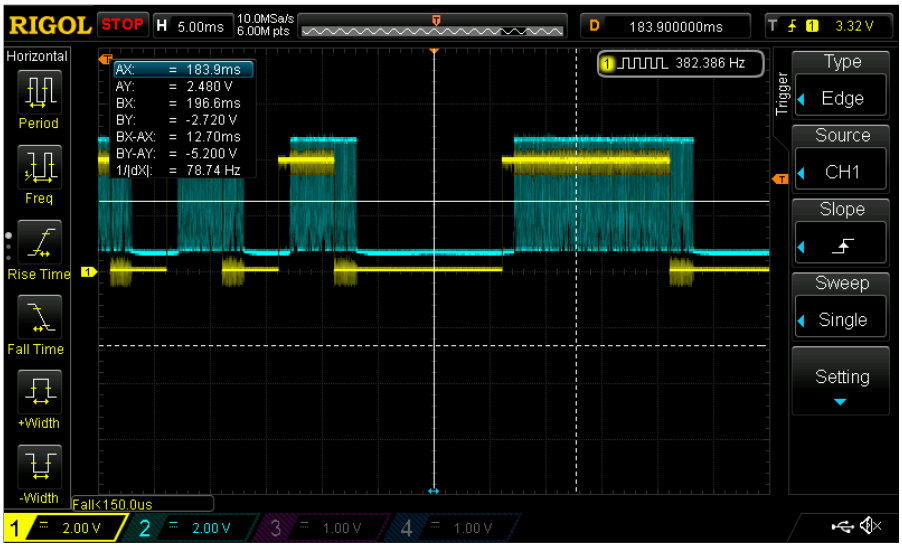
De volgende stap bestond uit het fabriceren van een stuk elektronica wat in staat was om de 40Khz ultrasone 'transducer' aan te sturen met de juiste frequentie. De aansturing van de pulsen moest gebeuren op 3.3 - 5 V omdat dit het bereik van de micro controller is, echter moet het daadwerkelijk aansturen van de transducer op hogere voltages mogelijk zijn. Dit is mogelijk gemaakt door met de lagere voltages de hogere voltages te schakelen. Hiermee is het mogelijk om een breed bereik aan micro controllers te gebruiken als aansturing van de modem hardware.



Figuur 4.16: Prototype van het zend- en ontvang gedeelte van het ultrasone modem

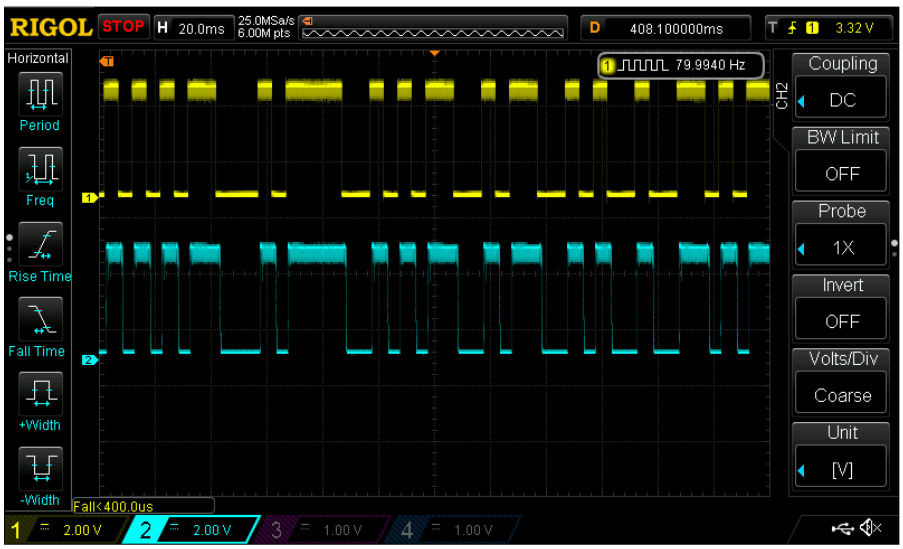
Nadat de zender gebouwd was werd het tijd voor de ontvanger. In principe moet de ontvanger alleen het binnenkomende signaal versterken tot het punt dat de micro controller het weer kan detecteren als puls. Hiervoor is een versterker gemaakt die in meerdere stappen het binnenkomende signaal versterkt tot 3.3 V wat gedetecteerd kan worden als een logische '1' door de micro controller.

Om het geheel te testen is gebruik gemaakt van een oscilloscoop om de signalen te kunnen visualiseren. De eerste tests leken goed te gaan, maar de micro controller registreerde geen logische '1'.



Figuur 4.17: Eerste test van bron signaal en ontvangen signaal

In eerste instantie was niet duidelijk waarom dit het geval was, tot er beter naar het ontvangen signaal gekeken werd. Dit was geen net 'blok' van boven de 3.3 V, maar een continue golf tussen de 0 V en de 3.3 V. De micro controller interpreteert dit pas als '1' als het een net blok is. Na het raadplegen van de nodige literatuur, is een oplossing gevonden in een techniek uit de tijd van de AM-radio genaamd 'AM Diode Detector' of 'Envelope Detector' (Poole, 2012). Deze radio 's werden ook gebruikt om digitale informatie te versturen en ook daar was het belangrijk om niet een golf, maar een net vlak signaal aan de computer aan te bieden. De gebruikte oplossing zorgt voor het volgende beeld op de oscilloscoop.



Figuur 4.18: Tweede test van bron signaal en ontvangen signaal

Te zien is dat het ontvangen signaal nu netjes een blok vormt, de micro controller ziet dit signaal als een logische '1' en de data word nu correct ontvangen. Dit was een eureka moment binnen het project, de techniek werkt, nu is het zaak om te testen welke afstanden er bereikt kunnen worden.

De resultaten van het Oceans10 (Benson, 2010, p 124-127) modem bewijzen dat het mogelijk is om tot 380 meter te communiceren op een frequentie rond de 40 kHz. De eerste tests boven water tonen een maximale afstand van 10 meter bij een vermogen van 12 V op de zender. Dat is niet voldoende afstand voor dit project. Onder water zal de afstand groter zijn (geluid verplaatst zich beter door water dan door lucht), maar dat zal niet voldoende meer afstand opleveren.

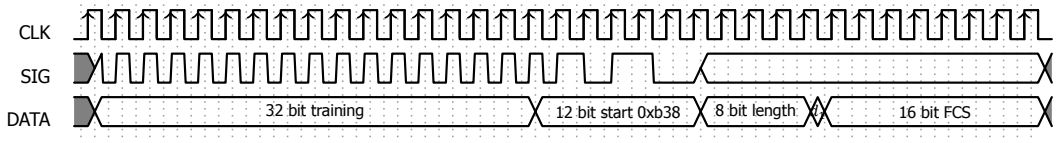
Na diverse mensen geraadpleegd te hebben, werd door een collega en oud elektrotechnicus geadviseerd om een transformator te gebruiken in de zender. Deze transformator verhoogt het voltage met een factor 10 (een 1:10 transformator). Na het raadplegen van de datasheets blijken de transducers aangestuurd te mogen worden met 110 V, dat betekent dus dat met 11 V voor de transformator het maximale voltage bereikt wordt. De beste transformator is bepaald als de K4000001 van het merk Prowave[6]. Deze transformator heeft een natuurlijke frequentie van 40 kHz en een verhouding van 1:10 (instelbaar).

Na het plaatsen van deze transformator en het verifiëren van de uitgangsspanning is het bereik in lucht gestegen tot meer dan 25 meter, meer afstand is er niet op de testlocatie. De eerste tests in het RDM test-bassin moeten een indicatie gaan geven voor de stabiliteit en afstand onder water.

[6]http://www.farnell.com/datasheets/1872315.pdf

4.4.2. Protocol

Nadat vastgesteld was dat het modem werkte, was het tijd om, binnen de limieten van de gebruikte techniek, een bruikbare set protocollen op te stellen. Zoals aangegeven is een beperking van de RadioHead (McCauley, 2008) library dat er maximaal 60 bytes per bericht verstuurd kunnen worden. Op 2000 bps duurt het verzenden van 60 bytes 384 milliseconden, zie *Implementation Details* in de RadioHead documentatie (McCauley, 2008). Binnen dit project communiceert het systeem op 200 bps, waardoor het verzenden een factor 10 langer duurt, dus 3840 milliseconden. Onderstaand is te zien hoe de berichten opgebouwd worden als er gebruik gemaakt wordt van deze library.



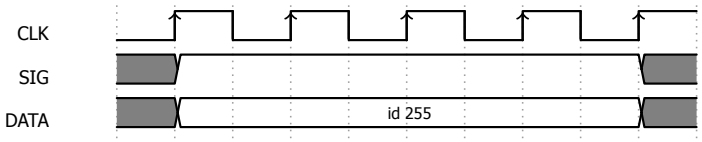
Figuur 4.19: RadioHead library bericht structuur

Het bericht begint altijd met 32 bits Hoog-laag, dit is de zogeheten *Training Preamble*, dit zorgt er effectief voor dat de zender en ontvanger in synchronisatie komen. Vervolgens wordt het bericht *0xb38* gestuurd, welke vertaalt naar de ideale oplopende combinatie van Hoog-laag (HLHHLLHHHLLL). Dit is het start signaal voor de rest van het bericht. Vervolgens wordt er in 1 byte (8 bit) de lengte van het bericht meegegeven, gevolgd door *n* bit data en als laatste de 16 bit checksum. Deze berichtstructuur zorgt voor de meest betrouwbare verzendwijze van de data, iets wat binnen dit project belangrijker is dan pure data doorvoersnelheid.

Binnen de 60 byte die ter beschikking staan kan alle data makkelijk geplaatst worden, de uitdaging is echter om de berichten zo klein mogelijk te maken. Zoals uitgelegd in hoofdstuk 4.3.5 is het belangrijk om zo snel mogelijk alle boeien een bericht te laten sturen. Daarnaast moet er ruimte zijn voor de duikers om hun positie door te geven. Gedurende de zendtijd van een boei of een duiker en korte tijd daarop mogen geen andere signalen verstuurd worden, dit zou alleen maar zorgen voor ruis en interferentie waardoor beide berichten waarschijnlijk niet goed aankomen. Effectief zijn er twee berichten te onderscheiden, de boeien die een *ping* sturen en de duikers die de positie en status willen doorgeven.

Boei bericht

Een boei geeft een zo kort mogelijk bericht op de exacte interval zoals deze overeengekomen is aan de oppervlakte (zie 4.3). Dit bericht is uitsluitend opgebouwd uit een 8 bit *identifier* voor de boei. Er kunnen op deze manier maximaal 255 boeien per duiklocatie onderscheiden worden. Daarmee is een ruime marge behouden ten opzichte van het verwachte aantal boeien per locatie en is er ruimte voor uitbreiding van het systeem. Een kleiner aantal bits heeft geen nut daar de RadioHead library altijd een hele byte reserveert voor een (deel van) een bericht.



Figuur 4.20: Boei voorbeeld bericht zonder RadioHead structuur

Uitgaande van de eerder genoemde cijfers (768 bits duurt 3840 milliseconden) kan het volgende berekend worden:

$$overhead_{bit} = 7 * 16 = 112bits$$
$$duur_{ms} = (3840/768) * (8 + 112) = 600ms$$

Figuur 4.21: Berekening voor bericht duur boeien

Duiker bericht

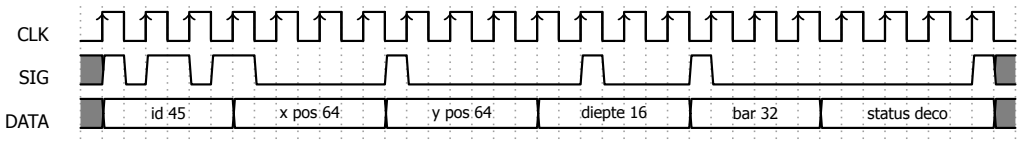
Het bericht wat door de duiker verstuurd wordt is complexer. Er is meer data die verstuurd moet worden, echter is het interval van het versturen veel kleiner. Om het bericht zo klein mogelijk te maken is besloten het duikgebied op te delen in kwadranten van 4 meter bij 4 meter. De diepte wordt als geheel getal meegegeven. Deze keuze is gemaakt enerzijds vanuit de overweging om het bericht kleiner te maken, anderzijds vanwege de inherente onnauwkeurigheid van het positiebepalingssysteem, door deze onnauwkeurigheid heeft het ook geen zin een hogere resolutie te versturen. De data die verstuurd moet worden is:

- *6 bit* duiker id (maximaal 63)
- *7 bit* x grid (maximaal 127, bereik van ongeveer 500 meter)
- *7 bit* y grid (maximaal 127, bereik van ongeveer 500 meter)
- *7 bit* diepte in m (maximaal 127)
- *6 bit* zuurstof in incrementen van 10 bar (maximaal 63, dus 630 bar)
- *8 bit* status vlaggen (bit masker)

Er kunnen dus maximaal 8 verschillende status vlaggen gedefinieerd worden in een standaard *bit mask* oplossing. Op dit moment worden de volgende statussen onderkend:

- *0x01* Decompressie limiet overschreden
- *0x02* Zuurstof niveau laag
- *0x04* Medische hulp nodig

Als alle bits uit staan dan is er dus geen bijzondere situatie met de duiker. Er zijn nog 7 bits vrij voor toekomstige uitbreiding van het systeem.



Figuur 4.22: Duiker voorbeeld bericht zonder RadioHead structuur

Ook hier kan een berekening gemaakt worden van het te verzenden bericht:

$$overhead_{bit} = 7 * 16 = 112bits$$
$$duur_{ms} = (3840/768) * (6 + 7 + 7 + 7 + 6 + 8 + 112) = 765ms$$

Figuur 4.23: Berekening voor bericht duur duiker

Hiermee is op een praktische en realistische manier invulling gegeven aan de protocollen. Hoe deze gebruikt worden in het grote systeem staat beschreven in 4.1. De gekozen *baudrate* is aan de conservatieve kant, dit geeft de mogelijkheid om deze in de toekomst te vergroten. Dat kan echter pas als de zend- en ontvangst systemen verbeterd zijn. Zie ook 6.

5

Conclusie en discussie

5.1. Conclusie

De hoofdvraag van dit project was: *Is het mogelijk om voor een realistisch bedrag een onderwater positiebepalingssysteem te ontwikkelen waar sportduikers gebruik van willen maken?*

Het antwoord daar op is niet geheel bepaald. Ja, het is mogelijk een dergelijk systeem te maken, maar we weten niet of het ook aanslaat bij de duikers zelf. Om dat te bepalen zullen we een proef moeten lanceren als het systeem beschikbaar is op een locatie. Ten aanzien van de specifieke deelvragen kunnen we het volgende vaststellen:

- **Wat is een realistisch bedrag voor de gemiddelde sportduiker?**
  We hebben kunnen vaststellen dat 750 euro een bedrag is wat gemiddeld gezien door duikers besteed wordt aan de duik computer.
- **Welke technieken zijn reeds beschikbaar?**
  Er zijn diverse onderdelen van de systemen beschikbaar, maar tegen te hoge bedragen om interessant te zijn voor de sport duiker.
- **Kunnen we betaalbare communicatie hard- en software ontwikkelen?**
  Ja, dat is mogelijk, alhoewel de werking en stabiliteit van de systemen nog wel verfijnd moet worden.
- **Werkt het positiebepaling concept?**
  Ja, het concept werkt, de nauwkeurigheid is nog te verbeteren door een stabielere communicatie en verfijning van de algoritmes.
- **Is bi-directionele communicatie mogelijk?**
  Ja, het systeem kan zowel zenden als ontvangen. Wel is het belangrijk vast te stellen dat we het hier over synchrone communicatie hebben, het is niet mogelijk om tegelijkertijd te zenden en te ontvangen. Mogelijk kan dit in de toekomst opgelost worden door andere frequenties te gebruiken zodat deze elkaar niet verstoren.
- **Hebben we te maken met conflicterende wetgeving?**
  Wetgeving is er niet specifiek voor dit systeem. Er is een voorstel gemaakt binnen de CHRRB groep wat de basis moet vormen voor regulering. Vanuit Rijkswaterstaat is toestemming verleend om bestaande boeien te gebruiken en om, waar nodig, nieuwe boeien te plaatsen. Dit was 1 van de grootste struikelblokken.
- **Welke opslag- en verwerkingstechnieken kunnen we gebruiken in het backend platform?**
  De architectuur van het backend platform is bepaald en grotendeels geïmplementeerd, de opzet is schaalbaar en maakt gebruik van technieken waarmee we deze op diverse besturingssystemen kunnen inzetten.

- **Hoe zetten we het systeem in de markt?**
  Er is gekozen voor een Open Source implementatie van het grootste deel van de hard- en software die ontwikkeld is. Daardoor kunnen we het systeem eenvoudiger in de markt plaatsen en zal een wijdse verspreiding eenvoudiger zijn dan bij een puur commerciële exploitatie van het systeem.
- **Hoe zorgen we voor een solide bedrijfsmodel?**
  Naast de Open Source hard- en software hebben we een aantal inkomstenstromen benoemd die kunnen zorgen voor een stabiele cashflow binnen het bedrijf. Op basis van die inkomstenstromen kunnen we het platform en ons productaanbod verder vergroten.
- **Willen duikers gebruik maken van het systeem?**
  Op deze vraag is nog geen concreet antwoord gegeven. Het systeem moet eerst verder ontwikkeld worden voordat we het aan kunnen bieden aan bijvoorbeeld testgroepen. De reacties op het systeem zijn echter zeer positief waardoor het zeer waarschijnlijk is dat er zeker een markt voor bestaat.

5.2. Discussie

Ik beschouw deze scriptie niet als het einde van het project, wel als het succesvol afsluiten van een belangrijke periode in het project. Ik heb een hoop bereikt door deze scriptie als doel te stellen en daar specifieke onderzoeksvragen aan op te hangen. De zaken die in deze scriptie besproken zijn vormen een solide opstap naar een echte implementatie van het systeem. De trajecten met de Nederlandse Onderwatersport Bond en de Provincie Zeeland lopen en binnenkort verwachten we de eerste echte uitrol van het systeem te doen in Scharendijke, 1 van de top duik locaties in Zeeland. Het doel om in de toekomst duikers te redden of terug te vinden lijkt daarmee steeds dichterbij te komen.

Het is echter ook niet zo dat ik op mijn lauweren kan gaan rusten. Het is nu, meer dan ooit, belangrijk om de voortgang van het project op peil te houden. Ik ben druk bezig om extra mensen in het project te verzamelen die specifieke expertise toe kunnen voegen aan de belangrijke onderdelen. Van daaruit moeten we de in hoofdstuk 6 genoemde vervolgstappen realiseren.

Naast de concrete realisatie van die vervolgstappen is het eveneens belangrijk om te bepalen of dit systeem aan gaat slaan bij de sportduikers. Vanuit de overheid is er zeker interesse, maar het systeem is natuurlijk allereerst bedoeld om levens te redden onder de sportduikers. Tijdens het schrijven van deze scriptie zijn er meerdere duikers overleden en zijn er langdurige zoekacties geweest om de lichamen te bergen. Dit systeem had, in een aantal situaties, de gevolgen van de ongelukken kunnen verzachten en in alle gevallen het bergen van de lichamen kunnen bespoedigen. Het is belangrijk het systeem aan te bieden aan een testgroep en om een marktonderzoek uitvoeren. Eventueel kunnen we nog bijsturen in de concrete functionaliteiten van het systeem zodat de kosten verder gereduceerd worden.

6

Vervolg

In dit hoofdstuk worden de vervolgstappen besproken die gemaakt gaan worden met het systeem.

6.1. Doorontwikkeling

De basiscomponenten van het systeem werken voldoende om het bewijs te leveren dat het gehele concept werkt, echter is het in deze toestand nog niet geschikt voor een brede implementatie. Het idee is om, in samenwerking met de partners, de verschillende componenten verder door te ontwikkelen tot op het punt dat alle onderdelen geschikt zijn voor een standaard eindgebruiker. Onderstaand is uitgewerkt wat de eisen zijn aan de verschillende systemen om een publieke release te kunnen overwegen.

- Het modem moet stabiel communiceren over een afstand van 250 meter.
- De hardware voor de duikers moet eenvoudig te bedienen en onderhouden zijn.
- De infrastructuur voor het backend moet stabiel publiek beschikbaar zijn.
- Het backend systeem moet ook op afgelegen locaties bereikbaar zijn voor de boeien.
- De positiebepaling moet voldoende stabiel zijn met een afwijking van maximaal 5 meter.
- De boeien moeten makkelijk te plaatsen en te onderhouden zijn.
- De backend moet ontsloten worden via een website en een mobiele applicatie welke eenvoudig te gebruiken is.

6.2. Commerciële exploitatie

In principe is het hoofddoel van het project altijd geweest 'zorgen dat het er komt', het is nooit een doel geweest om er geld mee te verdienen. Het uitgangspunt blijft dat het bestaan van het systeem en het verbreden van de implementatie het belangrijkste doel is. Er zijn echter wel kansen om op een aantal punten geld te kunnen verdienen. Dit geld is dan primair bedoeld voor verdere ontwikkeling van dit systeem en aanvullende concepten vanuit scubAssist. De hardware blijft Open Source en is niet exclusief voor scubAssist, daarmee wordt hopelijk een gezonde concurrentie gecreeërd en werkt dit bredere verspreiding in de hand.

- De verschillende hardware componenten worden voor net boven kostprijs aangeboden in de online scubAssist winkel.
- Complete geassembleerde sets voor duikers worden aangeboden via de online scubAssist winkel voor consumenten die het systeem niet zelf willen bouwen.
- Complete geassembleerde boeien worden aangeboden via de online scubAssist winkel voor instanties die deze zelf niet willen bouwen.

Aangenomen dat de marge op de hardware minimaal zal zijn om te blijven voldoen aan het Open Source gedachtegoed is het belangrijk om alternatieve inkomstenbronnen te vinden waarmee wel voldoende verdiend kan worden om er een gezond bedrijf op na te houden. Er is besloten de activiteiten vooral te richten op het leasen van het systeem aan (overheids) partners die zorgenvrij gebruik willen maken van het systeem. Er zullen jaarlijkse tarieven vastgesteld worden voor plaatsing en support van de systemen op permanente locaties en voor de levering en onderhoud van de ad hoc systemen.

- scubAssist biedt plaatsings- en onderhoudscontracten aan voor organisaties die het systeem geplaatst willen hebben in een duikgebied.
- scubAssist biedt onderhoudscontracten aan aan organisaties die het ad hoc systeem willen gebruiken zoals overheidsinstellingen.

Als derde inkomstenbron is er de community website. Zoals eerder aangegeven kunnen bedrijven op *relevante* plekken binnen deze website adverteren. Daarbij moet zorg gedragen worden dat dit geen negatieve invloed heeft op de ervaring van de gebruikers, maar juist een positieve bijdrage levert aan de functionaliteit en het gebruiksgemak van het systeem. Om dat te bewerkstelligen kunnen de gebruikers winkels voorzien van beoordelingen, welke invloed hebben op de plaatsing van deze bedrijven bij bijvoorbeeld een duiklocatie. Bedrijven met een betere beoordeling zullen hoger in de lijst getoond worden. Op die manier worden de gebruikers voorzien van relevante informatie in plaats van storende en mogelijk onjuiste advertenties.

- Bedrijven kunnen adverteren op relevante plekken binnen de community website.
- Bedrijven kunnen zelf aangeven wat hun budget en doelgroep is om zo gerichter te adverteren.

Figuur 6.1: Business Model Canvas voor scubAssist

7

Reflectie

7.1. Reflectie

Ik heb in dit traject vele aspecten van de *Telematica* doorlopen. Hiermee heb ik de kennis die ik reeds in de praktijk had opgedaan verbreed met een aantal aspecten waar ik in mijn 'normale' werk niet, of heel weinig, mee in aanraking kom. Daarnaast heb ik een aantal aan de Informatica grenzende vakgebieden ontdekt, zoals Elektrotechniek en Engineering. Ik heb met veel organisaties en personen gesproken over het concept en de realisatie, hiermee heb ik veel ervaring opgedaan op het gebied van Communicatie, een aspect waar ik tot nog toe vooral intern mee te maken had. Concluderend heb ik de volgende aspecten geraakt:

Planning

Situatie Er moest een planning gemaakt worden om binnen de gestelde 20 weken tot een scriptie en een meetbaar eindresultaat te komen.
Taak Het was aan mij om een logische planning op te leveren gebruik makend van een correcte project methodiek.
Actie Ik heb in het mandaat een 20 weken planning gemaakt waarbij ik de benodigde acties benoemd heb en deze heb ondergebracht in *timeboxes* en *sprints*, dit conform Agile SCRUM.
Resultaat Ik had een duidelijke leidraad gedurende het traject en was voldoende flexibel om bij te sturen door resultaten uit eerdere sprints. Verder voorkwam ik te lang doorwerken aan 1 onderdeel door duidelijke timeboxes te hanteren.
Reflectie Ik kende de Agile SCRUM methodiek reeds via mijn werk, daar werken we echter niet veel met timeboxes, ik kan deze ervaring ook weer op het werk toepassen en ben zeer tevreden met het verloop van het afstudeerproject.

Communicatie

Situatie Op diverse momenten in het project zijn er contact momenten geweest met (potentiële) stakeholders en sympathisanten.
Taak Voor elk van deze gesprekken moest ik op de juiste manier de juiste informatie overdragen om zo mensen te enthousiasmeren.
Actie Ik heb goed gekeken naar de wensen van de verschillende doelgroepen en heb de communicatie vorm en hulpmiddelen daar op aangepast.
Resultaat Alle gesprekken die ik gevoerd heb zijn positief. Dat komt ook deels door het project zelf, maar ook omdat ik iedere keer in staat ben gebleken de informatie op een juiste manier over te brengen.
Reflectie De visuals hebben heel erg geholpen bij het overdragen van het concept op de minder technisch onderlegde doelgroepen. Ik wil naar de toekomst toe, meer van deze visuals maken om de concepten over te brengen.

Architectuur

Situatie Voor het buddyAssist systeem moesten op diverse niveau 's architecturen ontworpen worden van de systemen.

Taak Voor elk van de onderdelen en het overkoepelende systeem was ik verantwoordelijk voor het opzetten van een passende architectuur.
Actie Ik heb geput uit mijn ervaring als Software Architect bij mijn werkgever en veel literatuuronderzoek gedaan naar die onderdelen waar ik zelf geen directe ervaring mee had.
Resultaat De gebruikte architecturen vormen samen een werkbaar geheel.
Reflectie Er is verbetering mogelijk bij een aantal componenten, hierbij denk ik dan voornamelijk aan optimalisaties in de firmware van het modem en de boeien, maar over het algemeen ben ik zeer tevreden over hoe ik in staat ben gebleken ook zaken die buiten mijn vakgebied liggen samen te brengen tot een functioneel geheel.

Security

Situatie De communicatie tussen de diverse onderdelen van het buddyAssist systeem moesten voorzien zijn van adequate beveiliging.
Taak Het bepalen van de potentiële probleemgebieden en de juiste beveiligingsmaatregelen.
Actie Ik heb in kaart gebracht waar data gecommuniceerd wordt in het overkoepelende systeem en daar oplossingen bij gezocht welke ik getoetst heb aan OWASP [1].
Resultaat Het systeem voldoet voor de oppervlakte communicatie aan de gestelde security eisen. Het onderwater aspect is niet voorzien van security.
Reflectie Ook hier heb ik gebruik gemaakt van mijn ervaringen bij mijn werkgever, oAuth2.0/OpenID zijn standaarden die we in diverse stukken software implementeren en ook hier toetsen we de software aan de OWASP bepalingen. Voor het onderwater communicatie aspect heb ik helaas nog geen goede oplossing gevonden. Het beveiligen van de onderwater communicatie word opgepakt als deze voldoende stabiel functioneert.

Business

Situatie Er moest een bedrijfsplan opgesteld worden voor scubAssist om het buddyAssist systeem aan te kunnen bieden en de verdere ontwikkeling te kunnen bekostigen.
Taak Als eigenaar van scubAssist was het mijn taak om een kloppend bedrijfsmodel te bepalen.
Actie Ik heb geredeneerd vanuit de kerngedachte, namelijk het Open Source ontwikkelen van de harden software. Daaromheen heb ik diverse potentiële inkomstenbronnen gedefiniëerd.
Resultaat Het bedrijfsplan biedt een basis voor een gezonde exploitatie van het systeem.
Reflectie Door goed te kijken naar de bepaalde en verwachte doelgroepen is er een goede basis neergelegd. Ik denk wel dat het zaak is deze continue te toetsen naarmate het systeem volwassener wordt. Er zijn potentieel nog veel inkomstenbronnen beschikbaar die zich nu nog niet hebben openbaart.

Multidisciplinair

Situatie Tijdens het bedenken van het buddyAssist systeem zijn veel disciplines buiten de Informatica geraakt.
Taak Als bedenker en ontwikkelaar van het systeem was ik verantwoordelijk voor het verzamelen van de juiste kennis en expertise.
Actie Ik heb bij alle verschillende onderdelen goed gekeken naar de beschikbare informatie en bepaald of ik zelf in staat was op basis van die informatie tot een correct resultaat te komen. Daarnaast heb ik diverse mensen met een hogere expertise benaderd en deze bij het project betrokken.
Resultaat Alle basis onderdelen van het systeem functioneren. Ook de omliggende zaken zoals Marketing en het bepalen van een correct bedrijfsmodel zijn naar tevredenheid afgerond.
Reflectie Ik ben een T-Shape, ik doe en weet graag alles zelf. Ik heb gedurende dit project ook een hoop kennis en ervaring opgedaan, wel accepteer ik nu sneller dat sommige onderdelen beter ontwikkeld kunnen worden door mensen die daar een diepgaande kennis van hebben. Ik ben nu echter wel in staat om over de verschillende disciplines mee te denken in de oplossingsrichtingen.

[1] https://www.owasp.org/index.php/Web_Application_Security_Testing_Cheat_Sheet

Bibliografie

Benson, B. (2010). *Design of a low-cost underwater acoustic modem for short- range sensor networks* (Academisch proefschrift, University of California, Sn Diego). Verkregen van http://escholarship.org/uc/item/8j70s26c#page-1

Burns, K. (2013). *A software defined underwater acoustic communications hardware platform* (Academisch proefschrift, Flinders University, Adelaide Australia). Verkregen van http://traktoria.org/files/sonar/underwater_communication/a_software_defined_underwater_acoustic_communications_hardware_platform__(thesis-final)K.Burns.pdf

Cottrell, L. (2012, 04). *Tulip algorithm alternative trilateration method.* Confluence. Verkregen van https://confluence.slac.stanford.edu/display/IEPM/TULIP+Algorithm+Alternative+Trilateration+Method

Entwicklung tauchcomputer. (2012). Op 2015-10-03 verkregen van http://tauchen.seveke.de/altes/1dekometer.htm

Freitag, L., Grund, M., Singh, S., Partan, J., Koski, P. & Ball, K. (2005). The whoi micro-modem: an acoustic communications and navigation system for multiple platforms. In *Oceans, 2005. proceedings of mts/ieee* (pp. 1086–1092). Verkregen van ftp://soest.hawaii.edu/bhowe/outgoing/AMM/ALOHA_Mooring/ALOHA%20Mooring/Sensors/Micro%20Modem/MicromodemOceans2005.pdf

Lanbo, L., Shengli, Z. & Jun-Hong, C. (2008). Prospects and problems of wireless communication for underwater sensor networks. *Wireless Communications and Mobile Computing*, *8*(8), 977–994.

McCauley, M. (2008). *Radiohead library.* Verkregen van http://www.airspayce.com/mikem/arduino/RadioHead/classRH__ASK.html

Poole, I. (2012). *Am diode detector.* Website. Verkregen van http://www.radio-electronics.com/info/rf-technology-design/am-reception/diode-detector-demodulator.php (Envelope detector circuit)

Telecommunicatiewet. (2016). wetten.nl. Verkregen van http://wetten.overheid.nl/BWBR0009950/2016-01-01

van Kleunen, W. A. P. (2014). *Echoes from the deep - communication scheduling, localization and time-synchronization in underwater acoustic sensor networks* (Academisch proefschrift, University of Twente, Enschede, the Netherlands). Verkregen van http://doc.utwente.nl/91104/

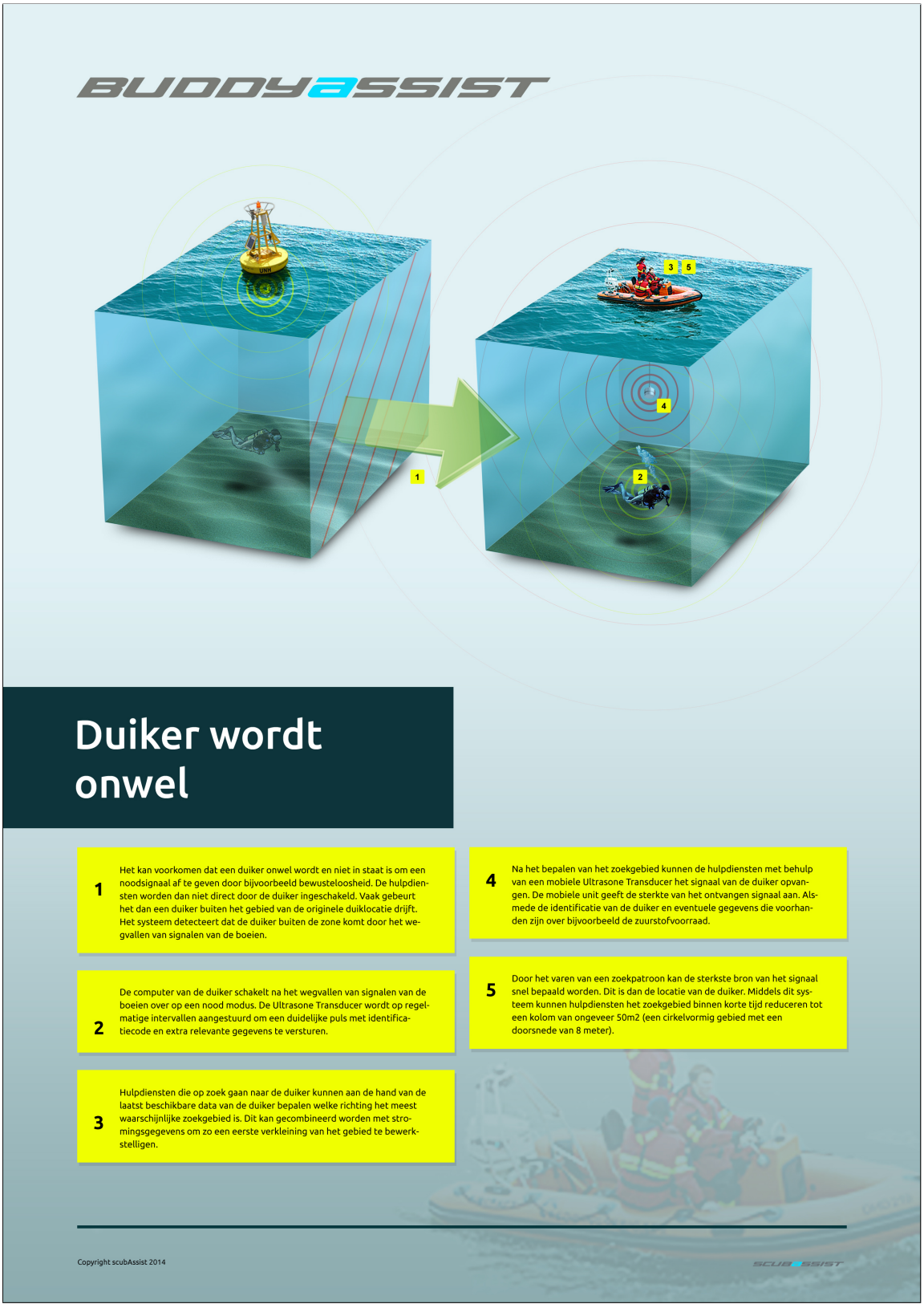
A

Appendix Visuals

Op de volgende bladzijden worden de visuals getoond die gebruikt zijn in de communicatie met de (potentiële) stakeholders.

Figuur A.1: Scenario 1 - Visual Werking systeem

Figuur A.2: Scenario 2 - Visual Noodsituatie duiker

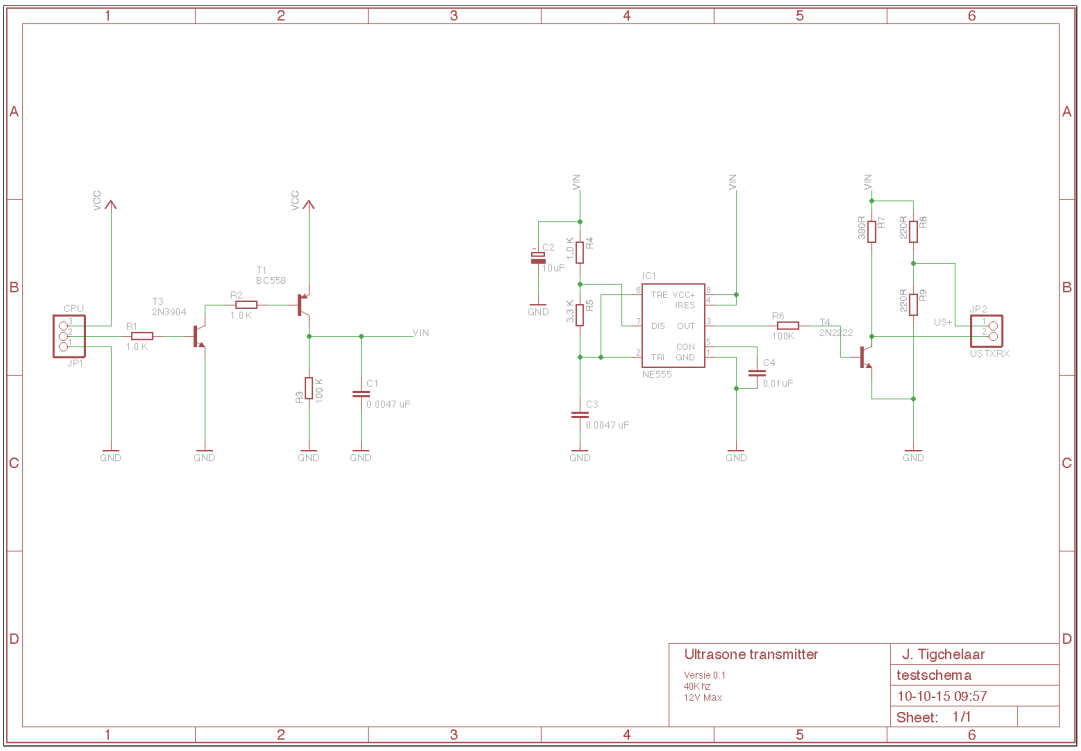


Figuur A.3: Scenario 3 - Visual Duiker onwel

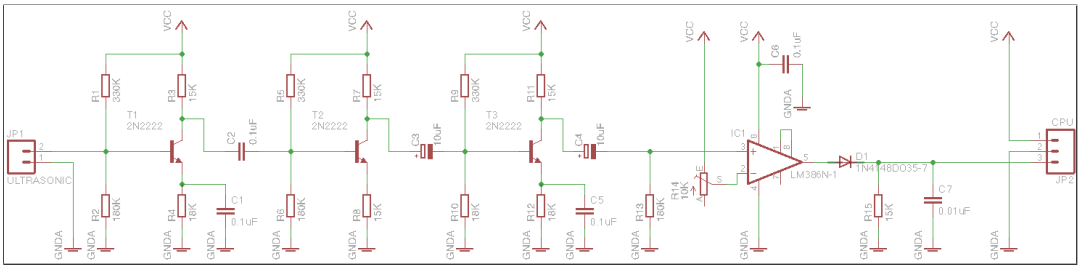
B

Appendix Hardware Schema's

De laatste schema's zijn terug te vinden op https://github.com/scubAssist/hardware.



Figuur B.1: Schema van het zend gedeelte van het ultrasone modem



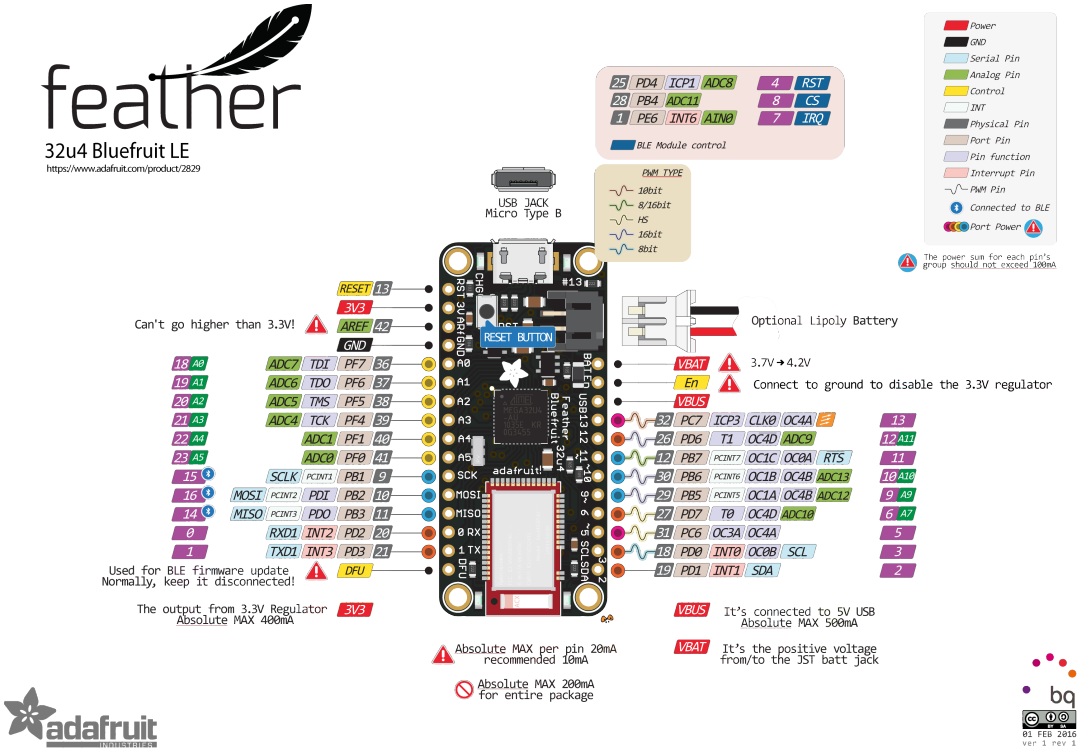
Figuur B.2: Schema van het ontvanger gedeelte van het ultrasone modem

C

Appendix Code

C.0.1. Modem

De volledige firmware van het modem is te vinden op https://github.com/scubAssist/software.



Figuur C.1: Het pinout schema van de gebruikte microcontroller.

In onderstaande code segment wordt het initialiseren van de interrupts op de Arduino microcontroller van het modem getoond. De code voor het modem is zo opgezet dat deze in het systeem van de duiker gebruikt kan worden alsmede in het systeem van de boeien. Dit wordt middels seriële commando 's bij het starten van de systemen ingesteld.

```
1 // D2 interrupt RTC
2 // D3 interrupt RF
3
4 void setup()
5 {
6   // Init pins
7   pinMode(rf_transmit_pin , OUTPUT);
8   pinMode(rtc_interrupt_pin , INPUT_PULLUP);
9   pinMode(rf_interrupt_pin , INPUT_PULLUP);
```

```
10
11   // Interrupts aanzetten
12   attachInterrupt(digitalPinToInterrupt(rtc_interrupt_pin), rtc_interrupt, RISING);
13   attachInterrupt(digitalPinToInterrupt(rf_interrupt_pin), rf_interrupt, RISING);
14 }
```

De aangeroepen interrupt methode als het RTC alarm afgaat is als volgt. Hierbij kan het modem in gebruik zijn in een boei (is_buoy), waarbij hij ook de hoofd boei kan zijn (is_master), of het modem is in gebruik in de duiker eenheid (is_diver).

```
1  void rtc_interrupt()
2  {
3    if (is_diver) {
4      zero_milliseconds = millis();
5
6      // Random status send interval
7      //send at random interval
8      driver.send((uint8_t *)diverstatus, sizeof(PayloadDiver));
9      driver.waitPacketSent();
10   }
11
12   if (is_buoy)
13   {
14     if (is_master)
15     {
16       // Send RF pulse every specified interval
17       if (frameCounter == 0)
18       {
19         digitalWrite(rf_transmit_pin, HIGH);
20         delayMicroseconds(1000); // Should work in ISR
21         digitalWrite(rf_transmit_pin, LOW);
22       }
23     }
24
25     // Send pulse when our ID equals the zero pulse + ID * seconds
26     if (frameCounter == buoy->id)
27     {
28       driver.send((uint8_t *)buoy, sizeof(PayloadBuoy));
29       driver.waitPacketSent();
30     }
31   }
32
33   // Always increment the framecounter
34   frameCounter++;
35
36   // if framecounter reaches max, reset to 0 (assume 16 frames, 0-15 = 16 frames, reset at
       16)
37   if (frameCounter == zero_pulse_reset)
38   {
39     frameCounter = 0;
40   }
41 }
```

Voor het ontvangen van de RF synchronisatie geldt de onderstaande methode.

```
1  void rf_interrupt()
2  {
3    // update rtc with zero seconds
4    setTime(17, 6, 0, 11, 4, 2016); // not-so-arbitrary date
5    rf_synched = true;
6    frameCounter = 0; // Only occurs on zero reset, so reset the frame counter as well
7  }
```

C.0.2. Simulatie

De volledige code van de simulator is te vinden op https://github.com/scubAssist/simulator.

Het volgende code segment toont (een deel van) de BAPE berekening. In dit code segment is te zien hoe BAPE alle gemeten afstanden gebruikt om te kijken welke kubussen 'op de lijn' vallen. Bij het snel stijgen van het aantal mogelijke deel kubussen stopt de berekening (regel 26), we zijn dan zo nauwkeurig mogelijk.

```
1  public static class LaggedPositionEstimate {
2    public static float timeDrift = 0.003;
3    public static ArrayList<PositionRange> calculate(HashMap<String, PingCollision>
       pingRegister, float now) {
4
5      //get latest ping
6      float maxT = 0;
7      PingCollision latestPing = null;
8      for (Map.Entry<String, PingCollision> ping : pingRegister.entrySet()) {
9        if (maxT == 0 || maxT > (ping.getValue()).timeCollision) {
10         maxT = (ping.getValue()).timeCollision;
11         latestPing = ping.getValue();
12       }
13     }
14
15     if (latestPing!=null) {
16       float y = latestPing.depth;
17       float yd = (now - latestPing.timeCollision) * timeDrift;
18       float i = 0;
19       ArrayList<PositionRange> posiblePositions = new ArrayList();
20       ArrayList<PositionRange> removePositions = new ArrayList();
21       posiblePositions.add(new PositionRange(-1000, 1000, y-yd, y+yd, -1000, 1000));
22
23       boolean loop = true;
24       while (loop) {
25
26         if (i > 64 || posiblePositions.size()> 12) {
27           return posiblePositions;
28         }
29
30         removePositions.addAll(posiblePositions);
31         posiblePositions.clear();
32         for (PositionRange p2 : removePositions) {
33           posiblePositions.addAll(p2.cubeSplit());
34         }
35         removePositions.clear();
36
37         for (Map.Entry<String, PingCollision> ping : pingRegister.entrySet()) {
38           for (PositionRange p2 : posiblePositions) {
39             if (!collisionCheck(p2, ping.getValue(), now)) {
40               removePositions.add(p2);
41             }
42           }
43           if (posiblePositions.size() != removePositions.size()) {
44             posiblePositions.removeAll(removePositions);
45           }
46           removePositions.clear();
47         }
48
49         i++;
50       }
51     }
52     return null;
53   }
54
55   public static boolean lineCircleCollision(PVector point1, PVector point2, PVector
       circleCenter, float radius, float margin) {
56     boolean collision = false;
57
58     //collide inner
59     if (   (point1.dist(circleCenter) < radius - margin && point2.dist(circleCenter) > radius
        - margin)
```

```
60        || (point1.dist(circleCenter) > radius - margin && point2.dist(circleCenter) < radius -
          margin))
61      {
62        return true;
63      }
64
65      //collide outer
66      if (   (point1.dist(circleCenter) < radius + margin && point2.dist(circleCenter) > radius
          + margin)
67        || (point1.dist(circleCenter) > radius + margin && point2.dist(circleCenter) < radius +
          margin))
68      {
69        return true;
70      }
71
72      //collide
73      if (      point1.dist(circleCenter) > radius - margin
74        && point2.dist(circleCenter) > radius - margin
75        && point1.dist(circleCenter) < radius + margin
76        && point2.dist(circleCenter) < radius + margin)
77      {
78        return true;
79      }
80
81      return false;
82    }
83
84    public static boolean collisionCheck(PositionRange range, PingCollision p, float now) {
85      float radiusAtDepth = pow(pow(p.distance, 2) - pow(p.depth, 2), 0.5);
86
87      PVector p1 = new PVector(range.x1, range.z1);
88      PVector p2 = new PVector(range.x2, range.z1);
89      PVector p3 = new PVector(range.x2, range.z2);
90      PVector p4 = new PVector(range.x1, range.z2);
91
92      //valt de ping helemaal in de range
93      if (p1.dist(new PVector(p.origin.x, p.origin.z)) > radiusAtDepth
94        && p2.dist(new PVector(p.origin.x, p.origin.z)) > radiusAtDepth
95        && p3.dist(new PVector(p.origin.x, p.origin.z)) > radiusAtDepth
96        && p4.dist(new PVector(p.origin.x, p.origin.z)) > radiusAtDepth
97
98        && p.origin.x > range.x1
99        && p.origin.x < range.x2
100       && p.origin.z > range.z1
101       && p.origin.z < range.z2
102     ) {
103       return true;
104     }
105
106     //of 1 van de edges
107     if ( lineCircleCollision(p1, p2, new PVector(p.origin.x, p.origin.z), radiusAtDepth, (now
          - p.timeCollision) * timeDrift)) {
108       return true;
109     }
110     if ( lineCircleCollision(p2, p3, new PVector(p.origin.x, p.origin.z), radiusAtDepth, (now
          - p.timeCollision) * timeDrift)) {
111       return true;
112     }
113     if ( lineCircleCollision(p3, p4, new PVector(p.origin.x, p.origin.z), radiusAtDepth, (now
          - p.timeCollision) * timeDrift)) {
114       return true;
115     }
116     if ( lineCircleCollision(p4, p1, new PVector(p.origin.x, p.origin.z), radiusAtDepth, (now
          - p.timeCollision) * timeDrift)) {
117       return true;
118     }
119
120     return false;
121   }
122 }
```