***Voordat je gaat werken in dit templatedocument***

*In dit document worden verschillende features van MS Word gebruikt. Deze features worden hier kort toegelicht.*

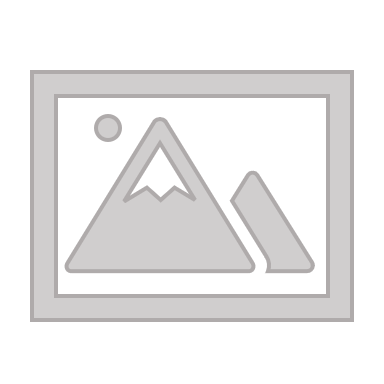
* ***Secties*** *worden gebruikt om documenten op te splitsen en op te maken. In dit document is ieder hoofdstuk een sectie. De overgang van de ene sectie naar de andere is zichtbaar te maken door de optie Alles weergeven ( of CTRL+\*) aan te zetten.*
* *De tekst in de* ***footer*** *(onder aan de pagina’s) kan aangepast worden door erop te dubbelklikken.*
* *Het hoofdstuk* ***Inhoudsopgave*** *wordt automatisch gegenereerd op basis van de hoofdstukindeling. Selecteer de inhoudsopgave en druk op F9 om de inhoudsopgave opnieuw te genereren.*
* *Voor het hoofdstuk* ***Verwijzingen*** *is gebruik gemaakt van de optie om bronnen binnen MS Word te specificeren. Deze bronnen zijn te vinden onder Verwijzingen > Bronnen beheren. Door nieuwe bronnen aan de huidige lijst toe te voegen of te verwijderen, wordt het hoofdstuk verwijzingen automatisch gegenereerd.*
* *Om tekst mooi uit te lijnen is hier en daar gebruik gemaakt van* ***tabellen*** *met een niet zichtbare rand. Om niet zichtbare randen van tabellen zichtbaar te maken moet de optie Tabelindeling > Rasterlijnen (Afbeelding met tekst, Rechthoek, schermopname, ontwerp

  Automatisch gegenereerde beschrijving) weergeven worden geselecteerd.*

**Embedded Systems Engineering**

**Productrapport**

*<Ondertitel van het project>*



***Vervang deze afbeelding met een afbeelding die in één oogopslag het project weergeeft.***

**Embedded Systems Engineering  
Academie Engineering en Automotive  
Hogeschool van Arnhem en Nijmegen**

Auteurs

|  |  |
| --- | --- |
| ***633992*** | ***Maarten van Riel*** |

Datum

|  |
| --- |
| ***februari 2025*** |

Versie

|  |
| --- |
| **0.1** |

# Revisies

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Wanneer** | **Wie** | **Wat** |
| 0.1 | 10 feb | Maarten | Eerste initiële aanzet |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Voorwoord

*Terugblik op de gang van zaken rond het project, hoe heeft de projectgroep het project ervaren, wat heeft de groep ervan geleerd, wat gaat de groep de volgende keer beter doen; kortom het voorwoord kan allerlei persoonlijke bespiegelingen bevatten over het project en het verloop ervan.*

# Samenvatting

*Beschrijving van de uitgangspunten, te bereiken doelen, wat wel en niet bereikt is,* ***bereikte resultaten****; de samenvatting moet een totaalindruk geven van de gehele opdracht en is maximaal 1 A4.*

# Inhoudsopgave

[Revisies 3](#_Toc168479485)

[Voorwoord 4](#_Toc168479486)

[Samenvatting 5](#_Toc168479487)

[Inhoudsopgave 6](#_Toc168479488)

[1 Inleiding 7](#_Toc168479489)

[1.1 Aanleiding 7](#_Toc168479490)

[1.2 Doelstelling 7](#_Toc168479491)

[1.3 Structuur van het rapport 7](#_Toc168479492)

[2 Functioneel ontwerp 8](#_Toc168479493)

[2.1 Functionele specificaties 8](#_Toc168479494)

[2.2 Technische specificaties 9](#_Toc168479495)

[2.3 User interface 10](#_Toc168479496)

[3 Technisch ontwerp 11](#_Toc168479497)

[3.1 Architectuur 11](#_Toc168479498)

[3.2 Interfaces 12](#_Toc168479499)

[3.2.1 Voedingsspanning 12](#_Toc168479500)

[3.2.2 Microcontroller – Sensor 12](#_Toc168479501)

[3.2.3 Microcontroller – Actuator 12](#_Toc168479502)

[3.2.4 Microcontroller – Communicatie – PC driver – App 13](#_Toc168479503)

[3.3 Software 13](#_Toc168479504)

[4 Realisatie 15](#_Toc168479505)

[4.1 Hardware 15](#_Toc168479506)

[4.2 Software 15](#_Toc168479507)

[5 Testen 16](#_Toc168479508)

[6 Conclusies en aanbevelingen 17](#_Toc168479509)

[7 Verwijzingen 18](#_Toc168479510)

[Bijlage A 19](#_Toc168479511)

[Bijlage B 20](#_Toc168479512)

[Bijlage n 21](#_Toc168479513)

# Inleiding

*De inleiding is de toegang tot het rapport. Er wordt beschreven waarom en hoe het project uitgevoerd wordt. De inleiding wordt geschreven aan de hand van de richtlijnen zoals beschreven in het boek Schrijven voor Technici* (Berckel, 2017)*. Dat betekent dat de inleiding hoofdstuk 1 is, geschreven is zonder (tussen)koppen en bestaat uit drie alineagroepen, waarin de aanleiding, de doelstelling en de structuur van het rapport worden beschreven.*

## Aanleiding

*De aanleiding is geschreven vanuit het perspectief van de opdrachtgever. De eerste alinea beschrijft relevante achtergrondinformatie en geeft een informatieve situatieschets van de huidige situatie. De tweede alinea beschrijft het probleem of de wens van de opdrachtgever. De derde alinea beschrijft het belang/relevantie van het project.*

## Doelstelling

*De doelstelling omvat een hoofdvraag (stellend of vragend) en het doel van het project. Tevens wordt beschreven op welke manier de hoofdvraag beantwoord zal worden, oftewel de werkwijze van het project. Tot slot worden de randvoorwaarden en uitgangspunten beschreven.*

## Structuur van het rapport

*Beschrijving van de inhoud van het rapport per hoofdstuk, startend bij hoofdstuk 2. Na het lezen van de structuurbeschrijving is de rode draad van het rapport duidelijk.*

# Functioneel ontwerp

*Het maken van een functioneel ontwerp heeft als doel het verkrijgen van een complete specificatie van het te ontwikkelen systeem in overleg met de klant/opdracht­gever. Je kijkt hierbij met de ogen van de klant. Een goed functioneel ontwerp is onmisbaar bij het ontwikkelen van een product. Het moet immers voldoen aan de eisen en wensen van de klant/opdrachtgever. Als deze niet tevreden is, is al het werk in feite voor niets geweest. Houd bij het schrijven van het functioneel ontwerp continue in gedachten dat er beschreven wordt* ***wat*** *het te ontwikkelen product moet doen. Niet* ***hoe*** *dat moet gebeuren.*

*De uitvoerder van het project vertaalt het pakket van specificaties van de klant/opdrachtgever in een gedetailleerde functionele specificatie. Er wordt onderscheid gemaakt tussen zuiver functionele specificaties en technische specificaties. Het is de kunst om tijdens het opstellen van de specificaties genoeg detail aan te brengen, zodat opdrachtgever en opdrachtnemer elkaar begrijpen, maar ook niet te veel detail, want dat gaat ten koste van de doorlooptijd van het project. Daarbij blijkt het vaak onmogelijk om alle details exact te vangen. Het is dan raadzaam om af te spreken in iteraties te werken, waarbij de specificaties gaandeweg geüpdatet worden.*

## Functionele specificaties

*In de functionele specificatie wordt nauwkeurig vastgelegd welke functies het te realiseren systeem allemaal moet kunnen uitvoeren. De basis hiervoor is gelegd in het hoofdstuk inleiding. Het gedetailleerd vastleggen wordt gedaan in dit hoofdstuk aan de hand van de SMART criteria* (contributors, SMART criteria, 2022)*. Idealiter zou elke* *specificatie derhalve aan de volgende criteria moeten voldoen:*

***Specifiek*** *– Specificaties moeten specifiek zijn en niet generiek. Ze mogen niet openstaan ​​voor verkeerde interpretaties wanneer ze door anderen worden gelezen.*

***Meetbaar*** *– Specificaties moeten kwantificeerbaar zijn om de voltooiing ervan te verifiëren. Voorkom dat een specificatie niet volledig kan worden geverifieerd door het gebruik van niet-kwantitatieve termen (beste, optimaal, snelste) te vermijden.*

***Haalbaar*** *– Zorg ervoor dat de specificatie realistisch kan worden bereikt, gegeven de bestaande omstandigheden en beschikbare middelen.*

***Relevant*** *– Specificaties moeten relevant zijn voor een project. Hoewel alle specificaties belangrijk zijn, krijgen ze een prioriteit om in een vroeg stadium aan te geven welke specificaties voor het bedrijf de meeste toegevoegde waarde bieden. Prioriteiten worden toegekend volgens de MoSCoW methode*(contributors, MoSCoW method, 2022)*. Engineers zullen in eerste instantie proberen om alle Must-have, Should-have en Could-have-specificaties te realiseren, maar de Could-have en Should-have specificaties zullen als eerste worden los gelaten als de doorlooptijd van het project in gevaar komt. De Won’t-have specificaties geven de mogelijkheid om het project af te bakenen.*

***Tijdgebonden*** *– Elke specificatie moet tijdgebonden zijn of specificeren wanneer of hoe snel een eis moet worden ingevuld of uitgevoerd.*

*Gebruik een tabel om de functionele specificaties overzichtelijk en gegroepeerd weer te geven. Er worden maximaal drie niveaus aangebracht om een functionele specificatie steeds verder uit te diepen en daarmee SMARTer te maken. Geef iedere specificatie ook een identificatienummer, zodat er in de rest van het rapport makkelijk naar verwezen kan worden.*

Dit document beschrijft hoe de Datalogger wordt ontworpen en welke functionele en technische eisen eraan gesteld worden. Het ontwerp heeft als doel om tijdwaarnemingen met GPS-coördinaten te registreren en de verzamelde gegevens overzichtelijk weer te geven. De datalogger zal real-time informatie tonen en opslaan, met een gebruiksvriendelijke interface voor de bediening.

### 2.1 Functionele eisen

Hieronder staan de functionele eisen geformuleerd volgens het MoSCoW-principe.

| **#** | **Prioriteit (MoSCoW)** | **Specificatie** | **Details** | **Opmerkingen** |
| --- | --- | --- | --- | --- |
| F1.1 | Must (M) | Tijdregistratie | De hoofdfunctie van de datalogger is het registreren van de tijd van gemeten gebeurtenissen. | Nauwkeurigheid is cruciaal. |
| F1.2 | Must (M) | Gebruikersinterface (OLED-scherm) | De userinformation van de datalogger wordt weergegeven op een monochroom OLED-scherm. | Duidelijke en efficiënte weergave. |
| F1.2 | Must (M) | Meetfrequentie | De datalogger moet een meetfrequentie van 10.000 Hz ondersteunen en de gegevens binnen 5 ms na detectie verwerken en opslaan. | Zorgt voor een nauwkeurige tijdregistratie. |
| F1.5 | Must (M) | Opslagcapaciteit | De datalogger moet minimaal 1000 metingen kunnen opslaan in intern geheugen. | Voldoende opslagruimte voor veldmetingen. |
| F1.6 | Must (M) | Detectie | De datalogger moet signalen van een infraroodafstandsdetector kunnen detecteren en verwerken. De detectie moet, Objecten tot een bereik van 200 cm kunnen waarnemen. Binnen 10 ms na een detectie een meetwaarde registreren en opslaan. Een tijdstempel genereren wanneer een object wordt waargenomen en deze opslaan in het geheugen. Een visuele indicatie tonen op het OLED-scherm zodra een object wordt gedetecteerd. Compatibel zijn met standaard IR-sensoren met een analoge of digitale output. | Moet een reflector kunnen detecteren die detectie van een detectielus signaleert. |
| F1.6 | Must (M) | Status opgeslagen metingen | Het scherm toont het aantal opgeslagen metingen (bijvoorbeeld "Samples: 5"). | Helpt de gebruiker bij het beheren van metingen. |
| F1.7 | Must (M) | Opslagbehoud | De opgeslagen data moet behouden blijven, zelfs bij stroomuitval. | Belangrijk voor data-integriteit. |
| F1.8 | Must (M) | Gebruikersinterface (OLED-scherm) | De meetwaarden worden weergegeven op een monochroom OLED-scherm. | Duidelijke en efficiënte weergave. |
| F1.9 | Must (M) | Live tijdsweergave | Rechtsonder wordt de actuele tijdwaarneming weergegeven in milliseconden (HH:MM:SS.mmmm) en moet binnen 10 ms geactualiseerd worden. | Real-time visualisatie van metingen. |
| F1.10 | Must (M) | Reflectiepaaltjes registratie | In het midden van het display worden de nummers 1-8 weergegeven, met een vinkje achter de nummers waarvan de tijd is opgeslagen. | Helpt bij het monitoren van voltooide metingen. |
| F1.11 | Must (M) | Bediening via knoppen | Onderin het scherm worden knoppen weergegeven: STOP, ERASE, NEXT, START. | Fysieke knoppen zorgen voor snelle bediening. |
| F1.12 | Must (M) | Datalog exporteren | De meetgegevens worden opgeslagen op een SD-kaart. | Handig voor verdere analyse. |
| F1.3 | Should (S) | GPS-ondersteuning | De datalogger moet real-time GPS-coördinaten opslaan bij elke meting. | Belangrijk voor locatiegebonden data. |
| F1.4 | Should (S) | Aantal Satellieten | Rechtsboven op het display moet het aantal verbonden satellieten worden weergegeven. | Biedt inzicht in signaalsterkte. |
| F1.13 | Could (C) | Energiebeheer | De datalogger moet een energiezuinige modus hebben die automatisch wordt geactiveerd na X minuten inactiviteit. | Verlengt de batterijduur. |
| F1.14 | Could (C) | Draadloze communicatie | Optioneel kan Bluetooth of WiFi worden toegevoegd voor draadloze data-export. | Handig voor realtime monitoring zonder kabels. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### 2.2 Technische eisen

Hieronder volgen de technische eisen die nodig zijn voor de implementatie van de datalogger.

| **#** | **Prioriteit (MoSCoW)** | **Omschrijving** |
| --- | --- | --- |
| T1 | Must (M) | De datalogger moet worden gebaseerd op de FRDM-KL25Z microcontroller. |
| T2 | Must (M) | De hardware wordt ontworpen als een uitbreidingsmodule (shield) voor de FRDM-KL25Z. |
| T3 | Must (M) | De firmware wordt geschreven in C (C11-standaard). |
| T4 | Must (M) | Het systeem moet via UART of USB data kunnen exporteren. |
| T5 | Must (M) | De datalogger moet worden gevoed door een 5V USB-voeding of batterijpack. |
| T6 | Must (M) | De microcontroller moet een RTC (Real-Time Clock) ondersteunen om tijdregistraties te synchroniseren. |
| T7 | Should (S) | Het OLED-scherm moet een minimale resolutie hebben van 128x64 pixels. |
| T8 | Should (S) | De datalogger moet GPS-gegevens via een NMEA 0183-compatibele module kunnen ontvangen. |
| T9 | Could (C) | Het systeem kan worden uitgebreid met een SD-kaartmodule voor grotere opslagcapaciteit. |

## User interface

*Geef een schets van het uiterlijk van het te ontwikkelen systeem. Hoe komt het systeem er voor de gebruiker uit te zien? Welke inputs zijn er en welke outputs? Wat verandert er aan de outputs als een bepaalde input verandert? Een schets van de user interface is een andere weergave van hetzelfde systeem zoals beschreven in de voorgaande paragrafen. Het is een communicatiemiddel tussen opdrachtgever en -nemer om de functionaliteit van het te realiseren systeem helder te krijgen.*

# 

# Technisch ontwerp

*In deze fase wordt nagedacht over de wijze van realisatie van de diverse functies. Het gaat hierbij om het* ***hoe*** *van het te ontwikkelen product. Het is niet de bedoeling dat er nieuwe specificaties worden beschreven. Dat is het wat en staat beschreven in het voorgaande hoofdstuk. Het technisch ontwerp heeft als doel het functioneel ontwerp te vertalen in een technische implementatie. Je kijkt nu met de ogen van de ontwerper.*

## Architectuur

*Het systeem wordt eerst onderverdeeld in deelsystemen. Elk deelsysteem heeft een sterke interne samenhang en relatief weinig interactie met de overige deelsystemen. De samenhang tussen de deelsystemen wordt weergegeven in een architectuurschema. Er wordt een onderbouwde keuze gemaakt voor de interface tussen de deelsystemen op basis van de functionele en/of technische specificaties.*

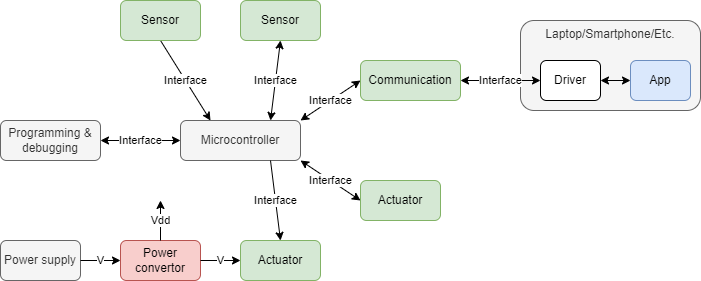
*In Figuur 5 wordt een architectuur getoond die algemeen toepasbaar is voor embedded systemen. Het hart is de microcontroller die communiceert via verschillende interfaces met de deelsystemen. Deze interfaces moeten ondubbelzinnig worden vastgelegd in deze fase van het project.*

*Sensoren hebben een pijl richting de microcontroller, bijvoorbeeld bij een analoge meting, maar kunnen ook een dubbele pijl hebben, bijvoorbeeld bij een seriële bus zoals I2C. Dit laatste geldt ook voor actuatoren, maar actuatoren kunnen ook gerealiseerd worden met een enkele pijl richting de actuator. Denk bijvoorbeeld aan een PWM signaal.*

*Een veel toegepast subsysteem is communicatie met een ander apparaat, zoals een laptop, smartphone, etc. Die interface communiceert doorgaans ook in twee richtingen, maar dat hoeft niet.*

*Het is gebruikelijk om er rekening mee te houden dat de microcontroller van software updates moet kunnen worden voorzien. Daarom is er vaak sprake van een programmeer en/of debugging subsysteem. Er zijn verschillende interfaces waarmee dat mogelijk is, zoals SWD en JTAG.*

*Tot slot wordt er getoond hoe de voedingshuishouding wordt geregeld. Vanaf een spanningsbron wordt een spanningsomvormer gebruikt om de voedingspanning (Vdd) voor het embedded systeem te realiseren. Om het schema overzichtelijk te houden wordt Vdd niet naar alle subsystemen getekend. Het kan nodig zijn om meerdere spanningsniveaus in het embedded systeem beschikbaar te hebben, bijvoorbeeld voor het aansturen van motoren.*



Figuur 5. Algemeen architectuurschema voor een embedded systeem.

## Interfaces

*Voor iedere interface wordt beschreven wat de elektrische- en/of de datacommunicatie-eigenschappen zijn. Soms worden deze keuzes gedicteerd door de (technische) eisen, maar vaak heb je hier als ontwerper ook keuzes te maken. Tevens wordt er voor iedere interface tussen de microcontroller en overige modules middels een UML sequencediagram een ontwerp voor de softwaredriver gemaakt.*

### Voedingsspanning

*De voedingsspanning specificeert welke spanningen er in het systeem nodig zijn, welke spanningsbronnen er zijn, hoe die omgezet worden en welke maximale stroom er verwacht kan worden. Een specificatie wordt duidelijk herkenbaar geformuleerd:*

|  |  |
| --- | --- |
| **Specificatie** | *Hier komt de tekst van de specificatie.* |

### Microcontroller – Sensor

*Voor sensoren geldt dat er beschreven wordt wat de sensor meet. Hierbij wordt waar mogelijk de keuze gekoppeld aan een specificatie. Het is belangrijk om zo volledig mogelijk te zijn, waarbij gedacht moet worden grootheden, eenheden, bereik, precisie, sample frequentie, etc. Een specificatie wordt duidelijk herkenbaar geformuleerd:*

|  |  |
| --- | --- |
| **Specificatie** | *Hier komt de tekst van de specificatie.* |

*Er wordt tevens beschreven dat er een software driver wordt gerealiseerd. Een driver voor een sensor kent in ieder geval een functie om de driver te initialiseren en één of meerdere functies om waarden van de sensor te lezen. Optioneel kan er een functie voor het schrijven naar de sensor worden beschreven, bijvoorbeeld configuratieparameters in te stellen. Kies als prefix voor de namen van de functies de namen die ook in het architectuurschema zijn gebruikt, of een afkorting daarvan.*

|  |  |
| --- | --- |
| **Specificatie** |  |

### Microcontroller – Actuator

*Voor actuatoren geldt dat het uitgangssignaal in zoveel mogelijk detail wordt beschreven. Hierbij wordt waar mogelijk de keuze gekoppeld aan een specificatie. Denk ook hier aan grootheden, eenheden, bereik, precisie, frequentie, etc. Een specificatie wordt duidelijk herkenbaar geformuleerd:*

|  |  |
| --- | --- |
| **Specificatie** | *Hier komt de tekst van de specificatie.* |

*Er wordt tevens beschreven dat er een software driver wordt gerealiseerd. Een driver voor een actuator kent in ieder geval een functie om de driver te initialiseren en één of meerder functies om waarden naar de actuator te schrijven. Optioneel kan er een functie voor het lezen van de actuator worden beschreven, bijvoorbeeld om de toestand van een actuator te lezen. Kies als prefix voor de namen van de functies de namen die ook in het architectuurschema zijn gebruikt, of een afkorting daarvan.*

|  |  |
| --- | --- |
| **Specificatie** |  |

### Microcontroller – Communicatie – PC driver – App

*De specificatie van de communicatie met andere apparaten kent twee onderdelen: de interface(s) en het gegevensformaat.*

*Met betrekking tot de interface moet het volgende worden gespecificeerd:*

* *elektrisch – spanning, stroom, etc.*
* *protocol – RS232, I2C, parallel, etc.*
* *protocolinstellingen – zoals bitrate, etc.*

*Daarnaast moet ondubbelzinnig vastgelegd worden hoe data tussen de microcontroller main en app wordt uitgewisseld, oftewel het gegevensformaat. Wordt er gebruik gemaakt van een bestaand gegevensformaat (zoals JSON, XML, CSV, etc.), of wordt er een zelfbedacht gegevensformaat geïmplementeerd? In het geval van dat laatste, dan moet dat gegevensformaat in deze paragraaf ondubbelzinnig gespecificeerd worden.*

*Tevens wordt er beschreven dat er een software driver wordt gerealiseerd. Een driver voor communicatie wordt wel voor een microcontroller gerealiseerd, maar niet voor de PC. Die laatste is namelijk doorgaans beschikbaar. Een communicatiedriver voor een microcontroller kent een initialisatiefunctie, een schrijffunctie en een leesfunctie. De parameters zijn afhankelijk van het gekozen gegevensformaat.*

|  |  |
| --- | --- |
| **Specificatie** |  |

## Software

*Van het hoofdprogramma worden een of meer ontwerpen van de software getoond en beschreven. Er zijn verschillende methoden om zo’n ontwerpen te beschrijven, zoals een flowchart, toestandsdiagram, sequencediagram, klassendiagram, etc. Uit de beschrijvingen moet duidelijk blijken welke architectuur er gekozen is, bijvoorbeeld event driven, cyclic executive met interrupts, een RTOS, een toestandsmachine, of iets dergelijks.*

# 

# Realisatie

*Details van de gerealiseerde hardware en software met bijbehorende toelichting en berekeningen (zoals voedingsstromen, waarden van componenten, etc.). Complete detailschema’s van de hardware en listings van de software worden in de bijlagen opgenomen.*

## Hardware

*Aan de hand van aansluitschema’s wordt de gerealiseerde hardware toegelicht. Het werkt soms ook verhelderend om een afbeelding op te nemen van bijvoorbeeld een gerealiseerde PCB. Gebruik bij voorkeur afbeeldingen van een deel van het aansluitschema. Niet alles hoeft te worden toegelicht. Kies twee of drie van de meest relevante deelsystemen. Het complete aansluitschema moet terug te vinden zijn in de bijlagen.*

## Software

*Aan de hand van code snippets wordt de gerealiseerde software toegelicht. Zorg ervoor dat de code goed leesbaar is middels syntax highlighting. Gebruik code snippets die niet langer zijn dan 20 regels en dat iedere regel code op één regel van het rapport past. Niet alle gerealiseerde code hoeft te worden toegelicht. Kies twee of drie van de meest relevante deelsystemen. De volledige code wordt opgenomen als bijlage. Besteed ook aandacht aan de software ontwikkelomgeving. Vraag je hierbij af wat belangrijke informatie is voor een collega engineer die voor het eerst dezelfde ontwikkelomgeving gaat gebruiken.*

# 

# Testen

*Ondubbelzinnige weergave hoe het systeem, de hardware en/of software getest is. Welke hardware en of software modules zijn getest, hoe zijn de functionele specificaties getest tijdens de acceptatietest? Welke testopstelling is gebruikt en wat zijn de uiteindelijke resultaten. Voldoen de testen aan de gestelde eisen? De resultaten worden voorzien van een duidelijk omschrijving welke eventuele problemen er nog zijn en hoe deze mogelijk zijn te verklaren. Zijn er eventuele 'work arounds' uitgevoerd tijdens het testen? De testen moeten zodanig omschreven zijn dat elke test door anderen te reproduceren is.*

# Conclusies en aanbevelingen

*Reflectie op de doelen van het project. Wat zijn de resultaten? Wat is wel en wat is niet gerealiseerd? Wat kan er aan het product worden aangevuld, uitgebreid, verbeterd?*

# Verwijzingen

Adrián Sánchez Cano. (2013, 3 5). *Using RTC module on FRDM-KL25Z.* Opgehaald van https://community.nxp.com/docs/DOC-94734

ARM. (2022, 04 26). *µVision® IDE*. Opgehaald van https://www2.keil.com/mdk5/uvision/

ARM Developer. (2022, 04 26). *KAN232 - MDK V5 Lab for Freescale Freedom KL25Z Board*. Opgehaald van https://developer.arm.com/documentation/kan232/latest/

Berckel, M. v.-v. (2017). *Schrijven voor technici.* Noordhoff Uitgevers B.V.

contributors, W. (2022, 07 06). *MoSCoW method*. (Wikipedia, The Free Encyclopedia) Opgeroepen op 07 06, 2022, van https://en.wikipedia.org/w/index.php?title=MoSCoW\_method&oldid=1091822315

contributors, W. (2022, 05 25). *SMART criteria*. (Wikipedia, The Free Encyclopedia) Opgeroepen op 07 07, 2022, van https://en.wikipedia.org/w/index.php?title=SMART\_criteria&oldid=1089766780

ELECFREAKS. (2022, 04 19). Opgehaald van Ultrasonic Ranging Module HC - SR04: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

Freescale Semiconductor, I. (sd). *FRDM-KL25Z Pinouts (Rev 1.0).* Opgeroepen op 3 31, 2023, van https://www.nxp.com/document/guide/get-started-with-the-frdm-kl25z:NGS-FRDM-KL25Z

Freescale Semiconductor, Inc. (2012, 9). *KL25 Sub-Family Reference Manual, Rev. 3.*

Freescale Semiconductor, Inc. (2013, 10 24). *FRDM-KL25Z User's Manual, Rev. 2.0.* Opgehaald van https://www.nxp.com/document/guide/get-started-with-the-frdm-kl25z:NGS-FRDM-KL25Z

Freescale Semiconductor, Inc. (2014, 08). *Kinetis KL25 Sub-Family, 48 MHz Cortex-M0+ Based Microcontroller with USB, Rev 5.*

Hmneverl. (2015, 11 18). *De beslismatrix, het maken van keuzes*. (Info.NU.nl) Opgeroepen op 07 06, 2022, van https://mens-en-samenleving.infonu.nl/diversen/164525-de-beslismatrix-het-maken-van-keuzes.html

NXP. (2022, 04 19). *Kinetis® KL2x-72/96 MHz, USB Ultra-Low-Power Microcontrollers (MCUs) based on Arm® Cortex®-M0+ Core*. Opgehaald van https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/kl-series-cortex-m0-plus/kinetis-kl2x-72-96-mhz-usb-ultra-low-power-microcontrollers-mcus-based-on-arm-cortex-m0-plus-core:KL2x?tab=Buy\_Parametric\_Tab#/

NXP. (2022). *OpenSDA Serial and Debug Adapter*. Opgehaald van https://www.nxp.com/design/software/development-software/sensor-toolbox-sensor-development-ecosystem/opensda-serial-and-debug-adapter:OPENSDA?&tid=vanOpenSDA

Solomon Systech Limited. (2008, 4). *SSD1306: Advanced Information.* Opgehaald van https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf

Vishay Semiconductors. (2017, 8 9). *TCRT5000(L), Reflective Optical Sensor with Transistor Output, Rev. 1.7*.

# Bijlage A

# Bijlage B

# Bijlage n