Integratie van draadloze aspecten

Hogeschool Utrecht

Onderzoeksverslag

|  |  |
| --- | --- |
| Auteur | René de Kluis |
| Student nummer | *1661627* |
| Datum | *27/12/2018* |
| Locatie | *Heidelberglaan 15, Utrecht* |
| Versie | *v0.0.3* |
| Status | *Concept* |

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.

|  |  |  |
| --- | --- | --- |
| **Versienummer** | **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 | Integratie BMPTK uitgewerkt |
| V0.0.6 | 30/12/2018 | Verandering van de opzet  Invulling HWLIB & RTOS |

Voorwoord

Managementsamenvatting

Inhoudsopgave

[1. Inleiding 7](#_Toc533951900)

[§1.1. Organisatorische Context 7](#_Toc533951901)

[§1.1.1. Relatie tot andere projecten 7](#_Toc533951902)

[§1.1.2. Contact 8](#_Toc533951903)

[§1.2. Probleemstelling 9](#_Toc533951904)

[§1.2.1. Kwestie 9](#_Toc533951905)

[§1.2.2. Vraagstelling 9](#_Toc533951906)

[2. Theoretisch kader 10](#_Toc533951907)

[§2.1. Kernbegrippen 10](#_Toc533951908)

[§2.1.1. Herbruikbaarheid van libraries 10](#_Toc533951909)

[§2.1.2. Hardware abstractie 10](#_Toc533951910)

[3. Keuze Microcontroller 11](#_Toc533951911)

[§3.1. Microcontrollers met Bluetooth 11](#_Toc533951912)

[§3.2. Microcontroller met Wifi 12](#_Toc533951913)

[§3.3. Conclusie 12](#_Toc533951914)

[4. Curriculum Technische Informatica 13](#_Toc533951915)

[§4.1. Opgaven 13](#_Toc533951916)

[§4.2. Implementatie van wifi en bluetooth aspecten 13](#_Toc533951917)

[5. Libraries en tools 14](#_Toc533951918)

[§5.1. BMPTK 14](#_Toc533951919)

[§5.1.1 BMPTK 14](#_Toc533951920)

[§5.1.2 ESP-IDF 15](#_Toc533951921)

[§5.1.3. Conclusie 16](#_Toc533951922)

[§5.2. HWLIB 18](#_Toc533951923)

[§5.2.1. Register aansturing 18](#_Toc533951924)

[§5.2.2. Wifi library 18](#_Toc533951925)

[§5.2.3. Bluetooth Library 18](#_Toc533951926)

[§5.3. RTOS 19](#_Toc533951927)

[§5.3.1. FreeRTOS 19](#_Toc533951928)

[6. Conclusie & Aanbeveling 21](#_Toc533951929)

[7. Bibliografie 22](#_Toc533951930)

[8. Appendix 23](#_Toc533951931)

[Appendix A – Plan van Aanpak 24](#_Toc533951932)

[Appendix B – Microcontroller Specificaties 25](#_Toc533951933)

[§B.1 - Arduino 25](#_Toc533951934)

[§B.2 - Espressif 26](#_Toc533951935)

[§B.3 - MediaTek 27](#_Toc533951936)

[§B.4 - Nordic 28](#_Toc533951937)

[§B.5 - NuFront 29](#_Toc533951938)

[§B.6 - RealTek 30](#_Toc533951939)

[§B.7 - Texas Instruments 31](#_Toc533951940)

[Appendix C – ESP-IDF Componenten 32](#_Toc533951941)

[Begrippen 36](#_Toc533951942)

# 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 bevind 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 betekend 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**

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

**Opdrachtgever**

|  |  |
| --- | --- |
| Naam | Wouter van Ooijen |
| Locatie | Heidelberglaan 15 – 4.060 3512 JE, UTRECHT |
| Telefoon | +31 (0)6 38150444 |
| E-Mail | 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 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, 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?

# 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.

## §2.1. Kernbegrippen

In deze paragraaf zal verduidelijking worden gegeven over begrippen die in dit document centraal zullen staan. Als eest 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. 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.

# 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.

# Curriculum Technische Informatica

## §4.1. Opgaven

## §4.2. Implementatie van wifi en bluetooth aspecten

# Libraries en tools

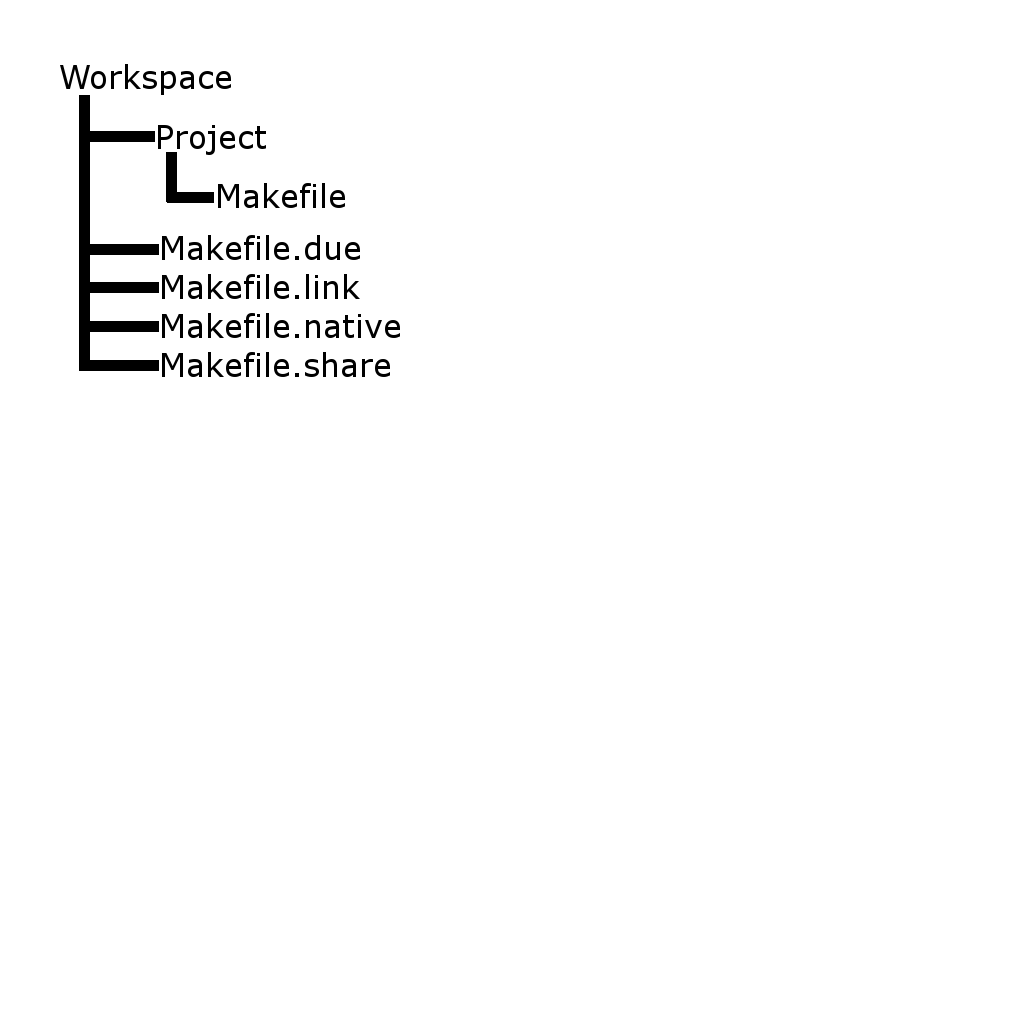
In de Organisatorische Context is de Hardware Library (HWLIB), Real-Time Operating System (RTOS) en Bare Metal Programming Tool Kit (BMPTK) make tool besproken, zie §1.1.1. Dit hoofdstuk zal dieper op deze libraries en tools in gaan en zal het integreren van de van de esp32 in deze libraries en tools 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 bot 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 gebeurd 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[[1]](#footnote-1), 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 bevast de core-functionaliteiten, die benodigd zijn voor het laten werken van het programma. Zo bevat deze de linkerscripts die de verschillende stukken programmeercode in 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
* I2C 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 Operaring 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 geld 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.

# Conclusie & Aanbeveling

# Bibliografie

Archlinux. (2018, November 26). *WPA supplicant*. Opgehaald van Archlinux: 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

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

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

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/

Vinschen, C., & Johnston, J. (z.d.). *Sourceware*. Opgehaald van Sourceware: http://www.sourceware.org/newlib/

# Appendix

## Appendix A – Plan van Aanpak

## 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** |
| **Fabriekant** | Arduino |
| **Processor** | Atmega328 |
| **Frequentie** | 20 MHz |
| **ROM** | 1 KB |
| **RAM** | 32 KB |
| **Wifi** | Nee |
| **Bluetooth** | Ja  Bluetooth 4.0 |
| **Prijs** | €21,00 - €41,00 |
| **Verkrijgbaar bij AliExpress** | Ja |
| **Anders verkrijgbaar** | Ja |

### §B.2 - Espressif

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

### §B.3 - MediaTek

|  |  |
| --- | --- |
| **Naam** | **MT7681** |
| **Fabriekant** | MediaTek |
| **Processor** | Andes N9 |
| **Frequentie** | 40 MHz |
| **ROM** | 1 MB |
| **RAM** | 48 KB |
| **Wifi** | Ja |
| **Bluetooth** | Nee |
| **Prijs** | €3,50 |
| **Verkrijgbaar bij AliExpress** | Ja |
| **Anders verkrijgbaar** | Ja  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  Bluetooth 4.0 |
| **Prijs** | €30,00 - €34,00 | €19,00 - €34,00 |
| **Verkrijgbaar bij AliExpress** | Ja | Ja |
| **Anders verkrijgbaar** | Ja | Ja |

### §B.5 - NuFront

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

### §B.6 - RealTek

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

### §B.7 - Texas Instruments

|  |  |
| --- | --- |
| **Naam** | **CC3200** |
| **Fabriekant** | Texas Instruments |
| **Processor** | ARM Cortex-M4 |
| **Frequentie** | 80 MHz |
| **ROM** | 1 MB |
| **RAM** | 256 KB |
| **Wifi** | Ja  802.11b/g/n |
| **Bluetooth** | Nee |
| **Prijs** | €26,00 - €39,00 |
| **Verkrijgbaar bij AliExpress** | Ja |
| **Anders verkrijgbaar** | 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 filesysteem dat ontwikkeld is voor kleine embedd systemen. (Chan, 2018) |
| freemodbus | Library wat hot 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. encyptie, 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-addressen toegekent kunnen worden aan andere apparaten in een klein netwerk. (Cheshire & Krochmal, 2013) |

|  |  |
| --- | --- |
|  |  |
| micro-ecc | Library waarmee ECDH en ECDSA gebruikt kan 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 variablen 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. |

## Begrippen

|  |  |
| --- | --- |
| **Begrip** | **Definitie** |
| App Trace | Application Level Tracing |
| ECDH |  |
| ECDSA |  |
| MQTT |  |
| ASIO | Audio Stream Input/Output |
| AWS IoT | Amazon Web Services Iot Platform |
| COAP | Contrained Application Protocol |
| FATFS | Generic FAT Filesystem Module |
| LWIP | Lightweight TCP/IP stack |
| MDNS | Multicast DNS |
| NVS | Non-Volatile Storage |
| SDMMC | Secure Digital and MultiMediaCard |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Open-source, cross-platform bundel van tools, waarmee software gebouwd, getest en verpakt kan worden. (CMake, z.d.) [↑](#footnote-ref-1)