

# Integratie van draadloze aspecten

## Hogeschool Utrecht

---

Onderzoeksverslag



|                       |                                   |
|-----------------------|-----------------------------------|
| <b>Auteur</b>         | <i>René de Kluis</i>              |
| <b>Student nummer</b> | <i>1661627</i>                    |
| <b>Datum</b>          | <i>06/01/2019</i>                 |
| <b>Locatie</b>        | <i>Heidelberglaan 15, Utrecht</i> |
| <b>Versie</b>         | <i>V1.0.0</i>                     |
| <b>Status</b>         | <i>Eerste Oplevering</i>          |

## Versiebeheer

De versie van dit document is genummerd door middel van drie getallen X, Y en Z. Hierbij staat de X voor de uitgave versie en Y voor de uitgebrachte conceptversie. Veranderingen van deze versies zullen opgenomen worden in de onderstaande tabel. De Z voor elke aanpassing dat aangebracht binnen de huidige conceptversie.

| Versie  | Datum      | Beschrijving   |
|---------|------------|--|
| v0.0.1  | 11/10/2018 | Aanmaak van het document   |
| V0.0.2  | 13/12/2018 | Eerste opzet van het document  |
| V0.0.3  | 18/12/2018 | Inleiding uitgeschreven  |
| V0.0.4  | 19/12/2018 | Keuze microcontroller uitgeschreven  |
| V0.0.5  | 27/12/2018 | Invulling §5.1   |
| V0.0.6  | 30/12/2018 | Verandering van de opzet<br>Invulling §5.2, §5.3                             |
| V0.0.7  | 02/01/2019 | Invulling §4.1   |
| V0.0.8  | 03/01/2019 | Invulling §4.2, §4.3<br>Invulling Conclusie                                  |
| V0.0.9  | 03/01/2019 | Invulling voorwoord  |
| V0.0.10 | 03/01/2019 | Opzet managementsamenvatting   |
| V0.0.11 | 04/01/2019 | Invulling advies, evaluatie<br>Toevoeging, Appendix D – Xtensa LX6 Assembler |
| V0.0.12 | 05/01/2019 | Invulling Management samenvatting<br>Spellings- en grammatica check          |
| V1.0.0  | 06/01/2019 | Eerste Oplevering  |

## Voorwoord

Voor u ligt het onderzoeksverslag “Integratie van draadloze aspecten”. Het onderzoek voor dit verslag is uitgevoerd bij de Hogeschool Utrecht. Dit verslag is geschreven in het kader van mijn onderzoek semester dat deel uit maakt van het vierde jaar van de afstudeerrichting Technische Informatica van HBO-ICT. Het onderzoek is gedaan in opdracht van de Hogeschool Utrecht. Van september 2018 tot en met januari 2019 ben ik bezig geweest met het onderzoek en het schrijven van het verslag.

Samen met mijn begeleider, Wouter van Ooijen, heb ik een opdracht voor het onderzoek bedacht en na goedkeuring van het begeleidend docent, Huib Aldewereld, uitgevoerd.

Bij dezen wil ik graag mijn begeleider bedanken voor de begeleiding en zijn ondersteuning tijdens dit traject. Ook wil ik mijn vriendin bedanken voor het nalezen van het verslag op spellings- en grammatica fouten.

René de Kluis

IJsselstein, 5 januari 2019

## Managementsamenvatting

Aangezien Internet of Things toepassingen steeds vaker voor komen in onze maatschappij, wil de Hogeschool Utrecht dit opnemen in het curriculum Technische Informatica van HBO-ICT. Alleen wordt bij deze opleiding gewerkt met de Arduino Due. Dit is een microcontroller waarmee studenten moeten programmeren. Deze microcontroller heeft geen geïntegreerde Wifi of Bluetooth modules, waardoor draadloze aspecten niet in de huidige cursussen van Technische Informatica voor komen.

De doelstelling van dit onderzoek is binnen het curriculum Technische Informatica van de Hogeschool Utrecht een nieuwe microcontroller te implementeren, waarbij een Wifi en/ of Bluetooth module geïntegreerd zit. Aangezien de studenten werken met libraries en tools van die door de Hogeschool Utrecht gemaakt zijn, zal de aansturing van de microcontroller hier ook in opgenomen dienen te worden. Hierdoor kunnen draadloze communicatie aspecten toegevoegd worden aan de huidige opgaven van Technische Informatica. Voor het onderzoek is de volgende onderzoeksvraag opgesteld: *“Hoe kan een nieuwe microcontroller met geïntegreerde wifi en/of bluetooth in de huidige libraries en tools van de Hogeschool Utrecht opgenomen worden, zodat draadloze communicatie aspecten toegevoegd kunnen worden in de opgaven van Technische Informatica?”*

Voor het beantwoorden van de onderzoeksvraag is onderzoek gedaan naar verschillende onderdelen. Eerst is onderzoek gedaan naar verschillende microcontrollers die Wifi en/ of Bluetooth ondersteunen. Daarna is onderzoek gedaan naar de opgaven van het curriculum Technische Informatica en de mogelijkheden waar draadloze communicatie aspecten toegevoegd konden worden. Als laatste is gekeken naar de werking en integratiemogelijkheden van de huidige Libraries en tools van de Hogeschool Utrecht. Het bleek dat het opnemen van de nieuwe microcontroller kan gebeuren door de aansturing te implementeren in de BMPTK make tool. Daarnaast moet een Wifi en/ of Bluetooth library in HWLIB toegevoegd worden. Alleen bleek de integratie met de RTOS library momenteel niet mogelijk. Hierbij kan wel de keuze gemaakt worden om bij de esp32 FreeRTOS te gebruiken, wat geïntegreerd zit in de huidige aansturing van deze microcontroller. Hierdoor kunnen nog steeds alle opdrachten van het curriculum Technische Informatica gerealiseerd worden en draadloze communicatie aspecten toegevoegd worden.

Op basis van dit onderzoek wordt aanbevolen om de esp32 te implementeren bij alle cursussen van Technische Informatica, waarbij huidig de Arduino Due gebruikt wordt. Dit zorgt namelijk voor Uniformiteit bij de opleiding. Daarnaast wordt aanbevolen om de bestanden in de BMPTK make tool beter te documenteren. Het werken met deze bestanden is zeer lastig, aangezien het niet duidelijk is waarvoor bepaalde vlaggen staan en met welke data deze gevuld dienen te worden. Als laatste kan verder onderzoek gedaan worden naar de aansturing van de esp32 en op welke manier FreeRTOS hierin geïntegreerd zit. Dit geeft de mogelijkheid om de RTOS library van de Hogeschool Utrecht op te nemen in de aansturing van de esp32 in plaats van FreeRTOS.

## Inhoudsopgave

|    |   |    |
|----|---|----|
| 1. | Inleiding.....                                  | 8  |
|    | §1.1 Organisatorische Context.....              | 8  |
|    | §1.1.1 Relatie tot andere projecten.....        | 8  |
|    | §1.1.2 Contact.....                             | 9  |
|    | §1.2 Probleemstelling .....                     | 10 |
|    | §1.2.1 Kwestie.....                             | 10 |
|    | §1.2.2 Vraagstelling.....                       | 10 |
| 2. | Theoretisch kader .....                         | 11 |
|    | §2.1 Kernbegrippen .....                        | 11 |
|    | §2.1.1 Herbruikbaarheid van libraries .....     | 11 |
|    | §2.1.2 Hardware abstractie .....                | 11 |
| 3. | Keuze Microcontroller .....                     | 12 |
|    | §3.1 Microcontrollers met Bluetooth .....       | 12 |
|    | §3.2 Microcontroller met Wifi .....             | 13 |
|    | §3.3 Conclusie .....                            | 14 |
| 4. | Curriculum Technische Informatica .....         | 15 |
|    | §4.1 Cursussen .....                            | 15 |
|    | §4.1.3 TICT-V1IDP-15 .....                      | 15 |
|    | §4.1.1 TICT-V1OOPC-15 .....                     | 16 |
|    | §4.1.2 TICT-IPASS-15 .....                      | 16 |
|    | §4.1.4 TCTI-V2CPSE1-16.....                     | 16 |
|    | §4.1.5 TCTI-V2THDE-16 .....                     | 16 |
|    | §4.1.7 TCTI-V2MRB-14 .....                      | 17 |
|    | §4.1.6 TCTI-R2D2-17 .....                       | 17 |
|    | §4.2 Implementatie van draadloze aspecten ..... | 18 |
|    | §4.3 Conclusie .....                            | 18 |
| 5. | Libraries en tools.....                         | 19 |
|    | §5.1 BMPTK.....                                 | 19 |
|    | §5.1.1 Werking.....                             | 19 |
|    | §5.1.2 ESP-IDF .....                            | 20 |
|    | §5.1.3 Conclusie .....                          | 22 |
|    | §5.2 HWLIB.....                                 | 23 |
|    | §5.3 RTOS.....                                  | 24 |

|  |    |
|--|----|
| §5.3.1 FreeRTOS.....                             | 25 |
| 6. Conclusie & Aanbeveling.....                  | 26 |
| §6.1 Aanbeveling.....                            | 27 |
| §6.1.1 RTOS.....                                 | 27 |
| §6.1.2 Uniformiteit.....                         | 27 |
| §6.1.3 Assembler.....                            | 27 |
| §6.1.4 Documentatie .....                        | 27 |
| 7. Evaluatie.....                                | 28 |
| §7.1 Knelpunten.....                             | 28 |
| §7.2 Leerpunten.....                             | 28 |
| §7.3 Conclusie .....                             | 29 |
| 8. Bibliografie .....                            | 30 |
| 9. Appendix .....                                | 31 |
| Appendix A – Plan van Aanpak .....               | 32 |
| 1 Inleiding.....                                 | 34 |
| 2 Organisatorische context .....                 | 35 |
| 3 Probleemstelling .....                         | 36 |
| 4 Theoretisch kader .....                        | 38 |
| 6 Planning & Aanpak.....                         | 41 |
| 7 Ethische afwegingen .....                      | 46 |
| 8 Risico analyse .....                           | 47 |
| 9 Communicatie .....                             | 49 |
| 10 Bibliografie .....                            | 50 |
| 11 Appendix .....                                | 51 |
| Appendix B – Microcontroller Specificaties ..... | 57 |
| §B.1 - Arduino .....                             | 57 |
| §B.2 - Espressif .....                           | 58 |
| §B.3 - MediaTek .....                            | 59 |
| §B.4 - Nordic.....                               | 60 |
| §B.5 - NuFront.....                              | 61 |
| §B.6 - RealTek.....                              | 62 |
| §B.7 - Texas Instruments.....                    | 63 |
| Appendix C – ESP-IDF Componenten .....           | 64 |
| Appendix D – Xtensa LX6 Assembler.....           | 68 |
| §D.1 Load instructions .....                     | 68 |
| §D.2 Store instructions .....                    | 68 |

|  |    |
|--|----|
| §D.3 Memory ordering instructions.....     | 68 |
| §D.4 Jump, Call instructions.....          | 68 |
| §D.5 Conditional Branch instructions ..... | 69 |
| §D.6 Move instructions.....                | 69 |
| §D.7 Bitwise logical instructions .....    | 69 |
| §D.8 Arithmetic instructions .....         | 70 |
| §D.9 Shift instructions.....               | 70 |
| §D.10 Processor control instructions.....  | 71 |
| Appendix E - Begrippen.....                | 72 |

## 1. Inleiding

In het vierde jaar van HBO-ICT hebben studenten de keuze om een onderzoeksemester te doen. Hierbij wordt een onderzoek gedaan naar een onderwerp dat door de student gekozen is. Dit verslag zal gaan over een onderzoek dat door René de Kluis uitgevoerd is bij Hogeschool Utrecht.

### §1.1 Organisatorische Context

HBO-ICT bestaat uit de richtingen Software-Netwerk Engineering, Software-Information Engineering, Business-IT & Management en Technische informatica. Deze opleidingen zijn allemaal bachelor niveau. Dit onderzoek is gemaakt door een student van Technische informatica en het verslag is heeft deze studierichting ook als doelgroep.

Het onderzoek is voornamelijk uitgevoerd in het Turing Lab van de Hogeschool Utrecht. Dit is een werkplek met meerdere elektronische componenten en apparaten voor studenten die met technische elementen willen werken. Het bevindt zich op de vierde verdieping van de Heidelberglaan 15 in Utrecht.

#### §1.1.1 Relatie tot andere projecten

Binnen de Hogeschool Utrecht wordt altijd gezocht naar vernieuwing en verrijking van de lesstof. Hierbij hoort ook het integreren van nieuwe microcontrollers in de lesstof. In dit onderzoek zal niet gewerkt worden met libraries en tools van de fabrikant van de microcontroller, maar met libraries en tools die door de Hogeschool Utrecht gemaakt zijn. Het zal hier gaan om de HWLIB en RTOS libraries en de BMPTK make tool. HWLIB, gemaakt door Wouter van Ooijen, is een C++ library waarmee, object georiënteerd, hardware aangestuurd kan worden. De RTOS (Real Time Operating System) library, gemaakt door Wouter van Ooijen en Marten Wensink, wordt gebruikt voor programma's waarbij de gebruiker op aangeduide tijdstippen bepaalde elementen wilt uitvoeren. Deze twee libraries worden in meerdere cursussen van Technische Informatica gebruikt. Het is de bedoeling dat deze libraries hardware abstractie bevatten, wat betekent dat de library voor elke mogelijke microcontroller gebruikt kan worden zonder aanpassingen. De BMPTK make tool, gemaakt door Wouter van Ooijen, is een op GNU make gebaseerde ontwikkel omgeving, waarmee een programma gemaakt kan worden voor microcontrollers. Deze tool zorgt dat de geschreven code op de hardware terecht komt en dat de microcontroller de code ook kan uitvoeren.

Bij de Hogeschool Utrecht streven ze naar herbruikbaarheid van deze libraries en tools. Bij het toevoegen van een nieuwe microcontroller zal hiernaar gekeken moeten worden. Dit onderzoek kan daardoor nieuwe inzichten geven of aanpassingen in de libraries of tools gemaakt dienen te worden, voor volledige herbruikbaarheid.



### §1.1.2 Contact

#### Onderzoeker

|                       |                            |
|-----------------------|----------------------------|
| <i>Naam</i>           | <i>René de Kluis</i>       |
| <i>Student nummer</i> | 1661627                    |
| <i>Mobiel</i>         | +31(0)6 429 405 74         |
| <i>E-Mail</i>         | rene.dekluis@student.hu.nl |

#### Opdrachtgever

|                 |   |
|-----------------|---|
| <i>Naam</i>     | <i>Wouter van Ooijen</i>                      |
| <i>Locatie</i>  | Heidelberglaan 15 – 4.060<br>3512 JE, UTRECHT |
| <i>Telefoon</i> | +31 (0)6 38150444                             |
| <i>E-Mail</i>   | Wouter.vanooijen@hu.nl                        |

## §1.2 Probleemstelling

In dit deze paragraaf zal de aanleiding voor dit onderzoek besproken worden (de kwestie) en een onderzoeksvraag gevormd worden wat de rode draad zal vormen voor dit onderzoek. Aan het eind van dit document zal deze onderzoeksvraag beantwoord worden. Bij de kwestie zal het huidige probleem beschreven worden en de wensen van de opdrachtgever.

### §1.2.1 Kwestie

Bij de afstudeerrichting Technische Informatica van HBO-ICT aan de Hogeschool Utrecht, wordt voor de meeste technische opdrachten de Arduino Due gebruikt. Dit is een programmeer bordje, waarmee onder andere sensoren uitgelezen en aangestuurd kunnen worden. De Arduino Due heeft geen geïntegreerde wifi of bluetooth modules, waardoor deze draadloze aspecten niet voorkomen in het curriculum van Technische Informatica.

Aangezien Internet of Things (IoT) toepassingen steeds meer in opkomst zijn, wil de Hogeschool Utrecht dit ook opnemen in de cursussen. Hiervoor willen zij een microcontroller gebruiken waarbij wifi en/of bluetooth geïntegreerd zit. Aangezien deze aspecten als toevoeging moeten dienen bij de huidige opgaven, zullen de huidige opgaven die voor de Arduino Due gegeven worden, met kleine aanpassingen ook op de nieuwe microcontroller moeten werken. Daarnaast wilt de Hogeschool Utrecht dat studenten op dezelfde manier kunnen programmeren voor verschillende microcontrollers, hiermee zal rekening gehouden moeten worden bij het toevoegen van de nieuwe microcontroller.

### §1.2.2 Vraagstelling

Om Bluetooth en Wifi aspecten toe te voeren in de lesstof van Technische Informatica van de Hogeschool Utrecht, kan de volgende onderzoeksvraag opgesteld worden:

*“Hoe kan een nieuwe microcontroller met geïntegreerde wifi en/of bluetooth in de huidige libraries en tools van de Hogeschool Utrecht opgenomen worden, zodat draadloze communicatie aspecten toegevoegd kunnen worden in de opgaven van Technische Informatica?”*

Aangezien de gehele onderzoeksvraag complex is, is gekozen om deze op te delen in verschillende deelvragen. Deze deelvragen zullen uiteindelijk samen een antwoord kunnen geven op de onderzoeksvraag. Hieronder zullen de deelvragen uitgewerkt worden die uit de onderzoeksvraag gefilterd kunnen worden:

1. Welke microcontroller ondersteunen wifi en/of bluetooth?
2. Met welke microcontrollers, die wifi en/of bluetooth ondersteunen, kunnen de huidige opgaven van Technische Informatica gerealiseerd worden?
3. Welke aanpassingen zijn nodig om de gekozen microcontroller op te nemen in libraries en tools van de Hogeschool Utrecht?
4. Op welke manier kunnen wifi en/of bluetooth aspecten toegevoegd worden in de opgaven van Technische Informatica?
5. Hoe kan een wifi library opgenomen worden in HWLIB?
6. Hoe kan een bluetooth library opgenomen worden in HWLIB?

## 2. Theoretisch kader

In dit hoofdstuk zal de belangrijkste theoretische achtergrond van dit onderzoek besproken worden. Hierbij wordt in de eerste paragraaf duidelijkheid verschaft over de kernbegrippen die in de documenten centraal zullen staan en welke relaties de kernbegrippen met elkaar hebben.

### §2.1 Kernbegrippen

In deze paragraaf zal verduidelijking worden gegeven over begrippen die in dit document centraal zullen staan. Als eerst zal de herbruikbaarheid van libraries en tools van de Hogeschool Utrecht behandeld worden en daarna hardware abstractie waarmee de integratie van de gekozen microcontroller gerealiseerd zal worden.

#### §2.1.1 Herbruikbaarheid van libraries

Bij de herbruikbaarheid van de libraries van de Hogeschool Utrecht zal gekeken op welke manier de gekozen microcontroller geïntegreerd kan worden en welke eventuele problemen dit met zich mee brengt. Voor de studenten moet de aansturing van de gekozen microcontroller op dezelfde manier gebeuren, zoals dit nu kan met de Arduino Due. Daarvoor zal gekeken moeten worden naar modulariteit van de libraries en tools van de Hogeschool Utrecht, zodat deze gebruikt kunnen worden voor de gekozen microcontroller.

#### §2.1.2 Hardware abstractie

Bij hardware abstractie laat de programmeur de software geloven dat alle mogelijke hardware aanwezig is, maar in feite zal de software alleen de functionaliteiten uitvoeren voor de aangesloten hardware. Hierdoor maakt het voor het systeem niet uit als net een ander module of sensor aangesloten wordt op het systeem.

In de huidige libraries van de Hogeschool Utrecht voor de aansturing van hardware is hardware abstractie aanwezig. Het is namelijk mogelijk om bijvoorbeeld de Arduino Uno, Arduino Nano en Arduino Due op dezelfde manier aan te sturen. Aangezien de gekozen microcontroller hierbij toegevoegd dient te worden, zal hierdoor gekeken moeten worden naar de huidige hardware abstractie en of dit herbruikbaar is.

### 3. Keuze Microcontroller

Voor het uitkiezen van een microcontroller voor het onderzoek, zal gekeken moeten worden welke microcontrollers wifi en/ of bluetooth ondersteunen. In dit hoofdstuk zal daarom uitgezocht worden bij welke microcontrollers één of beide aspecten geïntegreerd zit.

Aangezien studenten met de nieuwe microcontroller moeten gaan werken zijn een aantal filteringen gekozen bij het zoeken naar nieuwe microcontrollers met geïntegreerd wifi en/ of bluetooth. Als eerste zullen microcontrollers boven de €50,00 niet worden meegenomen in de analyse, aangezien de aanschaf van de microcontroller te veel is voor de studenten. Daarnaast zullen microcontrollers met zeer slechte of geen verkrijgbaarheid ook niet meegenomen worden.

In de eerste paragraaf zullen de mogelijke microcontrollers beschreven worden waarbij Bluetooth geïntegreerd zit. Daarna zullen de mogelijke microcontrollers beschreven worden waarbij Wifi geïntegreerd zit. Als laatste zal in de conclusie van dit hoofdstuk de keuze beschreven worden die gemaakt is voor de microcontroller dat gebruikt zal worden in dit onderzoek.

#### §3.1 Microcontrollers met Bluetooth

In deze paragraaf zullen microcontrollers behandeld worden waarbij bluetooth geïntegreerd zit. Voor het opstellen van de selectie zijn de volgende zoektermen op Google gebruikt:

- Microcontroller AND “integrated bluetooth”
- Bluetooth microcontroller
- Microcontroller AND bluetooth

In de onderstaande tabel staan de microcontroller die met deze zoektermen gevonden zijn. In appendix B zullen per fabrikant van de gevonden microcontrollers de specificaties uitgewerkt worden over de microcontrollers.

| Naam       | Fabrikant | Processor            | Prijs           | Appendix |
|------------|-----------|----------------------|-----------------|----------|
| Bluno Nano | Arduino   | Atmega328            | €21,00 - €41,00 | §B.1     |
| Esp32      | Espressif | Tensilica Xtensa LX6 | €5,00 - €11,00  | §B.2     |
| nRF52832   | Nordic    | ARM Cortex-M0        | €19,00 - €34,00 | §B.4     |
| nRF51822   | Nordic    | ARM Cortex-M4        | €30,00 - €34,00 | §B.4     |

### §3.2 Microcontroller met Wifi

In deze paragraaf zullen microcontrollers behandeld worden waarbij wifi geïntegreerd zit. Voor het opstellen van de selectie zijn de volgende zoektermen op Google gebruikt:

- Microcontroller AND “integrated wifi”
- Wifi microcontroller
- Microcontroller AND wifi

In de onderstaande tabel staan de microcontroller die met deze zoektermen gevonden zijn. In appendix A zullen per fabrikant van de gevonden microcontrollers de specificaties uitgewerkt worden over de microcontrollers.

| Naam    | Fabrikant         | Processor             | Prijs           | Appendix |
|---------|-------------------|-----------------------|-----------------|----------|
| Esp8266 | Espressif         | Tensilica Xtensa L106 | €2,00 - €12,00  | §B.2     |
| Esp8285 | Espressif         | Tensilica Xtensa L106 | €2,00           | §B.2     |
| Esp32   | Espressif         | Tensilica Xtensa LX6  | €5,00 - €11,00  | §B.2     |
| CC3200  | Texas Instruments | ARM Cortex-M4         | €26,00 - €39,00 | §B.7     |
| NL6621  | NuFront           | ARM Cortex-M3         | €2,70           | §B.5     |
| MT7681  | MediaTek          | Andes N9              | €3,50           | §B.3     |
| RTL8195 | RealTek           | ARM Cortex-M3         | €40,00          | §B.6     |
| RTL8710 | RealTek           | ARM Cortex-M3         | €5,00           | §B.6     |

### §3.3 Conclusie

De vraagstelling van dit onderzoek stelt dat dat wifi en/ of bluetooth aspecten in de opdrachten verwerkt dienen te worden. Voor studenten is de makkelijkste weg hiervoor als beide aspecten geïntegreerd zitten in de nieuwe microcontroller. Uit de gemaakte selectie van microcontrollers blijkt dat alleen de esp32 module van Espressif dit beide ondersteund. Hierdoor zal voor het de verdere stappen van dit onderzoek de deze module gebruikt worden.

## 4. Curriculum Technische Informatica

Voor het implementeren van draadloze aspecten in het Curriculum van Technische Informatica, is het van belang dat gekeken wordt naar de huidige cursussen die gegeven worden. In dit hoofdstuk zullen eerst de opgaven van Technische Informatica behandeld worden, waarbij gewerkt wordt met embedded systemen. Daarna zullen de implementatiemogelijkheden van draadloze aspecten bij deze cursussen besproken worden. Als laatste zal een conclusie gegeven worden hoe de cursussen mogelijk aangepast kunnen worden, zodat draadloze aspecten geïmplementeerd kunnen worden.

### §4.1 Cursussen

In dit onderzoek wordt gekeken dat studenten met draadloze aspecten leren werken. Tijdens het curriculum Technische Informatica wordt bij een aantal cursussen de nadruk gelegd op het programmeren op embedded systemen. Voor relevantie van het onderzoek, zullen daarom alleen de volgende cursussen behandeld worden:

| <b>Jaar 1</b>   | <b>Jaar 2</b>   |
|-----------------|-----------------|
| TICT-V1IDP-15   | TCTI-V2CPSE1-16 |
| TICT-V1OOPC-15  | TCTI-V2THDE-16  |
| TICT-V1IPASS-15 | TCTI-V2MRB-14   |
|                 | TCTI-R2D2-17    |

#### §4.1.3 TICT-V1IDP-15

In het propedeusejaar wordt een interdisciplinaire (IDP) themaopdracht gegeven, waarbij studenten uit elke ICT richting (Software Netwerk Engineering, Software Information Engineering, Technische Informatica en Business IT en Management) samen moeten werken aan één opdracht. Bij deze themaopdracht kunnen de studenten kiezen uit drie verschillende opdrachten:

- Sportschool
  - Hierbij moeten studenten de sportschool automatiseren.
- Maeslandkering
  - Hierbij moeten studenten een werkend schaalmodel maken van de Maeslandkering.
- Domotica huis
  - Hierbij moeten studenten domotica toepassingen ontwikkelen voor in een huis.

De uitwerking van deze opdrachten worden gemaakt in Python op een Raspberry Pi. Hierbij worden de studenten beoordeeld op het ontwerp van de opdracht, de complexiteit van de bedachte uitwerking, samenwerking van het team en de creativiteit in de uitwerking.

#### §4.1.1 TICT-V1OOPC-15

Bij de cursus TICT-V1OOPC-15 maken studenten van Technische Informatica voor het eerst kennis met C++ programmeren. In deze cursus worden C++ taal constructies en patronen aangeleerd, hoe zij hun programmeercode moeten documenteren en UML klasse diagrammen moeten lezen en opstellen. Ook krijgen de studenten voor het eerst tijdens de studie te maken met een Arduino Due waarop zij kleine programma's moeten maken, zoals lampjes laten knipperen en een schuifregister<sup>1</sup> aansturen.

#### §4.1.2 TICT-IPASS-15

Aan het eind van het propedeusejaar wordt de themaopdracht IPASS gegeven. IPASS staat voor Individuele Propedeuse Assignment. Hierbij moet elke student zelf een opdracht bedenken, die hij/zij na goedkeuring van de docent, moet uitvoeren. Bij deze cursus worden de meest uiteenlopende opdrachten uitgewerkt. Van studenten die een spelconsole maken met de aansturing van PlayStation controllers, tot sloten die open gaan wanneer een specifiek deuntje gefloten wordt.

Deze cursus brengt daarnaast een expliciete voorwaarde met zich mee; wanneer deze cursus niet met een voldoende afgerond wordt, krijgt de student een negatief bindend studieadvies en mag hij/zij de rest van de studie niet voortzetten.

#### §4.1.4 TCTI-V2CPSE1-16

In het eerste blok van het tweede jaar wordt voorgebouwd op de C++ kennis die in het eerste jaar opgedaan is. Bij de cursus TCTI-V2CPSE1-16 krijgen de studenten onder andere te maken met het programmeren van een Real-Time Operating System (RTOS), nieuwe C++ taalconstructies en patronen. Daarnaast maken de studenten ook kennis met Assembler programmeren op een Cortex-M0 processor, Makefiles en leren zij hoe het bouwproces van een programma in zijn werk gaat.

#### §4.1.5 TCTI-V2THDE-16

Aan het eind van het eerste blok van het tweede jaar krijgen de studenten de themaopdracht 'Devices'. Hierbij moet een Lazergame systeem gemaakt worden. Hierbij moeten spelers op elkaar kunnen schieten om punten te verzamelen. Aan het eind van het spel geeft de speler aan de gamemaster door hoeveel punten zij die ronde verkregen hebben, waardoor een winnaar bepaald kan worden. Het systeem werkt door middel van infrarood signalen die het geweer uitzend, waardoor een speler geraakt kan worden en gelijk ziet wie hem neergeschoten heeft. Bij het communiceren naar de gamemaster worden infrarood commando's gestuurd die de hoeveelheid punten van de speler bevatten, waardoor de gamemaster een winnaar kan bepalen.

---

<sup>1</sup> Een component waarmee het aantal GPIO pinnen van een microcontroller uitgebreid kan worden. Dit component werkt door middel van Binaire getallen die aangeven welke pin van het component stroom moet doorgeven.



#### §4.1.7 TCTI-V2MRB-14

Bij de cursus Meten, Regelen & Besturen (MRB) leren de studenten meer over hardware componenten, de regeling van deze componenten en de manier waarop deze componenten verschillende aspecten kunnen meten. Bij deze cursus worden de volgende onderdelen behandeld:

- Sensoren
- Elektronische Componenten
- Filteren en Signaalprocessing
- Besturingscomponenten
- Motoren
- Voedingen

Naast de lessen dienen de studenten ook een opdracht te realiseren. Hierbij kunnen de studenten kiezen uit twee opdrachten.

Bij de eerste opdracht wordt een muziekinstrument gemaakt. Hierbij moet een pingpongbal door een ventilatorbuis bewegen. Wanneer de student zijn/ haar hand op de buis houdt, moet de pingpongbal naar deze positie bewegen. Die positie genereert een bepaalde toon, waardoor muziek gemaakt kan worden.

De tweede opdracht is het balanceren van een pingpongbal op een plaat. Deze plaat staat op drie servomotoren die aangestuurd kunnen worden. De student moet hierbij zorgen dat, door middel van het aansturen van de servomotoren, de pingpongbal altijd in het midden van de plaat blijft liggen.

#### §4.1.6 TCTI-R2D2-17

De laatste cursus waarbij embedded systemen gebruikt worden en standaard in het curriculum van Technische informatica zit is Roving Robots en Distributed Devices (R2D2). Bij deze cursus moeten de studenten een winkel beginnen, waarbij losse modules te koop zijn. Deze modules zijn kleine subsystemen, waar de studenten in scrumteams aan werken. Hierdoor zou een klant verschillende modules uit kunnen kiezen en samen tot één systeem kunnen samenvoegen.

Bij deze cursus krijgen de studenten kennis met het scrum<sup>2</sup> werken en de taken die daarbij horen. Hierbij moeten zij sprints in teams uitvoeren met bijbehorende daily standup's<sup>3</sup> en aan het eind van de sprint een evaluatie opstellen hoe de sprint verlopen is. Daarnaast wordt ook beoordeeld op het documenteren van de code en de modellen die erbij gemaakt worden. Het is namelijk bij deze cursus de bedoeling dat aan het eind van een sprint de module waaraan gewerkt is, doorgegeven wordt aan een volgend team dat ermee verder gaat.

---

<sup>2</sup> Scrum is een Agile manier van werken, waarbij multidisciplinaire teams werken aan de realisatie van een product. Hierbij wordt gewerkt in korte sprints die maximaal 4 weken duren.

<sup>3</sup> Een dagelijkse bijeenkomst van de projectleden waarbij de werkzaamheden op elkaar af gestemd worden en een plan gemaakt wordt voor de komende 24 uur. (Agile Scrum Group, z.d.)

## §4.2 Implementatie van draadloze aspecten

In deze paragraaf zullen de implementatiemogelijkheden van draadloze aspecten in het curriculum van Technische Informatica beschreven worden.

Bij de cursus TICT-V1IDP-15 kan een keuze gemaakt worden uit drie verschillende opdrachten, zie §4.1.3. Bij deze opdrachten zijn meerdere mogelijkheden voor de implementatie van draadloze aspecten. Zo kan bij de sportschool-opdracht een toegangspoort ontwikkeld worden, waarbij leden van de sportschool een pas dienen te gebruiken om toegang te krijgen. De informatie van deze pas wordt dan over Wifi naar een computer gestuurd. Daarnaast kan bij de Maeslandkering-opdracht de waterstand over Wifi gecommuniceerd worden naar de centrale server. Bij de domotica-opdracht kunnen draadloze aspecten gebruikt worden voor communicatie tussen apparaten in het huis. Hierdoor maken de studenten kennis met het gebruik van draadloze aspecten voor hedendaagse toepassingen.

Bij de themaopdracht TICT-IPASS-15 bedenkt elke student zijn eigen opdracht, zie §4.1.2. Hierdoor zijn bij deze opdracht meerdere mogelijkheden om draadloze aspecten te gebruiken. Wanneer de student een IoT toepassing wil realiseren, kunnen hierbij draadloze aspecten gebruikt worden.

Bij de themaopdracht TCTI-V2THDE-16 wordt infrarood gebruikt voor de communicatie tussen spelers en naar de gamemaster, zie §4.1.5. Hierbij kunnen draadloze aspecten toegevoegd worden, zodat bijvoorbeeld de communicatie naar de gamemaster over Wifi verloopt i.p.v. infrarood. Ook kan de gamemaster door middel van draadloze aspecten naar alle spelers tegelijk commando's kunnen sturen, wanneer bijvoorbeeld het spel afgelopen is.

Bij de cursus TCTI-R2D2-17 kiezen studenten zelf wat voor module ze gaan ontwikkelen, zie wat voor module ze gaan ontwikkelen, zie §4.1.6. Hierdoor zijn, net zoals bij de cursus TICT-IPASS-15, veel mogelijkheden voor de implementatie van draadloze aspecten. Wanneer studenten bijvoorbeeld een module willen ontwikkelen waarbij apparaten met elkaar communiceren of data door moeten sturen naar een server, kunnen draadloze aspecten gebruikt worden.

## §4.3 Conclusie

Concluderend uit de vorige paragraaf, zijn er meerdere mogelijkheden om draadloze aspecten te implementeren in de cursussen van Technische Informatica. Maar het implementeren van draadloze aspecten in de cursussen zal een aanvulling moeten bieden op de oorspronkelijke leerdoelen van de cursus. Bij de cursussen TICT-IPASS-15 en TCTI-R2D2, moeten de studenten zelf een opdracht bedenken om te realiseren, dit maakt een vaste implementatie van draadloze aspecten in de cursus lastig.

Bij de cursus TICT-V1IDP-15 kunnen draadloze aspecten gebruikt worden door de student, maar in het eerste jaar zullen de meeste studenten hier weinig tot geen ervaring mee hebben. Hierdoor kunnen de draadloze aspecten, in het huidige curriculum van Technische informatica, het efficiënts geïmplementeerd worden bij de themaopdracht Devices (TCTI-V2THDE-16).

## 5. Libraries en tools

In de Organisatorische Context is de Hardware Library (HWLIB), Real-Time Operating System (RTOS) en Bare Metal Programming Tool Kit (BMPTK) make tool besproken, zie §1.1.1. Dit hoofdstuk zal dieper op deze libraries en tools in gaan en zal het integreren van de van de esp32 in deze libraries en tools beschreven worden.

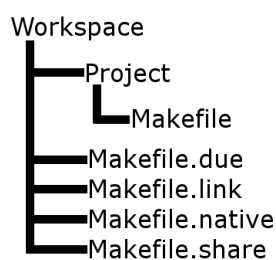
### §5.1 BMPTK

Zoals eerder vermeld is de BMPTK make door gemaakt door Wouter van Ooijen voor de Hogeschool Utrecht. Deze tool is een GNU make gebaseerde ontwikkelomgeving, waarmee een programma gebouwd kan worden. Ook zorgt deze tool dat de geschreven code op de hardware terecht komt waardoor de microcontroller de code kan uitvoeren. BMPTK is gemaakt voor kleine microcontrollers die GCC, C, C++ of assembler gebruiken en kan voor Windows en Linux gebruikt worden. Aangezien deze tool binnen de curricula van Technische Informatica gebruikt wordt, zal de aansturing van de esp32 hierdoor in deze tool geïntegreerd moeten worden.

In dit hoofdstuk zal in de eerste paragraaf de werking van BMPTK besproken en waar de mogelijkheden zitten voor in integreren van een nieuwe microcontroller. Daarna zal in de tweede paragraaf de aansturing van de esp32 aan bod komen.

#### §5.1.1 Werking

BMPTK werkt door middel van 'Makefiles'. Deze bestanden zitten op meerdere plaatsen verspreid om een programma te kunnen bouwen. In de BMPTK tool zelf zit een bestand 'Makefile.inc'. Dit bestand verzorgt het bouwproces van een 'Target'. De term Target staat voor de microcontroller waarvoor het programma gebouwd moet worden, aangezien dit per microcontroller verschilt. De keuze van de Target en het zetten van essentiële waarden gebeurt in de projectfolder en de workspace folder. Deze waarden bevatten bijvoorbeeld de naam van het project, welke bestanden meegenomen moeten worden bij het bouwproces en op welke seriële poort de microcontroller aangesloten zit. Om dit te visualiseren is een figuur gemaakt, zie figuur 6.1.



Figuur 6.1 - Workspace Makefile structuur

#### §5.1.1.1 Makefile.inc

Het bestand Makefile.inc in de bmptk folder is de kern van de BMPTK make tool. Dit bestand verzorgt namelijk het bouwen van een programma voor de verschillende Targets. In dit bestand wordt gekeken voor welke Target een programma gebouwd moet worden en zorgt voor de bijbehorende 'Flags' hiervoor. Deze Flags bestaan onder andere uit commando's die meegegeven dienen te worden aan de compiler of toolchain. Ook staan hierin extra bestanden en hun locaties die nodig zijn om de code te laten werken op de microcontroller. Zodra deze Flags correct gevult zijn, kan de code worden gecompileerd, gebouwd en daarna gedraaid worden op de microcontroller.

#### §5.1.2 ESP-IDF

De Espressif IoT Development Framework (ESP-IDF) is de het raamwerk voor de aansturing van de esp32. De ESP-IDF bevat de benodigde onderdelen om de esp32 op verschillende manier aan te kunnen sturen. Zo zou het programma gebouwd kunnen worden met CMake<sup>4</sup>, GNU Make of Python. Aangezien BMPTK op GNU Make gebaseerd is, zal in dit onderzoek voornamelijk naar dit onderdeel gekeken worden.

De ESP-IDF heeft drie verschillende onderdelen in zich:

- Components
- Make
- Tools

Hieronder zullen deze onderdelen verder toegelicht worden.

##### §5.1.2.1 Components

Alle functionaliteiten die gebruikt kunnen worden voor de esp32 worden componenten genoemd. Deze componenten zijn verschillende C-libraries die gebruikt kunnen worden bij het programmeren op de esp32. De belangrijkste basiscomponenten die gebruikt dienen te worden bij het bouwen van een applicatie voor de esp32 zijn:

- Bootloader
- Bootloader\_support
- Esp32
- Freertos

Bootloader en bootloadersupport zijn twee componenten die samen werken om het programma op de een plaats te geven in het memory van de esp32 en zorgen daarmee ook dat de esp32 op kan starten en de applicatie kan vinden/ uitvoeren.

Het component esp32 bevat de core-functionaliteiten, die benodigd zijn voor het laten werken van het programma. Zo bevat deze de linkerscripts die de verschillende stukken programmeercode in

---

<sup>4</sup> Open-source, cross-platform bundel van tools, waarmee software gebouwd, getest en verpakt kan worden. (CMake, z.d.)

kunnen delen in het memory, maar ook C-bestanden die zorgen voor het starten van de CPU en dat basisfunctionaliteiten gebruikt kunnen worden.

Freertos is een component dat ook standard gebruikt moet worden. Deze wordt gebruikt, aangezien de esp32 een dual core microprocessor is en met Freertos threads aangemaakt kunnen worden. Deze threads zorgen dat beide cores van de esp32 benut kunnen worden voor de programmeur.

Alle andere componenten/ functionaliteiten die gebruikt kunnen worden op de esp32 zullen verder uitgewerkt worden in appendix C.

#### *§5.1.2.2 Make*

Make bevat de Makefiles waarmee de benodigde bestanden verzameld, gecompileerd en gelinkt worden en het gehele programma gebouwd wordt.

Bij het verzamelen van de bestanden wordt gekeken naar de componenten die nodig zijn voor het draaien van de applicatie. Hierbij worden altijd de eerdergenoemde componenten meegenomen. Wanneer bijvoorbeeld het ethernet protocol 'Modbus' gebruikt wordt in een applicatie, zal dit component ook mee gecompileerd moeten worden.

Wanneer in beeld is gebracht welke componenten benodigd zijn voor het bouwen van de applicatie, wordt gekeken waar de source files (.c / .asm / .S/ .cpp/ etc.) en headerfiles (.h / .hpp) staan om deze om te zetten naar objecten.

Als dit allemaal zonder problemen verloopt kunnen de objecten daarna aan elkaar gelinkt worden door middel van de linker scripts die in de ESP-IDF zitten. Deze zorgen dat de gevormde objecten een plek in het geheugen krijgen. Wanneer voor alle objecten een plaats in het geheugen is toegekend kan dit geheel samengevoegd worden tot één applicatie dat op de esp32 geflashed kan worden.

#### *§5.1.2.3 Tools*

Tools bevat een groot aantal van onderdelen dat gebruikt kan worden voor de esp32, maar ook voor onderdelen in de ESP-IDF. Zo staan hier bestanden die het mogelijk maken om een applicatie met CMake te bouwen, default configuraties voor het bouwen van een applicatie aangemaakt kunnen worden, maar ook tools waarmee elementen van de esp32/ ESP-IDF getest kunnen worden of een programma op de esp32 geflashed kan worden.

Voor het flashen van een programma moet de esptool of idf tool aangeroepen worden. Hieraan worden configuraties meegegeven en op welke plaats de applicatie in het geheugen moet komen te staan. Wanneer het programma op de esp32 geflashed is, zal deze daarna direct uitgevoerd worden.

### §5.1.3 Conclusie

Om de aansturing van de esp32 te integreren in de BMPTK make tool zijn een aantal onderdelen van belang. Een nieuwe target zal aangemaakt moeten worden voor de microcontroller, Waarbij de correcte vlaggen gevuld worden. De meest essentiële onderdelen die in de vlaggen van BMPTK opgenomen dienen te worden zijn als volgt:

- De componenten uit de ESP-IDF die gebruikt worden in de applicatie
- De source files (en de locatie hiervan) die omgevormd moeten worden tot objecten
- De Headerfiles (en de locatie hiervan) die bij de source files horen
- Libraries die meegenomen moeten worden
- Linker scripts die uitgevoerd moeten worden
- Opties die bij het compileren, linken of bouwen meegenomen worden

Daarnaast zijn nog andere vlaggen die gezet moeten worden voor essentiële informatie over de microcontroller. Deze vlaggen bestaan onder andere uit:

- De poort waarop de microcontroller aangesloten is
- De Baudrate van de microcontroller
- Het ROM beginadres van de applicatie
- Het RAM beginadres van de applicatie
- Etc.

Na onderzoek blijkt dat informatie voor het invullen van deze vlaggen te verkrijgen is in de ESP-IDF. Wanneer de invulling van de bovengenoemde vlaggen in BMPTK worden opgenomen, kan een applicatie voor de esp32 gebouwd worden. Hierdoor kan geconcludeerd worden, dat integratie met BMPTK mogelijk is.

## §5.2 HWLIB

De Hardware Library (HWLIB), gemaakt is door Wouter van Ooijen, is een C++ library waarmee, object georiënteerd, hardware aangestuurd kan worden. Deze library heeft verschillende functionaliteiten waaronder:

- Digitaal analoog conversie
- Aansturing van OLED scherm
- Grafische elementen voor OLED schermen
- I<sup>2</sup>C aansturing
- SPI aansturing
- Aansturing voor shift-registers
- Aansturing voor matrix keypads
- Aansturing en uitlezen van GPIO pinnen

De library is zeer modulair, waardoor deze functionaliteiten bij een groot aantal microcontrollers gebruikt kan worden. Alleen met de functionaliteiten waarbij GPIO pinnen gebruikt worden, zal gezorgd moeten worden, dat de juiste registers ingesteld worden voor de aansturing. Hiervoor heeft de HWLIB library voor elke microcontroller een eigen header file, waarbij deze benodigde waarden ingesteld worden. In het bestand 'hwlib.hpp' wordt gekeken welke microcontroller aangesloten is, en voor welke het de register waarden moet instellen. De informatie over welke microcontroller aangesloten zit, wordt in de Makefile gespecificeerd onder de waarde "Target", zie §5.1.1.

Wanneer een Wifi en/ of bluetooth library gemaakt wordt, zal deze toegevoegd moeten worden in de library folder in HWLIB. De header file van de Wifi en/ of bluetooth library moet vervolgens toegevoegd worden in de header file "hwlib-all.hpp" en in het bestand "Makefile.inc". Daarna zal deze gebruikt kunnen worden door de student.

### §5.3 RTOS

Een Real-Time Operating System (RTOS) is een besturingssysteem, waarbij een gebruiker op aangeduide tijdstippen bepaalde taken kan uitvoeren. De taken die uitgevoerd kunnen worden hebben ieder een eigen prioriteit, wat het systeem zal volgen bij het uitvoeren van de taken.

De RTOS library van de Hogeschool Utrecht is gemaakt door Wouter van Ooijen en Marten Wensink. Deze library een aantal functionaliteiten die gebruikt kunnen worden door de programmeur. Deze functionaliteiten zullen in de onderstaande tabel uitgewerkt worden, samen met een korte beschrijving.

| Functionaliteit | Beschrijving   |
|-----------------|--|
| Mutex           | Een Mutex is een taak dat aangemaakt kan worden binnen het besturingssysteem. Deze kan “tegelijk” uitgevoerd worden met andere taken.  |
| Eventflag       | Een Eventflag is een vlag die ingesteld kan worden. De code zal delen alleen uitvoeren, wanneer de bijbehorende vlag ingesteld wordt.  |
| Channel         | Een Channel is een lijst met data elementen. Een Channel kan alleen gelezen worden door de taak waarin deze is aangemaakt. Andere taken kunnen wel data elementen aan deze lijst toevoegen. De mogelijkheid om uit de Channel te lezen is geblokkeerd als de lijst van data elementen leeg is. |
| Timer           | Een Timer is een tijdselement dat aangemaakt kan worden. Deze zal een bericht geven als de, door de programmeur ingestelde, tijd verstreken is.  |
| Clock           | Een Clock is, net als de Timer, een tijdselement dat aangemaakt kan worden. Het verschil met de Timer is dat de Clock voor altijd draait en een Timer aangezet moet worden. Een Clock wordt gebruikt wanneer iets uitgevoerd dient te worden na een bepaalde tijd.                             |
| Pool            | Een Pool lijkt in functionaliteit op een Channel, met het verschil dat wanneer een data element in de pool gezet wordt, het lezen van de pool geblokkeerd wordt. Dit geldt ook als een data element gelezen wordt uit de pool.   |

Tabel 5.3.1 – RTOS functionaliteiten



### §5.3.1 FreeRTOS

FreeRTOS is een gratis RTOS die gebruikt kan worden door programmeurs. Zoals in §5.1.2.1 beschreven is, maakt FreeRTOS deel uit van de componenten van de ESP-IDF. Dit component zorgt voor de aansturing van de verschillende cores van de esp32 en voor extra functionaliteiten die diep in de aansturing van het apparaat gebruikt worden. Met het gegeven tijdsbestek voor het onderzoek, is het niet gelukt om een programma te kunnen bouwen zonder FreeRTOS voor de esp32. Als gebruik gemaakt dient te worden van het RTOS dat door de Hogeschool Utrecht ontwikkeld is, zal hiervoor een vervolgonderzoek nodig zijn.

## 6. Conclusie & Aanbeveling

De doelstelling van dit onderzoek was om draadloze aspecten op te nemen binnen het curriculum Technische Informatica van HBO-ICT aan de Hogeschool Utrecht. Aangezien Internet of Things (IoT) toepassingen steeds meer in opkomst zijn, wil de Hogeschool Utrecht dit ook opnemen in de lessen van Technische Informatica.

Om dit te kunnen is in dit onderzoek eerst gekeken bij welke microcontrollers Wifi en/ of Bluetooth geïntegreerd zit en gebruikt kan worden voor studenten. Hieruit is gebleken dat, van de onderzochte microcontrollers, de esp32 van Espressif de beste optie was om te gebruiken. Deze module was gekozen, aangezien het als enige Wifi en Bluetooth geïntegreerd heeft, een lage aanschafprijs heeft voor studenten en makkelijk verkrijgbaar is.

Daarna is gekeken naar de verschillende cursussen van het curriculum Technische Informatica en waar hierbij draadloze aspecten geïmplementeerd konden worden. Hieruit bleek dat de draadloze aspecten het efficiënts toegevoegd konden worden aan de themaopdracht TCTI-V2THDE. Aangezien deze opdracht in het tweede jaar gegeven wordt, hebben studenten meer ervaring met programmeren en zal het werken met draadloze aspecten daardoor toegevoegd kunnen worden. Vervolgens is gekeken naar de huidige libraries en tools van de Hogeschool Utrecht die gebruikt worden voor het programmeren op de Arduino Due. Hierbij is gekeken naar de BMPTK make tool, de HWLIB library en de RTOS library. Het integreren van de aansturing van de esp32 bleek mogelijk bij de BMPTK make tool, wanneer de vlaggen uit de ESP-IDF hierin opgenomen worden. Het integreren van een Wifi en/ of Bluetooth library in HWLIB bleek ook mogelijk te zijn. Hierbij moet gelet worden dat de juiste bestanden toegevoegd worden aan "hwlib-all.hpp" en in het bestand "Makefile.inc" in de HWLIB library. Alleen het integreren van de RTOS library van de Hogeschool Utrecht in de esp32, bracht problemen met zich mee. Het bleek dat de basis aansturing van de esp32 gebruik maakt van FreeRTOS. In het gegeven tijdsbestek was het niet mogelijk om de RTOS library van de Hogeschool Utrecht op te nemen in de aansturing.

De hoofdvraag van dit onderzoek *"Hoe kan een nieuwe microcontroller met geïntegreerde wifi en/of bluetooth in de huidige libraries en tools van de Hogeschool Utrecht opgenomen worden, zodat draadloze communicatie aspecten toegevoegd kunnen worden in de opgaven van Technische Informatica?"* kan door deze informatie beantwoordt worden. Het opnemen van de nieuwe microcontroller kan namelijk gebeuren door de aansturing te implementeren in de BMPTK make tool. Daarnaast moet een Wifi en/ of Bluetooth library in HWLIB toegevoegd worden. Alleen bleek de integratie met de RTOS library niet mogelijk. Hierbij kan wel de keuze gemaakt worden om bij de esp32 FreeRTOS te gebruiken. Hierdoor kunnen nog steeds alle opdrachten van het curriculum Technische Informatica gerealiseerd worden en draadloze communicatie aspecten toegevoegd worden.

## §6.1 Aanbeveling

In deze paragraaf worden aanbevelingen gegeven voor vervolg onderzoeken of veranderingen van onderdelen in het curriculum. De aanbevelingen zijn in vier onderdelen opgedeeld:

- RTOS
- Uniformiteit
- Assembler
- Documentatie

Hieronder zullen deze onderdelen verder beschreven worden.

### §6.1.1 RTOS

Uit de conclusie blijkt dat het lastig is om de RTOS library van de Hogeschool Utrecht te gebruiken bij de aansturing van de esp32. De doelstelling van de cursus waar geleerd wordt met een RTOS te werken, stelt dat de student leert om te gaan met een RTOS. Hierdoor kan geadviseerd worden om bij het implementeren van de esp32 in het curriculum FreeRTOS te gebruiken in plaats van de RTOS library van de Hogeschool Utrecht. De bijbehorende opgaven kunnen van de cursus kunnen dan nog steeds gerealiseerd worden, waardoor dit geen invloed heeft op deze doelstelling van de cursus.

### §6.1.2 Uniformiteit

Voor studenten kan het lastig zijn om met meerdere microcontrollers te werken tijdens zijn/ haar opleiding. Hierdoor kan het verstandig zijn om de esp32 op alle plaatsen in te zetten waar de Arduino Due nu gebruikt wordt. De esp32 heeft namelijk alle functionaliteiten om de meeste opgaven te realiseren die bij de cursussen van Technische Informatica gegeven worden. Hierdoor hoeven de studenten niet met meerdere microcontrollers te werken die misschien verschillend functioneren en kunnen draadloze aspecten toegevoegd worden aan de huidige opgaven. Het enige onderdeel dat veranderd dient te worden zijn de lessen over assembler en de bijbehorende opgaven. Hier zal de volgende paragraaf dieper op in gaan.

### §6.1.3 Assembler

Wanneer gekozen wordt om de esp32 in te zetten bij de cursus TCTI-V2CPSE1-16, zal gekeken moeten worden hoe assembler op de esp32 werkt. Dit verschilt namelijk tussen fabrikanten van processoren. Tijdens de realisatie van het Proof of Concept bleek dat de assembler van de Xtensa LX6 processor op de esp32 nagenoeg overeen komt met de assembler vorm voor de Cortex-M0 processor. Hierbij zijn maar kleine verschillen die klein genoeg zijn, waardoor dit met kleine veranderingen opgenomen kan worden in deze cursus. De assembler die gebruikt kan worden op de Xtensa LX6 zal uitgewerkt worden in appendix D.

### §6.1.4 Documentatie

Tijdens de realisatie van het Proof of Concept, bleek dat de bestanden in de BMPTK make tool slecht gedocumenteerd zijn. Het bestand "Makefile.inc" bevat zeer weinig documentatie over de vlaggen die gezet worden en waarvoor deze gebruikt worden. Dit gaf zeer veel moeilijkheden bij de realisatie van het Proof of Concept, waardoor veel vertraging opgelopen is bij het project. Hierdoor wordt sterk geadviseerd om de Makefiles van de BMPTK make tool duidelijker te documenteren, zodat de integratie van de aansturing van andere microcontrollers makkelijker gerealiseerd kan worden.

## 7. Evaluatie

In dit hoofdstuk zal geëvalueerd worden over het verloop van het onderzoek. Hierbij zullen eerst de knelpunten beschreven worden, waar tegenaan gelopen werd. Daarna zullen de leerpunten van het onderzoek beschreven worden en afgesloten worden met een conclusie.

### §7.1 Knelpunten

Tijdens de realisatie van dit onderzoek moest veel gewerkt worden met Makefiles. De taal die hierbij gebruikt wordt, is in basis uitgelegd in de cursus TCTI-V2CPSE1-16. Alleen bevatte de code die bij de opgaven gemaakt moesten worden maar een aantal regels. Tijdens dit onderzoek moest gewerkt worden met Makefiles waarbij sommige meer dan 1000 regels bevatte. Dit maakte het soms lastig om goed alles door te krijgen. Daarnaast werden in deze Makefiles taalconstructies gebruikt, waar geen les in gegeven is. Dit maakte het zeer lastig om de bestanden te begrijpen en te integreren in de BMPTK make tool.

Daarnaast was het lastig om door te krijgen welke componenten uit de ESP-IDF meegenomen moesten worden tijdens het bouwproces van een applicatie. Dit heeft veel tijd van het onderzoek gekost, voordat de applicatie gebouwd kon worden. De rede waarom het Proof of Concept nog niet werkt is dat naast de applicatie van de student, nog twee interne applicaties gebouwd moeten worden uit de ESP-IDF en meegegeven dienen te worden bij de aansturing van de esp32. Om te programmeren dat deze onderdelen gebouwd konden worden was helaas te weinig tijd, waardoor het Proof of Concept mislukt is.

Als laatste kwamen tijdens de realisatie zeer vreemde fouten voor tijdens het integreren van de aansturing in de BMPTK make tool. Een voorbeeld hiervan is dat de source files correct omgezet werden naar objecten. Maar bij het linken van de objecten, zocht de linker naar objecten, waarbij een random letter verdwenen was. Zo werd bijvoorbeeld de source file “dport\_access.c”, het object “dport\_access.o” gemaakt. Maar bij het linken werd gezocht naar “dport\_aces.o”, waardoor deze niet gevonden werd. Na dagen zoeken en navragen, bleek dat te veel paden naar sourcefiles in het bouw commando stond dat uitgevoerd werd. Wanneer deze namelijk verminderd werden, verdwenen de letter niet meer. Helaas hebben dit soort problemen voor veel vertraging in het onderzoek gezorgd.

### §7.2 Leerpunten

Zoals eerder vermeld, is in dit onderzoek veel gewerkt met Makefiles. De taalconstructies die gebruikt zijn in de ESP-IDF waren lastig te begrijpen aangezien deze niet in de cursussen voor kwamen. Hierdoor is tijdens de realisatie van het onderzoek veel kennis verkregen over Makefiles, wat het maken van projecten in de toekomst makkelijker zal maken.

Daarnaast was maken van de aansturing voor een apparaat ook zeer interessant. Tijdens de cursussen gebruiken studenten namelijk alleen microcontrollers waarbij de aansturing al gerealiseerd is. Hierdoor heb ik veel geleerd over de werking van hardware van microcontrollers en welke stappen bij de realisatie van de aansturing ondernomen moeten worden.

### §7.3 Conclusie

Het is teleurstellend dat het Proof of Concept niet gerealiseerd kon worden binnen het gegeven tijdsbestek. Graag had ik gezien dat aan het eind van dit onderzoek een werkend product opgeleverd kon worden, wat direct gebruikt toepasbaar was in de cursussen van Technische Informatica. Bij het bedenken van het onderzoek heb ik mij verkeken op de moeilijkheidsgraad van dit onderzoek. De kennis die ik had over Makefiles was namelijk niet voldoende om het Proof of Concept binnen het gegeven tijdsbestek te realiseren. Wanneer ik hier meer kennis over had was de realisatie waarschijnlijk wel gelukt, aangezien tijdens de realisatie van het Proof of Concept veel tijd verloren ging aan het doorkrijgen van bepaalde taalconstructies in Makefiles.

Maar zoals eerder in dit hoofdstuk vermeld is, heb ik tijdens de realisatie van dit onderzoek veel geleerd. De kennis die ik heb over de werking van Makefiles is zeer gegroeid, waardoor ik hier makkelijker mee zal kunnen werken in toekomstige projecten. Het project is helaas niet geslaagd, maar ik ben zeer tevreden over de kennis die ik verkregen heb door de realisatie van dit onderzoek.

## 8. Bibliografie

- Agile Scrum Group. (z.d.). *De Daily standup meeting: uitleg en tips*. Opgehaald van Scrum Guide: <https://scrumguide.nl/daily-standup-meeting/>
- Archlinux. (2018, November 26). *WPA supplicant*. Opgehaald van Archlinux: [https://wiki.archlinux.org/index.php/WPA\\_supplicant](https://wiki.archlinux.org/index.php/WPA_supplicant)
- ARM MBED. (z.d.). *ARM MBED*. Opgehaald van ARM MBED: <https://tls.mbed.org/>
- Chan, E. (2018, Oktober 14). *FatFs - Generic FAT Filesystem Module*. Opgehaald van elm-chan: [http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html)
- Cheshire, S., & Krochmal, M. (2013). *Internet Engineering Task Force (IETF)*. Apple Inc.: Februari. Opgehaald van <https://tools.ietf.org/html/rfc6762>
- CMake. (z.d.). *CMake*. Opgehaald van CMake: <https://cmake.org/>
- Cooper, C. (1999, September 1). *Using Expat*. Opgehaald van XML: <https://www.xml.com/pub/a/1999/09/expat/index.html>
- Espressif. (z.d.). *Non-volatile storage library*. Opgehaald van ESP-IDF Programming Guide: [https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/storage/nvs\\_flash.html](https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/storage/nvs_flash.html)
- Espressif. (z.d.). *Partition Tables*. Opgehaald van ESP-IDF Programming Guide: <https://docs.espressif.com/projects/esp-idf/en/latest/api-guides/partition-tables.html>
- Espressif. (z.d.). *Smart Config*. Opgehaald van ESP-IDF Programming Guide: [https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/network/esp\\_smartconfig.html?highlight=smartconfig](https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/network/esp_smartconfig.html?highlight=smartconfig)
- Espressif. (z.d.). *ULP coprocessor programming*. Opgehaald van ESP-IDF Programming Guide: <https://docs.espressif.com/projects/esp-idf/en/latest/api-guides/ulp.html>
- Johnston, P. (2017, Mei 26). *JSMN: JSON parsing library*. Opgehaald van EMBEDDED ARTISTRY: <https://embeddedartistry.com/blog/2017/3/28/jsmn-json-parser>
- Karl, M. J., LofgrenRobert, D., NormanGregory, B., & ThelinAnil, G. (1991). *Wear leveling techniques for flash EEPROM systems*. Washington D.C., Verenigde Staten: Western Digital Corp SanDisk Technologies LLC.
- Libsodium. (2018, Augustus). *Introduction*. Opgehaald van Libsodium documentation: <https://libsodium.gitbook.io/doc/>
- MQTT. (z.d.). *MQTT*. Opgehaald van MQTT: <http://mqtt.org/>
- Vinschen, C., & Johnston, J. (z.d.). *Sourceware*. Opgehaald van Sourceware: <http://www.sourceware.org/newlib/>

## 9. Appendix

---

# Integratie van draadloze aspecten Hogeschool Utrecht

---

Plan van Aanpak



|                       |                      |
|-----------------------|----------------------|
| <b>Auteur</b>         | <b>René de Kluis</b> |
| <b>Student nummer</b> | 1661627              |
| <b>Locatie</b>        | Hogeschool Utrecht   |
| <b>Datum</b>          | 18/10/2018           |
| <b>Versie</b>         | V1.0                 |
| <b>Status</b>         | Eind Document        |



## Inhoudsopgave

|   |    |
|---|----|
| 1 Inleiding.....                            | 34 |
| 2 Organisatorische context.....             | 35 |
| §2.1 Relatie tot andere projecten.....      | 35 |
| §2.2 Contact.....                           | 35 |
| 3 Probleemstelling .....                    | 36 |
| §3.1 Kwestie.....                           | 36 |
| §3.2 Vraagstelling.....                     | 37 |
| 4 Theoretisch kader .....                   | 38 |
| §4.1 Kernbegrippen .....                    | 38 |
| §4.1.1 Herbruikbaarheid van libraries ..... | 38 |
| §4.1.2 Hardware abstractie .....            | 38 |
| §4.3 Kwaliteit .....                        | 39 |
| §4.3.1 ISO25010.....                        | 39 |
| §4.3.2 Richtlijnen Hogeschool Utrecht ..... | 40 |
| 6 Planning & Aanpak.....                    | 41 |
| §6.1 Aanpak .....                           | 41 |
| §6.1.1 Aanpak per deelvraag .....           | 41 |
| §6.2 MoSCoW analyse .....                   | 43 |
| 7 Ethische afwegingen .....                 | 46 |
| §7.1 Druk op student .....                  | 46 |
| §7.2 Extra kosten .....                     | 46 |
| 8 Risico analyse .....                      | 47 |
| §8.1 Externe Risico's .....                 | 47 |
| 9 Communicatie .....                        | 49 |
| 10 Bibliografie .....                       | 50 |
| 11 Appendix .....                           | 51 |
| Appendix A – Deelvraag Analyse .....        | 52 |
| Appendix B – Methoden pad .....             | 55 |
| Appendix Z – Begrippenlijst .....           | 56 |

## 1 Inleiding

In dit verslag zult u de aanpak voor het onderzoek naar de integratie van een nieuwe microcontroller in de bestaande libraries en tools van de Hogeschool Utrecht vinden en toevoeging van draadloze aspecten bij de opdrachten van de studierichting Technische Informatica.

Het onderzoek is onderdeel van het onderzoeksemester, dat in het vierde jaar van HBO-ICT aan de Hogeschool Utrecht gegeven wordt. Hierbij zal onderzoek gedaan worden naar microcontrollers waarbij wifi en/of bluetooth geïntegreerd zit. In dit document zal de probleemstelling voor het onderzoek naar voren komen, de aanpak hoe dit onderzoek uitgevoerd zal worden en de stappen die genomen zullen worden voor de realisatie van een Proof of Concept (PoC).

## 2 Organisatorische context

Het onderzoek zal uitgevoerd worden voor de afstudeerrichting Technische Informatie van de Hogeschool Utrecht. Hierbij zal de onderzoeker voornamelijk in het Turing lab van de Hogeschool Utrecht. Dit is een werkplek met meerdere elektronische componenten en apparaten voor studenten die met technische elementen willen werken. Het bevindt zich op de vierde verdieping van de Heidelberglaan 15 in Utrecht.

### *§2.1 Relatie tot andere projecten*

Binnen de Hogeschool Utrecht wordt altijd gezocht naar vernieuwing en verrijking van de lesstof. Hierbij hoort ook het integreren van nieuwe microcontrollers in de lesstof. In dit onderzoek zal niet gewerkt worden met libraries en tools van de fabrikant van de microcontroller, maar met libraries en tools die door de Hogeschool Utrecht gemaakt zijn. Het zal hier gaan om de HWLIB en RTOS libraries en de BMPTK make tool. HWLIB, gemaakt door Wouter van Ooijen, is een C++ library waarmee je, object georiënteerd, hardware kan aansturen. De RTOS (Real Time Operating System) library, gemaakt door Wouter van Ooijen en Marten Wensink, wordt gebruikt voor programma's waarbij de gebruiker op aangeduide tijdstippen bepaalde elementen wilt uitvoeren. Deze twee libraries worden in meerdere cursussen van Technische Informatica gebruikt. Het is de bedoeling dat deze libraries hardware abstractie bevatten, wat betekent dat de library voor elke mogelijke microcontroller gebruikt kan worden zonder aanpassingen. De BMPTK make tool, gemaakt door Wouter van Ooijen, is een, op make gebaseerde ontwikkel omgeving, waarmee een programma gemaakt kan worden voor microcontrollers. Deze tool zorgt dat de geschreven code op de hardware komt en dat deze de code ook kan uitvoeren.

Bij de Hogeschool Utrecht streven ze naar herbruikbaarheid van deze libraries en tools. Bij het toevoegen van een nieuwe microcontroller zal hiernaar gekeken moeten worden. Dit onderzoek kan daardoor nieuwe inzichten geven of aanpassingen in de libraries of tools gemaakt dienen te worden, voor volledige herbruikbaarheid.

### *§2.2 Contact*

#### Onderzoeker

|                       |                            |
|-----------------------|----------------------------|
| <b>Naam</b>           | René de Kluis              |
| <b>Student nummer</b> | 1661627                    |
| <b>Mobiel</b>         | +31(0)6 429 405 74         |
| <b>E-Mail</b>         | rene.dekluis@student.hu.nl |

#### Opdrachtgever

|                 |   |
|-----------------|---|
| <b>Naam</b>     | Wouter van Ooijen                             |
| <b>Locatie</b>  | Heidelberglaan 15 – 4.060<br>3512 JE, UTRECHT |
| <b>Telefoon</b> | +31 (0)6 38150444                             |
| <b>E-Mail</b>   | Wouter.vanooijen@hu.nl                        |

### 3 Probleemstelling

In dit hoofdstuk zal in de eerste paragraaf de aanleiding voor dit onderzoek besproken worden. Hierbij wordt het probleem beschreven en de wensen van de opdrachtgever. Daarna zullen de hoofd- en deelvragen opgesteld worden uit deze aanleiding. Deze vragen zullen de rode draad vormen voor het onderzoeksverslag.

#### *§3.1 Kwestie*

Bij de afstudeerrichting Technische Informatica van HBO-ICT aan de Hogeschool Utrecht, wordt voor de meeste technische opdrachten de Arduino Due gebruikt. Dit is een programmeer bordje, waarmee onder andere sensoren uitgelezen en aangestuurd kunnen worden. De Arduino Due heeft geen geïntegreerde wifi of bluetooth modules, waardoor deze aspecten niet voorkomen in de opdrachten die voor deze studierichting gegeven worden.

Aangezien Internet of Things (IoT) toepassingen steeds meer in opkomst zijn, wilt de Hogeschool Utrecht dit ook opnemen in de cursussen. Hiervoor willen zij een microcontroller gebruiken waarbij wifi en/of bluetooth geïntegreerd zit. Aangezien deze aspecten als toevoeging moeten dienen bij de huidige opgaven, zullen de huidige opgaven die voor de Arduino Due gegeven worden, ook op de nieuwe microcontroller moeten werken. Daarnaast wilt de Hogeschool Utrecht dat studenten op dezelfde manier kunnen programmeren voor verschillende microcontrollers, hiermee zal rekening gehouden moeten worden bij het toevoegen van de nieuwe microcontroller.

Ook zal in dit onderzoek gekeken worden naar de herbruikbaarheid van de libraries en tools die in §2.1 beschreven zijn. Als Proof of Concept (PoC) zal de integratie met de libraries en tools gerealiseerd worden voor de gekozen microcontroller. Na deze integratie zal gekeken worden of themaopdracht devices (een project wat in het eerste blok van jaar 2 gegeven wordt) gerealiseerd kan worden met de gekozen microcontroller.

### §3.2 Vraagstelling

Om een nieuwe microcontroller te integreren in de huidige libraries en tools van de Hogeschool Utrecht kan de volgende hoofdvraag opgesteld worden:

*“Hoe kan een nieuwe microcontroller met geïntegreerde wifi en/of bluetooth in de huidige libraries en tools van de Hogeschool Utrecht opgenomen worden, zodat draadloze communicatie aspecten toegevoegd kunnen worden in de opgaven van technische informatica?”*

Aangezien de gehele onderzoeksvraag complex is, is gekozen om deze op te delen in verschillende deelvragen. Deze deelvragen zullen uiteindelijk samen een antwoord kunnen geven op de hoofdvraag. Hieronder zullen de deelvragen uitgewerkt worden die uit de hoofdvraag gefilterd kunnen worden.

1. Welke microcontroller ondersteunen wifi en/of bluetooth?
2. Met welke microcontrollers, die wifi en/of bluetooth ondersteunen, kunnen de huidige opgaven van Technische Informatica gerealiseerd worden?
3. Welke aanpassingen zijn nodig om de gekozen microcontroller op te nemen in libraries en tools van de Hogeschool Utrecht?
4. Op welke manier kunnen wifi en/of bluetooth aspecten toegevoegd worden in de opgaven van Technische Informatica?
5. Hoe kan een wifi library opgenomen worden in HWLIB?
6. Hoe kan een bluetooth library opgenomen worden in HWLIB?

## 4 Theoretisch kader

In dit hoofdstuk zal de belangrijkste theoretische achtergrond van dit onderzoek besproken worden. Hierbij wordt in de eerste paragraaf duidelijkheid verschaft over de kenbegrippen die in de documenten centraal zullen staan en welke relaties de kernbegrippen met elkaar hebben. Als laatste zullen ook de kwaliteitseisen besproken worden waar rekening mee gehouden zal worden bij de realisatie van het Proof of Concept (PoC).

### §4.1 Kernbegrippen

In deze paragraaf zal verduidelijking worden gegeven over begrippen die in de documenten centraal zullen staan. Als eerst zal de herbruikbaarheid van de libraries en tools van de Hogeschool Utrecht behandeld worden en daarna hardware abstractie waarmee de integratie van de gekozen microcontroller gerealiseerd zal worden.

#### §4.1.1 Herbruikbaarheid van libraries

Bij de herbruikbaarheid van de libraries van de Hogeschool Utrecht zal gekeken op welke manier de gekozen microcontroller geïntegreerd kan worden en welke eventuele problemen dit met zich mee brengt. Voor de studenten moet de aansturing van de gekozen microcontroller op dezelfde manier gebeuren, zoals dit nu kan met de Arduino Due. Daarvoor zal gekeken moeten worden naar modulariteit van de libraries en tools van de Hogeschool Utrecht, zodat deze gebruikt kunnen worden voor de gekozen microcontroller.

#### §4.1.2 Hardware abstractie

Bij hardware abstractie laat de programmeur de software geloven dat alle mogelijke hardware aanwezig is, maar in feite zal de software alleen de functionaliteiten uitvoeren voor de aangesloten hardware. Hierdoor maakt het voor het systeem niet uit als net een ander module of sensor aangesloten wordt op het systeem.

In de huidige libraries van de Hogeschool Utrecht voor de aansturing van hardware is hardware abstractie aanwezig. Het is namelijk mogelijk om bijvoorbeeld de Arduino Uno, Arduino Nano en Arduino Due op dezelfde manier aan te sturen. In figuur 1 is de deel van de code te zien dat dit mogelijk maakt.

Aangezien de gekozen microcontroller hierbij toegevoegd dient te worden, zal hierdoor gekeken moeten worden naar de huidige hardware abstractie en of dit herbruikbaar is.

```
ifeq ($(TARGET),arduino_uno)
  CHIP      ?= atmega328
  XTAL      ?= 12000
endif

ifeq ($(TARGET),arduino_nano)
  CHIP      ?= atmega328
  XTAL      ?= 12000
endif

ifeq ($(TARGET),arduino_due)
  CHIP      ?= sam3x8e
  XTAL      ?= 12000
endif
```

Figuur 2- Hardware abstractie in BMPTK

#### §4.3 Kwaliteit

Zorgen voor een hoge kwaliteit van software is een hoge prioriteit voor een software ontwikkelaar. Daarom zal deze paragraaf gaan over de verschillende beoordelingscriteria waar rekening mee gehouden dient te worden bij het realiseren van onderzoek en het Proof of Concept (POC).

##### §4.3.1 ISO25010

Bij het ontwikkelen van het POC zullen enkele kwaliteitskenmerken van ISO25010 in acht genomen worden (SYSQA B.V., 2012). Volgend zullen de productkenmerken beschreven worden, die meegenomen worden bij de realisatie.

###### §4.3.1.1 Functionele geschiktheid

Bij functionele geschiktheid wordt gekeken naar de veronderstelde behoeften voor de gebruiker. Aangezien het product voor studenten is, zal hier rekening mee gehouden moeten worden.

###### §4.3.1.2 Prestatie-efficiëntie

Bij het maken van de wifi en bluetooth libraries, is het van belang dat het gebruik geen grote gevolgen heeft voor andere processen. Hierom zal bij de realisatie gelet worden op de snelheid van het product.

###### §4.3.1.3 Bruikbaarheid

Aangezien het product als lesstof bedoeld is, zal ook rekening gehouden moeten worden met de bruikbaarheid van het product. Studenten moeten leren om met de microcontroller om kunnen gaan, hierdoor zal de Leerbaarheid en bedienbaarheid een grote rol spelen bij de realisatie. Daarnaast stijgt het niveau programmeren van de studenten over de jaren, daardoor is het ook van belang om toegankelijkheid en het voorkomen van gebruikersfouten mee te nemen bij de ontwikkeling. Beginnende studenten zullen namelijk meer gebruikersfouten maken dan ervaren studenten; En de ervaren studenten zullen meer uit de microcontroller willen halen dan de beginnende.

###### §4.3.1.4 Onderhoudbaarheid

Onderhoudbaarheid zal ook een rol spelen bij de ontwikkeling, aangezien dit onderzoek door een huidige student gemaakt wordt, die in de komende jaren de studie waarschijnlijk zal verlaten. Als het gemaakte product door anderen onderhouden dient te worden, zal hierom rekening gehouden moeten worden met de modulariteit, herbruikbaarheid en wijzigbaarheid.

###### §4.3.1.5 Overdraagbaarheid

Net als bij onderhoudbaarheid, zal ook gezorgd moeten worden voor goede overdraagbaarheid van het product. Als uitbreidingen of aanpassingen in het product gemaakt dienen te worden, moet dit kunnen als de maker van het product niet meer aanwezig is. Hierdoor zal bij de realisatie gelet moeten worden op de aanpasbaarheid en vervangbaarheid. Ook zal installeerbaarheid meegenomen worden, zodat studenten het gemaakte product makkelijk kunnen installeren of verwijderen.

#### §4.3.2 Richtlijnen Hogeschool Utrecht

Naast de productkenmerken van ISO25010, zullen ook criteria in acht worden gehouden die aangeleerd zijn door de Hogeschool Utrecht. Zo zal bij het ontwikkelen van de software gekeken worden dat weinig codeduplicatie voorkomt. Dit betekent dat elementen in de code niet meermalig voor zullen komen, maar getracht wordt dit samen te voegen. Ook zal de software soft-coded gemaakt worden, zodat de modulariteit van het product groot zal zijn. Als laatste zal gekeken worden naar de kwaliteit van de Doxygen documentatie. Hierdoor zal de code ook leesbaar en begrijpelijk zijn als aanpassingen gemaakt dienen te worden of studenten ermee moeten werken. Om deze criteria te kunnen controleren zal feedback gevraagd worden aan het begeleidend docent of medestudenten.



## 6 Planning & Aanpak

Om het onderzoek tot een succesvol einde te kunnen brengen, is het van belang om een planning op te stellen en de te bedenken welke stappen ondernomen dienen te worden voor de aanpak van het onderzoek. In dit hoofdstuk zullen deze twee onderdelen besproken worden. In de eerste paragraaf zal de aanpak behandeld worden, daarna zal de planning opgesteld worden.

### §6.1 Aanpak

Om een goed onderzoek te kunnen houden moet bepaald worden welke stappen genomen dienen te worden voor de realisatie. Om deze stappen te kunnen bepalen zal in de volgende paragraaf per deelvraag uitgewerkt worden welke stappen ondernomen zullen worden. Daarna zal ook de aanpak besproken worden voor de realisatie van het Proof of Concept.

#### §6.1.1 Aanpak per deelvraag

Volgend zal per deelvraag uitgewerkt worden welke stappen ondernomen dienen te worden om de deelvragen te kunnen beantwoorden.

In appendix A zal per deelvraag een analyse uitgewerkt staan wat voor soort deelvraag het is, op welke manier de dataverzameling zal plaatsvinden, op welke manier het de gevonden data geanalyseerd zal worden en in wat voor vorm het antwoord op de deelvraag verwacht kan worden.

#### **Welke microcontroller ondersteunen wifi en/of bluetooth?**

- Onderzoeken welke microcontroller wifi geïntegreerd hebben
- Onderzoeken welke microcontroller bluetooth geïntegreerd hebben
- Onderzoeken welke microcontroller relevant zijn voor studenten
  - Kosten van de microcontroller
  - Betrouwbaarheid dat de microcontroller nog lang in gebruik zal zijn

#### **Met welke microcontrollers, die wifi en/of bluetooth ondersteunen, kunnen de huidige opgaven van Technische Informatica gerealiseerd worden?**

- Onderzoeken welke functionaliteiten benodigd zijn voor de opdracht.
- Onderzoeken of er genoeg poorten aanwezig zijn op de gekozen microcontroller om de opgave te kunnen realiseren.
- Onderzoeken of de vereiste functionaliteiten op de gekozen microcontroller gemaakt kunnen worden

**Welke aanpassingen zijn nodig om de gekozen microcontroller op te nemen in libraries en tools van de Hogeschool Utrecht?**

Onderzoek naar BMPTK:

- Onderzoeken hoe BMPTK een programma bouwt.
- Onderzoeken hoe een programma voor de gekozen microcontroller gebouwd dient te worden.
- Onderzoeken hoe het bouwen van een programma voor de gekozen microcontroller door BMPTK kan gebeuren.
- Onderzoeken hoe de gekozen microcontroller in BMPTK opgenomen kan worden.

Onderzoek naar HWLIB:

- De functionaliteiten van HWLIB onderzoeken.
- Onderzoeken hoe HWLIB de hardware aanstuurt.
- Aansturing gekozen microcontroller onderzoeken.
- Onderzoeken hoe de gekozen microcontroller in HWLIB opgenomen kan worden.

Onderzoek naar RTOS:

- Functionaliteiten van RTOS onderzoeken.
- Onderzoeken of de gekozen microcontroller deze functionaliteiten kan ondersteunen.
- Onderzoeken hoe de RTOS functionaliteiten de gekozen microcontroller kunnen aansturen.

**Op welke manier kunnen wifi en/of bluetooth aspecten toegevoegd worden in de opgaven van Technische Informatica?**

- Onderzoeken welke functionaliteiten benodigd zijn voor de opdracht
- Analyseren welke functionaliteiten overgenomen zouden kunnen worden m.b.v. wifi of bluetooth
- Onderzoeken of opdrachten gemaakt kunnen worden met wifi, die aansluiten bij de leerdoelen
- Onderzoeken of opdrachten toegevoegd kunnen worden met bluetooth, die aansluiten bij de leerdoelen

**Hoe kan een wifi library opgenomen worden in HWLIB?**

- Aansturing van de verschillende Cores van de gekozen microcontroller onderzoeken.
- Snelheid van het aansturen van WiFi onderzoeken.
- Snelheid van het uitlezen van WiFi onderzoeken.
- Snelheid van het versturen van data over WiFi onderzoeken.
- Ontvangstsnelheid van dataoverdracht over WiFi onderzoeken.

### Hoe kan een bluetooth library opgenomen worden in HWLIB?

- Aansturing van de verschillende Cores van de gekozen microcontroller onderzoeken.
- Snelheid van het aansturen van Bluetooth onderzoeken.
- Snelheid van het uitlezen van Bluetooth onderzoeken.
- Snelheid van het versturen van data over Bluetooth onderzoeken.
- Ontvangstsnelheid van dataoverdracht over Bluetooth onderzoeken

### §6.2 MoSCoW analyse

Om de realisatie van het Proof of Concept (PoC) zo goed mogelijk te laten verlopen. Is het van belang van de eisen gedefinieerd worden. Dit zal in deze paragraaf behandeld worden. De eisen waaraan het PoC moet voldoen zullen aan de hand van een MoSCoW analyse worden uitgewerkt (Linde, 2015). Hierdoor kan een duidelijk beeld gevormd worden, welke taken vereist zijn voor het behalen van een werkend product (Must Have), welke taken eigenlijk vereist zijn, maar het product alsnog geslaagd is zonder (Should Have), welke taken in het product kunnen, maar niet direct noodzakelijk zijn (Could Have) en welke taken niet in het product zullen komen, maar misschien ideeën zijn voor in de toekomst (Won't Have).

#### BMPTK

| Taak  | Prioriteit |
|---|------------|
| Het programma kan gebouwd worden  | Must Have  |
| Het programma kan serieel op de microcontroller geflashed worden              | Must Have  |
| Het BMPTK scherm wordt geopend als het programma op de microcontroller draait | Must Have  |
| Het programma kan draadloos op de microcontroller geflashed worden            | Could Have |
| Real time communicatie kan gebeuren via het BMPTK scherm                      | Could Have |

#### Integratie met Opdrachten

| Taak   | Prioriteit  |
|--|-------------|
| De benodigde poorten kunnen aangestuurd worden                   | Must Have   |
| Het programma kan voldoen aan de originele eisen van de opdracht | Must Have   |
| Assembler opdrachten kunnen gemaakt worden op de microcontroller | Should Have |
| 'Free RTOS' kan samenwerken met HWLIB                            | Could Have  |
| RTOS opdrachten kunnen gerealiseerd worden met 'Free RTOS'       | Could Have  |

**RTOS**

| Taak                              | Prioriteit |
|-----------------------------------|------------|
| Channel kan gemaakt worden        | Must Have  |
| Channel kan aangestuurd worden    | Must Have  |
| Channel kan uitgelezen worden     | Must Have  |
| Clock kan aangemaakt worden       | Must Have  |
| Clock kan aangestuurd worden      | Must Have  |
| Clock kan uitgelezen worden       | Must Have  |
| Event kan aangemaakt worden       | Must Have  |
| Event kan aangestuurd worden      | Must Have  |
| Event kan uitgelezen worden       | Must Have  |
| Flag kan aangemaakt worden        | Must Have  |
| Flag kan aangestuurd worden       | Must Have  |
| Flag kan uitgelezen worden        | Must Have  |
| Mailbox kan aangemaakt worden     | Must Have  |
| Mailbox kan gevult worden         | Must Have  |
| Mailbox kan uitgelezen worden     | Must Have  |
| Mutex kan aangemaakt worden       | Must Have  |
| Mutex kan aangestuurd worden      | Must Have  |
| Pool kan aangemaakt worden        | Must Have  |
| Pool kan aangestuurd worden       | Must Have  |
| Pool kan uitgelezen worden        | Must Have  |
| Task kan aangemaakt worden        | Must Have  |
| Task kan aangestuurd worden       | Must Have  |
| Timer kan aangemaakt worden       | Must Have  |
| Timer kan aangestuurd worden      | Must Have  |
| Timer kan uitgelezen worden       | Must Have  |
| Waitable kan aangemaakt worden    | Must Have  |
| Waitable kan aangestuurd worden   | Must Have  |
| Co-routines kunnen gemaakt worden | Must Have  |
| Co-routines kunnen gestart worden | Must Have  |

|  |            |
|--|------------|
| Wifi kan aangestuurd worden met RTOS                     | Must Have  |
| Bluetooth kan aangestuurd worden                         | Must Have  |
| Meerdere microcontrollers kunnen met elkaar communiceren | Could Have |

**HWLIB**

| Taak   | Prioriteit  |
|--|-------------|
| Pin_in kan aangemaakt worden                   | Must Have   |
| Pin_in kan uitgelezen worden                   | Must Have   |
| Pin_out kan aangemaakt worden                  | Must Have   |
| Pin_out kan aangestuurd worden                 | Must Have   |
| Pin_in_out kan aangemaakt worden               | Must Have   |
| Pin_in_out kan aangestuurd worden              | Must Have   |
| Pin_in_out kan uitgelezen worden               | Must Have   |
| Keypad kan aangemaakt worden                   | Must Have   |
| Keypad kan uitgelezen worden                   | Must Have   |
| Window kan aangemaakt worden                   | Must Have   |
| Een cirkel kan op het window getekend worden   | Must Have   |
| Een vierkant kan op het window getekend worden | Must Have   |
| Oled kan aangemaakt worden                     | Should Have |
| Oled kan aangestuurd worden                    | Should Have |
| ADC kan aangestuurd worden                     | Should Have |
| ADC kan uitgelezen worden                      | Should Have |
| DAC kan aangestuurd worden                     | Should Have |
| DAC kan uitgelezen worden                      | Should Have |
| RS232 verbinding kan gemaakt worden            | Could Have  |
| RS-232 verbinding kan aangestuurd worden       | Could Have  |
| RS-232 verbinding kan uitgelezen worden        | Could Have  |

## 7 Ethische afwegingen

Bij de realisatie van elk onderzoek, zal rekening gehouden moeten worden met Ethische dilemma's. Dit houdt in dat gekeken dient te worden wat voor impact de realisatie van het onderzoek kan hebben voor andere mensen. In dit hoofdstuk zullen deze vraagstukken behandeld worden en op welke manier hiermee omgegaan zal worden.

### *§7.1 Druk op student*

Als een extra microcontroller opgenomen wordt in de lesstof, zal dit zorgen voor meer werkdruk bij de student. Deze zal namelijk ook moeten leren over deze microcontroller en hoe deze gebruikt dient te worden. Echter zal dit niet veel meer zijn dan de student nu moet doen, hierdoor zal dit vraagstuk geen impact hebben op het onderzoek.

### *§7.2 Extra kosten*

Als de gekozen microcontroller in de lesstof opgenomen wordt, betekend dit dat de student hier ook naslagwerk voor dient te hebben. Dit zal waarschijnlijk resulteren in het kopen van extra boeken, wat kosten met zich mee draagt. Echter zullen deze kosten niet dusdanig hoog zijn dat dit waarschijnlijk een probleem zal vormen voor de studenten, waardoor dit vraagstuk geen impact zal hebben op het onderzoek.

Daarnaast wordt momenteel van de student verwacht dat hij/ zij de Arduino Due aanschaft voor de studie. Als de gekozen microcontroller wordt geïntegreerd, zal deze ook door de student aangeschaft moeten worden. Dit brengt extra kosten met zich mee, wat de student moet betalen. Hierdoor zal bij het bepalen van nieuwe de microcontroller, gelet worden op de kosten hiervan. Om het betaalbaar te houden voor studenten is, in overleg met Wouter van Ooijen, gekozen om het onderzoek te doen naar microcontrollers onder de €50,-.

## 8 Risico analyse

In dit hoofdstuk zal de risico analyse behandeld worden. Dit zal gaan over eventuele risico's die het onderzoek zouden kunnen belemmeren. Aangezien dit onderzoek door één persoon gemaakt zal worden, is geen sprake van grote invloeden van interne risico's. Hierdoor zal in dit hoofdstuk alleen de externe risico's behandeld worden.

### §8.1 Externe Risico's

Externe Risico's zijn invloeden van buitenaf die het onderzoek zouden kunnen belemmeren. Voor het slagen van het onderzoek, is het van belang dat rekening gehouden wordt dat deze risico's zich kunnen voordoen. Deze risico's zullen, samen met de bijbehorende maatregelen, in tabel 2 behandeld worden.

| Risico  | Uitleg   | Maatregel   |
|---|--|---|
| Levering van de gekozen microcontroller duur lang                           | Om de aansturing van de gekozen microcontroller te kunnen maken, is het nodig om deze op tijd in bezit te hebben. Anders kan dit voor vertraging zorgen van het onderzoek.   | Als de levering van de gekozen microcontroller te lang duurt, zal gekeken worden of andere mensen deze chip hebben en of deze geleend kan worden tot de bestelde microcontroller binnen is.   |
| BMPTK kan niet samen werken met de toolchain van de gekozen microcontroller | Het kan zijn dat BMPTK niet samen kan werken met de toolchain van de gekozen microcontroller. Dit betekent dat het programma niet door BMPTK gebouwd kan worden en een extern make programma gebruikt dient te worden. | Mocht dit probleem zich voordoen, zal eerst met Wouter van Ooijen overleg plaats vinden om een oplossing hiervoor te vinden. Anders zal een omweg gemaakt worden om het alsnog te realiseren. |

Tabel 2 – Externe risico's

| Risico   | Uitleg   | Maatregel  |
|--|--|--|
| RTOS kan niet gebruikt worden op de gekozen microcontroller  | Als de RTOS van de Hogeschool Utrecht niet op de gekozen microcontroller gebruikt kan worden, kan die gevolgen hebben voor de realisatie van opdrachten.   | Bij het voordoen van dit probleem zal gekeken worden naar het gebruik van een andere RTOS of een omweg gerealiseerd worden.  |
| Functionaliteiten op de gekozen microcontroller kunnen niet door HWLIB aangestuurd worden          | Het kan zijn dat sommige functionaliteiten op de gekozen microcontroller niet aangestuurd kunnen worden door HWLIB. Dit kan gevolgen hebben voor de realisatie van opdrachten.   | Deze functionaliteiten zullen toegevoegd worden aan HWLIB.   |
| Opdrachten van de Hogeschool Utrecht kunnen niet gerealiseerd worden op de gekozen microcontroller | Ook is een risico van dit onderzoek, dat de opdrachten die door de Hogeschool Utrecht gegeven worden, niet op de gekozen microcontroller te realiseren zijn. Dit heeft grote gevolgen voor de realisatie van dit onderzoek, aangezien de integratie van de microcontroller in de lesstof hierbij centraal staat. | Om dit risico te vermeiden zal gekeken worden naar de opdrachten en de functionaliteiten die benodigd worden. Mochten functionaliteiten niet gerealiseerd kunnen worden, zal overleg plaats vinden met de opdrachtgever. |
|  |  | Tabel 2 – Externe risico's (vervolg)   |



## 9 Communicatie

Goede communicatie is essentieel bij het maken van een onderzoek. Tijdens de onderzoeksperiode zijn er verschillende betrokkenen. Als iets verkeerd gaat of anders loopt als verwacht/overeengekomen is, zal dit opgelost moeten worden.

Elke week zal minimaal één overleg plaats vinden van een half uur met het begeleidend docent, hierdoor blijft hij op de hoogte van de huidige status en het verloop van het onderzoek. In het geval dat het onderzoek veranderd dient te worden, zal hierbij eerst overleg plaats vinden met het begeleidend docent. Aangezien het begeleidend docent de huidige libraries van de Hogeschool Utrecht gemaakt heeft, zal met hem waarschijnlijk meer contact zijn dan boven beschreven is, aangezien hij daardoor waarschijnlijk het beste kan helpen als zich knelpunten voordoen tijdens het onderzoek.

## 10 Bibliografie

Cee, Y. (sd). *ESP-WROOM-32 ESP32*. Your Cee. Opgehaald van <https://nl.aliexpress.com/item/ESP-WROOM-32-ESP32-Bluetooth-and-WIFI-Dual-Core-CPU-with-Low-Power-Consumption-MCU/32793415575.html>

Hogeschool Utrecht. (sd). *Tutorials*. Opgehaald van Hogeschool Utrecht:  
<https://www.bibliotheek.hu.nl/~media/HU-BIBLIOTHEEK/Files/beoordelen.pdf?la=nl>

Linde, U. v. (2015, augustus). *Prioriteiten bepalen met MoSCoW*. Opgehaald van Xcess:  
<https://www.xcess.nl/Blog/articleType/ArticleView/articleId/179/Prioriteiten-bepalen-met-MoSCoW>

SYSQA B.V. (2012). *ISO 25010: 2011*. Almere: SYSQA B.V.

## 11 Appendix

---

*Appendix A – Deelvraag Analyse*

In deze bijlage is de aanpak te vinden van de deelvragen. Deze aanpak zal bestaan uit het onderzoeken welk type de deelvraag heeft, welke methode gebruikt zal worden voor dataverzameling, hoe de verkregen data geanalyseerd zal worden en het verwachte resultaat op de deelvraag.

***Welke microcontroller met geïntegreerde wifi en/of bluetooth kan het beste gekozen worden voor studenten?***

|                         |  |
|-------------------------|--|
| Type                    | Vergelijkend<br>Evaluerend   |
| Methode dataverzameling | Literatuur onderzoek   |
| Analyse Methode         | Analyse bruikbaarheid van microcontroller voor studenten.<br>Analyse betrouwbaarheid van de microcontroller. |
| Resultaat               | Document met overzicht: Verschillende bruikbare microcontrollers met geïntegreerde wifi en/of bluetooth      |

***Met welke microcontrollers, die wifi en/of bluetooth ondersteunen, kunnen de huidige opgaven van Technische Informatica gerealiseerd worden?***

|                         |  |
|-------------------------|--|
| Type                    | Beschrijvend   |
| Methode dataverzameling | Literatuur onderzoek<br>Experimenteel onderzoek            |
| Analyse Methode         | Analyse bruikbaarheid microcontrollers voor de opdrachten. |
| Resultaat               | Proof of Concept   |

***Welke aanpassingen zijn nodig om de gekozen microcontroller op te nemen in libraries en tools van de Hogeschool Utrecht?***

|                         |  |
|-------------------------|--|
| Type                    | Ontwerpend<br>Verklarend   |
| Methode dataverzameling | Experimenteel onderzoek<br>Literatuur onderzoek  |
| Analyse Methode         | Analyse bruikbaarheid van de aansturing voor de gekozen microcontroller met de libraries en tools van de Hogeschool Utrecht.             |
| Resultaat               | Proof of Concept<br>Document met uitleg om de gekozen microcontroller aan te sturen met de libraries en tools van de Hogeschool Utrecht. |

***Op welke manier kunnen wifi en/of bluetooth aspecten toegevoegd worden in de opgaven van Technische Informatica?***

|                         |  |
|-------------------------|--|
| Type                    | Beschrijvend<br>Definiërend<br>Verklarend  |
| Methode dataverzameling | Observatie<br>Experimenteel onderzoek<br>Literatuur onderzoek  |
| Analyse Methode         | Analyse bruikbaarheid van de gekozen microcontroller voor de huidige opdrachten van de Hogeschool Utrecht.   |
| Resultaat               | Document hoe wifi en/of bluetooth aspecten toegevoegd kunnen worden aan de opdracht en hoe de opdracht gerealiseerd kan worden met de gekozen microcontroller. |

***Hoe kan een wifi library opgenomen worden in HWLIB?***

|                         |   |
|-------------------------|---|
| Type                    | Desk research<br>Ontwerpend<br>Verklarend   |
| Methode dataverzameling | Experimenteel onderzoek<br>Literatuur onderzoek   |
| Analyse Methode         | Analyse herbruikbaarheid van HWLIB.<br>Analyse bruikbaarheid van wifi library in HWLIB. |
| Resultaat               | Proof of Concept  |

***Hoe kan een bluetooth library opgenomen worden in HWLIB?***

|                         |   |
|-------------------------|---|
| Type                    | Desk research<br>Ontwerpend<br>Verklarend   |
| Methode dataverzameling | Experimenteel onderzoek<br>Literatuur onderzoek   |
| Analyse Methode         | Analyse herbruikbaarheid van HWLIB.<br>Analyse bruikbaarheid van wifi library in HWLIB. |
| Resultaat               | Proof of Concept  |

*Appendix B – Methoden pad*

In deze appendix zal beschreven worden welk pad gevolgd zal worden door de verschillende mogelijke methoden. Dit pad zal helpen bij de realisatie van het onderzoek. Eerst zal het pad gedefinieerd worden, waarna de verklaring volgt waarom deze stappen gekozen zijn en waarom precies in die volgorde.

**Stakeholder Analyse (Field) → Expert interview (library) → Requirements Prioritization (Workshop) → Prototyping (Workshop) → Peer review (Showroom)**

Eerst zal fieldwork gedaan worden door middel van een stakeholder analyse. Als de gekozen microcontroller de Arduino Due moet vervangen, zal met de docenten van Technische Informatica overlegd worden, wat zij van de module willen en wat volgens hen minimaal bij de realisatie opgenomen dient te worden.

Vervolgens zal met Wouter van Ooijen een expert interview gehouden worden, aangezien hij de huidige libraries gemaakt heeft die op de Hogeschool Utrecht gebruikt worden. Hierbij kunnen inzichten naar voren komen die nuttig zijn voor de ontwikkeling van de library voor de aansturing van de gekozen microcontroller.

Daarna zal in de workshop aan bod komen. Hier zal requirements prioritization gehouden worden. Om de library voor de gekozen microcontroller zo snel mogelijk bruikbaar te kunnen maken voor de stakeholder (Hogeschool Utrecht) is het van belang dat alle essentiële onderdelen aanwezig zijn. Daarna kan nog gekeken worden naar uitbreiding van de library, mocht dit nodig zijn.

Ook zal een Proof of Concept (PoC) gemaakt worden, wat de initiele variant van de library zal vormen. Dit valt onder Prototyping uit de methode workshop.

Als laatste zullen peer reviews gehouden worden met Wouter van Ooijen, waarbij hij nuttige inzichten kan geven voor de verdere ontwikkeling van de library en advies geven voor aanpassing van gekozen implementaties.

*Appendix C – Begrippenlijst*

**Arduino Due**

Arduino Due is een microcontroller van het bedrijf Arduino. De microcontroller heeft een Cortex-M3 processor op het board en wordt in de lesstof van Technische Informatica gebruikt voor het aanleren van C++ en assembler programmeren op microcontrollers.

**HWLIB**

HWLIB is een C++ library van de Hogeschool Utrecht. Deze library kan gebruikt worden voor het aansturen van hardware.

**RTOS**

RTOS staat voor Real-Time Operating System. Dit is een besturingssysteem waarbij taken uitgevoerd kunnen worden op tijdstippen dat de gebruiker dit wilt.

**BMPTK**

Bare Metal Programming Tool Kit (BMPTK) is een op make gebaseerde ontwikkelomgeving dat door de Hogeschool Utrecht ontwikkeld is.



## Appendix B – Microcontroller Specificaties

In deze bijlage zijn de specificaties van de mogelijke microcontrollers, waarbij wifi en/ of bluetooth geïntegreerd zit verder uitgewerkt. De microcontrollers staan per fabrikant uitgewerkt.

## §B.1 - Arduino

| Naam                        | Bluno Nano          |
|-----------------------------|---------------------|
| Fabrikant                   | Arduino             |
| Processor                   | Atmega328           |
| Frequentie                  | 20 MHz              |
| ROM                         | 1 KB                |
| RAM                         | 32 KB               |
| Wifi                        | Nee                 |
| Bluetooth                   | Ja<br>Bluetooth 4.0 |
| Prijs                       | €21,00 - €41,00     |
| Verkrijgbaar bij AliExpress | Ja                  |
| Anders verkrijgbaar         | Ja                  |

## §B.2 - Espressif

| Naam                               | Esp32  | Esp8285                  | ESP8266  |
|------------------------------------|--|--------------------------|--|
| <b>Fabriekant</b>                  | Espressif                                    | Espressif                | Espressif  |
| <b>Processor</b>                   | Tensilica Xtensa LX6                         | Tensilica Xtensa L106    | Tensilica Xtensa L106  |
| <b>Frequentie</b>                  | 80 – 240 MHz                                 | 26 – 52 MHz              | 80 MHz   |
| <b>ROM</b>                         | 448 KB                                       | Geen programmeerbare ROM | 64 KB (Niet programmeerbaar)   |
| <b>RAM</b>                         | 520 KB                                       | 50 KB                    | 32 KB instruction RAM<br>32 KB instruction Cache RAM<br>80 KB user-data RAM<br>16 KB ETS system-data RAM |
| <b>Wifi</b>                        | Ja<br><br>(2.4 – 2.5 GHz)<br>IEEE802.11b/g/n | Ja                       | Ja   |
| <b>Bluetooth</b>                   | Ja<br><br>Bluetooth 4.2                      | Nee                      | Nee  |
| <b>Prijs</b>                       | €5,00 - €11,00                               | €2,00                    | €2,00 - €12,00   |
| <b>Verkrijgbaar bij AliExpress</b> | Ja   | Ja                       | Ja   |
| <b>Anders verkrijgbaar</b>         | Ja   | Ja                       | Ja   |

## §B.3 - MediaTek

|                                    |               |
|------------------------------------|---------------|
| <b>Naam</b>                        | <b>MT7681</b> |
| <b>Fabriekant</b>                  | MediaTek      |
| <b>Processor</b>                   | Andes N9      |
| <b>Frequentie</b>                  | 40 MHz        |
| <b>ROM</b>                         | 1 MB          |
| <b>RAM</b>                         | 48 KB         |
| <b>Wifi</b>                        | Ja            |
| <b>Bluetooth</b>                   | Nee           |
| <b>Prijs</b>                       | €3,50         |
| <b>Verkrijgbaar bij AliExpress</b> | Ja            |
| <b>Anders verkrijgbaar</b>         | Ja<br>Mouser  |

## §B.4 - Nordic

| Naam                        | nRF51822        | nRF52832            |
|-----------------------------|-----------------|---------------------|
| Fabriekant                  | Nordic          | Nordic              |
| Processor                   | ARM Cortex-M0   | ARM Cortex-M4       |
| Frequentie                  | 150 - 275 MHz   | 64 MHz              |
| ROM                         | 256 KB          | 512 KB              |
| RAM                         | 32 KB           | 64KB                |
| Wifi                        | Nee             | Nee                 |
| Bluetooth                   | JA              | Ja<br>Bluetooth 4.0 |
| Prijs                       | €30,00 - €34,00 | €19,00 - €34,00     |
| Verkrijgbaar bij AliExpress | Ja              | Ja                  |
| Anders verkrijgbaar         | Ja              | Ja                  |

## §B.5 - NuFront

|                                    |                       |
|------------------------------------|-----------------------|
| <b>Naam</b>                        | <b>NL6621</b>         |
| <b>Fabrikant</b>                   | NuFront               |
| <b>Processor</b>                   | Arm Cortex-M3         |
| <b>Frequentie</b>                  | 40 MHz                |
| <b>ROM</b>                         | 64 KB                 |
| <b>RAM</b>                         | 192 KB                |
| <b>Wifi</b>                        | Ja                    |
| <b>Bluetooth</b>                   | Nee                   |
| <b>Prijs</b>                       | €3,00                 |
| <b>Verkrijgbaar bij AliExpress</b> | Ja                    |
| <b>Anders verkrijgbaar</b>         | Ja, Niet in Nederland |

## §B.6 - RealTek

| Naam                        | RTL8710           | RTL8195       |
|-----------------------------|-------------------|---------------|
| Fabrikant                   | RealTek           | RealTek       |
| Processor                   | ARM Cortex-M3     | ARM Cortex-M3 |
| Frequentie                  |                   | 166MHz        |
| ROM                         | 1 MB              | 1MB           |
| RAM                         | 48 KB             | 1 MB          |
| Wifi                        | Ja<br>802.11b/g/n | Ja            |
| Bluetooth                   | Nee               | Nee           |
| Prijs                       | €5,00             | €40,00        |
| Verkrijgbaar bij AliExpress | Ja                | Ja            |
| Anders verkrijgbaar         | Ja<br>Mouser      | Ja<br>Mouser  |

## §B.7 - Texas Instruments

|                                    |                   |
|------------------------------------|-------------------|
| <b>Naam</b>                        | <b>CC3200</b>     |
| <b>Fabriekant</b>                  | Texas Instruments |
| <b>Processor</b>                   | ARM Cortex-M4     |
| <b>Frequentie</b>                  | 80 MHz            |
| <b>ROM</b>                         | 1 MB              |
| <b>RAM</b>                         | 256 KB            |
| <b>Wifi</b>                        | Ja<br>802.11b/g/n |
| <b>Bluetooth</b>                   | Nee               |
| <b>Prijs</b>                       | €26,00 - €39,00   |
| <b>Verkrijgbaar bij AliExpress</b> | Ja                |
| <b>Anders verkrijgbaar</b>         | Ja                |

## Appendix C – ESP-IDF Componenten

In deze appendix zullen de verschillende componenten/ functionaliteiten uitgewerkt worden die de ESP-IDF bevat. Elk component is een C-library die gebruikt kan worden bij het maken van een programma voor de esp32.

| Component          | Beschrijving   |
|--------------------|--|
| app_trace          | Library dat zorgt dat arbitraire data via een JTAG interface tussen de host en esp32 verstuurd kan worden tijdens het uitvoeren van het programma. |
| app_update         | Library voor het updaten van apps. Deze library wordt o.a. gebruikt door de esp_https_ota library.   |
| asio               | Library voor Audio Stream Input/Output protocol.   |
| aws_iot            | Library gat apparaten die aan AWS services gekoppeld zijn, verbonden kunnen worden met andere apparaten.   |
| bootloader         | Library met basisinformatie die benodigd zijn voor het draaien van een programma op de esp32.  |
| bootloader_support | Library met extra bestanden die benodigd zijn voor de bootloader.  |
| bt                 | Bluetooth library.   |
| coap               | Library voor web-transfer protocol voor IoT applicaties.   |
| console            | Library dat een interactief console verzorgt over de seriële poort.  |
| cxx                | Library voor CXX functionaliteiten.  |
| driver             | Library dat ADC en SPI configuraties verzorgt.   |



|             |   |
|-------------|---|
|             |   |
| esp-tls     | Library voor het checken van authenticiteit van een server.                                       |
| esp32       | Core-elementen voor de aansturing van de esp32.   |
| espcoredump | Library waardoor informatie van de esp32 opgeslagen worden tijdens het crashen van het programma. |
| esptool_py  | Tool waarmee het programma op de esp32 geflashed kan worden.                                      |
| esp_adc_cal | Library waarmee analoge waarden geconvergeerd kunnen worden naar digitale waarden.                |
|             |   |

|                  |  |
|------------------|--|
|                  |  |
| esp_event        | Library waarmee events aangemaakt kunnen worden op de esp32.   |
| esp_https_ota    | HTTPS library.   |
| esp_https_server | HTTPS library.   |
| esp_http_client  | HTTP library.  |
| esp_http_server  | HTTP library.  |
| esp_ringbuf      | Library waarmee ring buffers aangemaakt kunnen worden.   |
| ethernet         | Library voor aansturing van ethernet.  |
| expat            | Library voor het lezen en schrijven van XML bestanden. (Cooper, 1999)  |
| fatfs            | Library dat FAT filesystemen ondersteund. Dit is een filesystem dat ontwikkeld is voor kleine embedd systemen. (Chan, 2018)                          |
| freemodbus       | Library wat het Modbus protocol mogelijk maakt voor de esp32.  |
| freertos         | Library waardoor gebruik gemaakt kan worden van een RTOS.  |
| heap             | Library waardoor gewerkt kan worden met een Heap.  |
| jsmn             | Library voor verwerking van JSON bestanden. Dit is een lichtgewicht library, wat het bruikbaar maakt voor kleine embedded systemen. (Johnston, 2017) |
| json             | Library voor verwerking van JSON bestanden.  |
| libsodium        | Library voor o.a. encryptie, decryptie en wachtwoord hashing. (Libsodium, 2018)  |
| log              | Library waarmee informatie gelogd kan worden en uitgeprint op het terminal.  |
| lwip             | Library voor het maken van een TCP/IP stack.   |
| mbedtls          | Library voor het gebruik van de embed library, waarmee SSL gebruikt kan worden. (ARM MBED, z.d.)   |
| mdns             | Library waarmee IP-adressen toegekent kunnen worden aan andere apparaten in een klein netwerk. (Cheshire & Krochmal, 2013)                           |
| micro-ecc        | Library waarmee de Elliptic-curve Diffie-Hellman en Elliptic Curve Digital Signature Algorithm protocollen gebruikt kunnen worden.                   |
| mqtt             | Library de gebruik van MQTT mogelijk maakt.  |
| newlib           | Library dat een bundel is van verschillende libraries voor embedded systemen. (Vinschen & Johnston, z.d.)  |
| nghttp           | Library om HTTP mogelijk te maken voor embedded systemen.  |

|                     |   |
|---------------------|---|
| nvs_flash           | Library waarmee variabelen een waarde toegekent kunnen krijgen. Deze paren worden opgeslagen in flash. (Espressif, z.d.)  |
| openssl             | Library voor het gebruik van SSL.   |
| partition_table     | Library dat op de esp32 geflashed wordt. Als meerdere applicaties op de esp32 moeten draaien, bevat deze library informatie over de secties waar de applicaties in het geheugen opgeslagen staan. (Espressif, z.d.) |
| protobuf-c          | Library voor protocol buffers.  |
| protocomm           | Library voor protocol communicatie.   |
| pthread             | Library voor het gebruik van pthread.   |
| sdmmc               | Library voor het gebruik van Secure Digital and MultiMediaCard's.   |
| smartconfig_ack     | Library waarmee informatie verkregen kan worden als SSID en wachtwoorden van een verbonden Acces Point. (Espressif, z.d.)   |
| soc                 | Library voor het gebruik van System on a Chip.  |
| spiffs              | Library voor het gebruik van SPIFFS file systemen.  |
| spi_flash           | Library dat zorgt voor het leven, schrijven, verwijderen en mappen van secties in het memory.   |
| tcpip_adapter       | Library voor het gebruik van TCP/IP.  |
| tcp_transport       | Library voor het versturen van informatie over TCP.   |
| ulp                 | Library voor Ultra Low Power coprocessor programmeren. (Espressif, z.d.)  |
| unity               | Library voor het gebruik van unity aspecten.  |
| vfs                 | Library voor het creëren van een Viruteel FileSystem.   |
| wear_levelling      | Library waarmee flash memory opgeslagen kan worden. (Karl, LofgrenRobert, NormanGregory, & ThelinAnil, 1991)  |
| wifi_provisioning   | Library dat aansturing en het gebruik maken van Wifi mogelijk maakt.  |
| wpa_supplicant      | Library dat ethernetprotocollen mogelijk maakt voor point-to-point lan netwerken. (Archlinux, 2018)   |
| xtensa-debug-module | Library dat debugging op Xtensa processoren mogelijk maakt.   |

## Appendix D – Xtensa LX6 Assembler

In deze bijlage zullen de assembler instructies beschreven worden die gebruikt kunnen worden op de Xtensa LX6 processor in de esp32.

Volgens de handleiding “*Xtensa Instruction Set Architecture (ISA)*” (ISA, 2010), kunnen de onderstaande instructies gebruikt worden op de Xtensa LX6 processor.

### §D.1 Load instructions

| INSTRUCTION  | DESCRIPTION   |
|--------------|---|
| <b>L8UI</b>  | 8-bit unsigned load (8-bit offset)                    |
| <b>L16SI</b> | 16-bit signed load (8-bit shifted offset)             |
| <b>L16UI</b> | 16-bit unsigned load (8-bit shifted offset)           |
| <b>L32I</b>  | 32-bit load (8-bit shifted offset)                    |
| <b>L32R</b>  | 32-bit load PC-relative (16-bit negative word offset) |

### §D.2 Store instructions

| INSTRUCTION | DESCRIPTION                         |
|-------------|-------------------------------------|
| <b>S8I</b>  | 8-bit store (8-bit offset)          |
| <b>S16I</b> | 16-bit store (8-bit shifted offset) |
| <b>S32I</b> | 32-bit store (8-bit shifted offset) |

### §D.3 Memory ordering instructions

| INSTRUCTION | DESCRIPTION   |
|-------------|---|
| <b>MEMW</b> | Order memory accesses before with memory access after             |
| <b>EXTW</b> | Order all external effects before with all external effects after |

### §D.4 Jump, Call instructions

| INSTRUCTION   | DESCRIPTION   |
|---------------|---|
| <b>CALLO</b>  | Call subroutine, PC-relative  |
| <b>CALLX0</b> | Call subroutine, address in register  |
| <b>RET</b>    | Unconditional jump, PC-relative   |
| <b>J</b>      | Unconditional jump, address in register   |
| <b>JX</b>     | Subroutine return—jump to return address. Used to return from a routine called by CALLO/CALLX0. |

## §D.5 Conditional Branch instructions

| INSTRUCTION  | DESCRIPTION  |
|--------------|--|
| <b>BALL</b>  | Branch if all of masked bits set                   |
| <b>BNALL</b> | Branch if not all of masked bits set               |
| <b>BANY</b>  | Branch if any of masked bits set                   |
| <b>BNONE</b> | Branch if none of masked bits set (All Clear)      |
| <b>BBC</b>   | Branch if bit clear                                |
| <b>BBCI</b>  | Branch if bit clear immediate                      |
| <b>BBS</b>   | Branch if bit set                                  |
| <b>BBSI</b>  | Branch if bit set immediate                        |
| <b>BEQ</b>   | Branch if equal                                    |
| <b>BEQI</b>  | Branch if equal immediate                          |
| <b>BEQZ</b>  | Branch if equal to zero                            |
| <b>BNE</b>   | Branch if not equal                                |
| <b>BNEI</b>  | Branch if not equal immediate                      |
| <b>BNEZ</b>  | Branch if not equal to zero                        |
| <b>BGE</b>   | Branch if greater than or equal                    |
| <b>BGEI</b>  | Branch if greater than or equal immediate          |
| <b>BGEU</b>  | Branch if greater than or equal unsigned           |
| <b>BGEUI</b> | Branch if greater than or equal unsigned immediate |
| <b>BGEZ</b>  | Branch if greater than or equal to zero            |
| <b>BLT</b>   | Branch if less than                                |
| <b>BLTI</b>  | Branch if less than immediate                      |
| <b>BLTU</b>  | Branch if less than Unsigned                       |
| <b>BLTUI</b> | Branch if less than unsigned immediate             |
| <b>BLTZ</b>  | Branch if less than zero                           |

## §D.6 Move instructions

| INSTRUCTION    | DESCRIPTION                                       |
|----------------|---|
| <b>MOVI</b>    | Load register with 12-bit signed constant         |
| <b>MOVEQZ</b>  | Conditional move if zero                          |
| <b>MOVGEZ</b>  | Conditional move if greater than or equal to zero |
| <b>MOVL TZ</b> | Conditional move if less than zero                |
| <b>MOVNEZ</b>  | Conditional move if non-zero                      |

## §D.7 Bitwise logical instructions

| INSTRUCTION | DESCRIPTION                  | EXAMPLE                                     |
|-------------|------------------------------|---|
| <b>AND</b>  | Bitwise logical AND          | $AR[r] \leftarrow AR[s] \text{ and } AR[t]$ |
| <b>OR</b>   | Bitwise logical OR           | $AR[r] \leftarrow AR[s] \text{ or } AR[t]$  |
| <b>XOR</b>  | Bitwise logical exclusive OR | $AR[r] \leftarrow AR[s] \text{ xor } AR[t]$ |

## §D.8 Arithmetic instructions

| INSTRUCTION  | DESCRIPTION                                  | EXAMPLE   |
|--------------|--|---|
| <b>ADDI</b>  | Add signed constant to register              | $AR[t] \leftarrow AR[s] + (imm87\ 24 \mid imm8)$                                  |
| <b>ADDMI</b> | Add signed constant shifted by 8 to register | $AR[t] \leftarrow AR[s] + (imm87\ 16 \mid imm8 \mid 08)$                          |
| <b>ADD</b>   | Add two registers                            | $AR[r] \leftarrow AR[s] + AR[t]$  |
| <b>ADDX2</b> | Add register to register shifted by 1        | $AR[r] \leftarrow (AR[s]30..0 \mid 0) + AR[t]$                                    |
| <b>ADDX4</b> | Add register to register shifted by 2        | $AR[r] \leftarrow (AR[s]29..0 \mid 02) + AR[t]$                                   |
| <b>ADDX8</b> | Add register to register shifted by 3        | $AR[r] \leftarrow (AR[s]28..0 \mid 03) + AR[t]$                                   |
| <b>SUB</b>   | Subtract two registers                       | $AR[r] \leftarrow AR[s] - AR[t]$  |
| <b>SUBX2</b> | Subtract register from register shifted by 1 | $AR[r] \leftarrow (AR[s]30..0 \mid 0) - AR[t]$                                    |
| <b>SUBX4</b> | Subtract register from register shifted by 2 | $AR[r] \leftarrow (AR[s]29..0 \mid 02) - AR[t]$                                   |
| <b>SUBX8</b> | Subtract register from register shifted by 3 | $AR[r] \leftarrow (AR[s]28..0 \mid 03) - AR[t]$                                   |
| <b>NEG</b>   | Negate                                       | $AR[r] \leftarrow 0 - AR[t]$  |
| <b>ABS</b>   | Absolute value                               | $AR[r] \leftarrow \text{if } AR[s]31 \text{ then } 0 - AR[s] \text{ else } AR[s]$ |

## §D.9 Shift instructions

| INSTRUCTION  | DESCRIPTION   |
|--------------|---|
| <b>EXTUI</b> | Extract unsigned field immediate Shifts right by 0..31 and ANDs with a mask of 1..16 ones The operation of this instruction when the number of mask bits exceeds the number of significant bits remaining after the shift is undefined and reserved for future use. |
| <b>SRLI</b>  | Shift right logical immediate by 0..15 bit positions There is no SRLI for shifts $\geq 16$ ; use EXTUI instead  |
| <b>SRAI</b>  | Shift right arithmetic immediate by 0..31 bit positions   |
| <b>LLI</b>   | Shift left logical immediate by 1..31 bit positions (see page 525 for encoding of the immediate value).   |
| <b>SRC</b>   | Shift right combined (a funnel shift with shift amount from SAR) The two source registers are catenated, shifted, and the least significant 32 bits returned.   |
| <b>SLL</b>   | Shift left logical (Funnel shift $AR[s]$ and 0 by shift amount from SAR)  |
| <b>SRL</b>   | Shift right logical (Funnel shift 0 and $AR[s]$ by shift amount from SAR)   |
| <b>SRA</b>   | Shift right arithmetic (shift amount from SAR)  |
| <b>SSL</b>   | Set shift amount register (SAR) for shift left logical  |
| <b>SSR</b>   | Set shift amount register (SAR) for shift right logical This instruction differs from WSR to SAR in that only the five least significant bits of the register are used.   |
| <b>SSAI</b>  | Set shift amount register (SAR) immediate   |
| <b>SSA8B</b> | Set shift amount register (SAR) for big-endian byte align The t field must be zero.   |
| <b>SSA8L</b> | Set shift amount register (SAR) for little-endian byte align  |

## §D.10 Processor control instructions

| INSTRUCTION  | DESCRIPTION   |
|--------------|---|
| <b>RSR</b>   | Read Special Register   |
| <b>WSR</b>   | Write Special Register  |
| <b>XSR</b>   | Exchange Special Register (combined RSR and WSR) Not present in T1030 and earlier processors  |
| <b>RUR</b>   | RUR reads 32 bits of TIE state into an address register.  |
| <b>WUR</b>   | WUR writes 32 bits to a TIE state register from an address register.  |
| <b>ISYNC</b> | Instruction fetch synchronize: Waits for all previously fetched load, store, cache, and special register write instructions that affect instruction fetch to be performed before fetching the next instruction.                                       |
| <b>RSYNC</b> | Instruction register synchronize: Waits for all previously fetched WSR and XSR instructions to be performed before interpreting the register fields of the next instruction. This operation is also performed as part of ISYNC                        |
| <b>ESYNC</b> | Register value synchronize: Waits for all previously fetched WSR and XSR instructions to be performed before the next instruction uses any register values. This operation is also performed as part of ISYNC and RSYNC.                              |
| <b>DSYNC</b> | Load/store synchronize: Waits for all previously fetched WSR and XSR instructions to be performed before interpreting the virtual address of the next load or store instruction. This operation is also performed as part of ISYNC, RSYNC, and ESYNC. |
| <b>NOP</b>   | No operation  |

## Appendix E - Begrippen

| Begrip  | Definitie  |
|---|--|
| <b>App Trace</b>                                  | Application Level Tracing  |
| <b>ASIO</b>                                       | Audio Stream Input/Output  |
| <b>AWS IoT</b>                                    | Amazon Web Services IoT Platform   |
| <b>COAP</b>                                       | Constrained Application Protocol   |
| <b>Daily Standup</b>                              | Een dagelijkse bijeenkomst van de projectleden waarbij de werkzaamheden op elkaar af gestemd worden en een plan gemaakt wordt voor de komende 24 uur.  |
| <b>Elliptic-curve Diffie-Hellman</b>              | Een encryptie protocol, waarbij beide partijen een publieke en privé elliptische kromme hebben. Met deze elliptische krommen kunnen berichten versleuteld worden.  |
| <b>Elliptic Curve Digital Signature Algorithm</b> | Een algoritme voor de versleuteling van digitale handtekeningen dat gebruik maakt van elliptische krommen.   |
| <b>FATFS</b>                                      | Generic FAT Filesystem Module  |
| <b>LWIP</b>                                       | Lightweight TCP/IP stack   |
| <b>MDNS</b>                                       | Multicast DNS  |
| <b>MQTT</b>                                       | Een communicatie protocol voor kleine sensoren en mobile apparaten. Dit protocol wordt veel gebruikt bij Internet of Things toepassingen. (MQTT, z.d.)   |
| <b>NVS</b>  | Non-Volatile Storage   |
| <b>Schuifregister</b>                             | Een component waarmee het aantal GPIO pinnen van een microcontroller uitgebreid kan worden. Dit component werkt door middel van Binaire getallen die aangeven welke pin van het component stroom moet doorgeven. |
| <b>Scrum</b>                                      | Scrum is een Agile manier van werken, waarbij multidisciplinaire teams werken aan de realisatie van een product. Hierbij wordt gewerkt in korte sprints die maximaal 4 weken duren.                              |
| <b>SDMMC</b>                                      | Secure Digital and MultiMediaCard  |



