***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

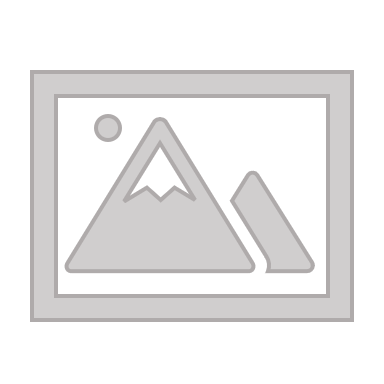
  Automatisch gegenereerde beschrijving) weergeven worden geselecteerd.*

**Embedded Systems Engineering**

Embedded Datalogger voor de validatie van tijdwaarneming bij detectielusverstoringen in verkeersregelinstallaties

**Productrapport**

*Ontwerp en ontwikkeling van een systeem voor efficiënte validatie van tijdwaarneming in verkeersregelinstallaties*



***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](#_Toc193037909)

[Voorwoord 4](#_Toc193037910)

[Samenvatting 5](#_Toc193037911)

[Inhoudsopgave 6](#_Toc193037912)

[1 Inleiding 8](#_Toc193037913)

[1.1 Aanleiding 8](#_Toc193037914)

[1.2 Doelstelling 8](#_Toc193037915)

[1.3 Structuur van het rapport 9](#_Toc193037916)

[2 Functioneel ontwerp 10](#_Toc193037917)

[2.1 Functionele specificaties 10](#_Toc193037918)

[2.1 Functionele eisen 10](#_Toc193037919)

[2.2 Technische eisen 12](#_Toc193037920)

[2.2 User interface 14](#_Toc193037921)

[3 Technisch ontwerp 16](#_Toc193037922)

[3.1 Onderzoek & Analyse 16](#_Toc193037923)

[3.1.1 Vergelijking van bestaande dataloggers 16](#_Toc193037924)

[3.1.2 Keuze van hardwarecomponenten 17](#_Toc193037925)

[3.1.3 Software-architectuur 19](#_Toc193037926)

[3.1.4 Prestatie-eisen voor tijdsregistratie 19](#_Toc193037927)

[3.1.6 Keuze van de software-omgeving 19](#_Toc193037928)

[3.1.7 Keuze van het GPS-systeem 20](#_Toc193037929)

[3.1.5 Validatie & Testplan 20](#_Toc193037930)

[3.2 Architectuur 21](#_Toc193037931)

[3.3 Interfaces 22](#_Toc193037932)

[Voedingsspanning 22](#_Toc193037933)

[Microcontroller – Sensor 22](#_Toc193037934)

[Microcontroller – Actuator 23](#_Toc193037935)

[Microcontroller – Communicatie – PC driver – App 23](#_Toc193037936)

[3.4 Software 24](#_Toc193037937)

[4 Realisatie 25](#_Toc193037938)

[4.1 Hardware 25](#_Toc193037939)

[4.2 Software 25](#_Toc193037940)

[5 Testen 26](#_Toc193037941)

[6 Conclusies en aanbevelingen 27](#_Toc193037942)

[7 Verwijzingen 28](#_Toc193037943)

[Bijlage A 29](#_Toc193037944)

[Bijlage B 30](#_Toc193037945)

[Bijlage n 31](#_Toc193037946)

# Inleiding

Het project “Embedded Datalogger voor de validatie van tijdwaarneming bij detectielusverstoringen in verkeersregelinstallaties” omvat het ontwikkelen van een datalogger door middel van het programmeren van een Arduino Nano ESP32-S3.

Het hoofddoel van dit project is het opslaan van tijdstempels tijdens de detectie van een reflector. Dit hoofdstuk introduceert de basisconcepten en de technische aspecten van het project.

## Aanleiding

Bij ernstige verkeersongevallen onderzoekt het team Forensische Opsporing de toedracht. Op door verkeerslichten geregelde kruispunten slaat de verkeersregelinstallatie (VRI) relevante data op. Wanneer een voertuig het kruispunt nadert, registreert een detectielus in het wegdek verstoringen in het magnetische veld. Deze verstoringen worden gelogd en kunnen inzicht geven in factoren zoals voertuigsnelheid en roodlichtnegatie.

Een cruciaal aspect hierbij is de nauwkeurigheid van de tijdstempels. De VRI-computer registreert tijdstempels met een resolutie van 0,1 seconde (10 Hz), wat kan leiden tot afrondingsverschillen. Deze tijdsregistratie wordt gebruikt om de snelheid van een voertuig tussen de detectielussen te berekenen. De politie gebruikt een datalogger om de tijdstempels van de VRI-computer te valideren en eventuele afrondingsverschillen vast te stellen. In een metingstabel worden de tijdwaarnemingen van de VRI en de datalogger naast elkaar gezet, inclusief het verschil tussen beide. Zelfs een afwijking van 0,1 seconde kan leiden tot een snelheidsverschil van tot wel 90 km/h.

De huidige datalogger voldoet echter niet meer aan de gestelde eisen. De tekortkomingen liggen voornamelijk bij verouderde hardware en software, en het ontbreken van bepaalde functionaliteiten. Daarom is er een behoefte aan een nieuw en verbeterd model dat voldoet aan de moderne eisen op het gebied van betrouwbaarheid, gebruiksgemak en toekomstbestendigheid. Dit project biedt de mogelijkheid om een dergelijke datalogger te ontwikkelen.

## Doelstelling

Het doel van dit project is om de hoofdonderzoeksvraag te beantwoorden:  
"Welke hard- en software is het meest efficiënt en effectief voor een datalogger die nauwkeurig de tijdwaarneming kan loggen na een inkomend signaal?"

Om deze vraag te beantwoorden, is onderzoek gedaan naar geschikte componenten. Hierbij zijn verschillende hardware- en softwareopties geanalyseerd op basis van datasheets en onderlinge vergelijking. De meest efficiënte en effectieve componenten zijn vervolgens geselecteerd en samengebracht in een functionerend prototype.

Dit prototype is getest en vergeleken met bestaande dataloggers op het gebied van nauwkeurigheid en prestaties. Op basis van de testresultaten wordt beoordeeld of het nieuwe ontwerp voldoet aan de gestelde eisen.

Als randvoorwaarden en uitgangspunten werden door mij aangenomen dat:

* Detectie van referentiepunten met behulp van een optisch meetsysteem;
* Tijdwaarneming met hoge precisie (10 KHz) voor nauwkeurige tijdwaarneming;
* Opslag op SD-kaart van ten minste 20 metingen voor latere analyse;
* Een gebruikersinterface waarmee instellingen kunnen worden aangepast en meetgegevens kunnen worden gecontroleerd;
* Robuuste en modulaire architectuur, zodat de datalogger eenvoudig kan worden uitgebreid met extra functionaliteiten.

## Structuur van het rapport

De opbouw van dit rapport bestaat uit de volgende hoofdstukken:

* In hoofdstuk 2 is het functionele ontwerp te lezen. Het functioneel ontwerp beschrijft de technische en de functionele eisen, hierin is beschreven welke producten zijn gebruikt en hoe de datalogger werkt.
* In hoofdstuk 3 is het technisch ontwerp beschreven. Hierin is beschreven hoe de technische oplossingen zijn ontworpen. Aan de hand van een diagram is te zien hoe de architectuur van de datalogger is opgebouwd. Daarnaast is beschreven hoe de deelsystemen en de software-architectuur zijn vormgegeven.
* In hoofdstuk 4 is de realisatiefase en de testfase beschreven. In de realisatiefase is beschreven hoe de hardware en software ontwerpen tot realisatie zijn gebracht. Aan de hand van voorbeelden en schema's wordt de realisatie in details uitgelegd.
* In hoofdstuk 5 worden de testresultaten beschreven.
* In hoofdstuk 6 wordt afgesloten met het eindresultaat en enkele aanbevelingen. Er wordt teruggekeken naar het project en daarmee wordt er een conclusie beschreven.
* In hoofdstuk 7 worden de bronvermeldingen weergegeven.
* Aan het einde van het rapport bevindt zich de bijlagen, de bijlagen bevat onder andere de gebruikershandleiding van de datalogger en de pin-bezetting van de microcontroller.

# Functioneel ontwerp

## Functionele specificaties

Dit document beschrijft het ontwerp van de Datalogger en de bijbehorende functionele en technische eisen. Het doel van de datalogger is om tijdstempels te registreren bij detectie van een object door een afstandssensor en de verzamelde gegevens overzichtelijk weer te geven. De datalogger zal real-time informatie tonen en opslaan en beschikt over een gebruiksvriendelijke interface voor de bediening.

De functionele en technische eisen zijn opgesteld volgens de MoSCoW-methode(contributors, MoSCoW method, 2022), waarbij onderscheid wordt gemaakt tussen Must-have, Should-have, Could-have en Won’t-have specificaties. Dit helpt bij het prioriteren van functies en waarborgt dat de belangrijkste functionaliteiten worden geïmplementeerd binnen de beschikbare tijd en middelen.

### 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 tijdstempels 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 (128x64). | Duidelijke en efficiënte weergave. |
| F1.3 | Must (M) | Meetfrequentie | De datalogger moet een meetfrequentie van 1000 Hz ondersteunen en de gegevens na detectie verwerken en opslaan. | Zorgt voor een nauwkeurige tijdregistratie. |
| F1.4 | Must (M) | Opslagcapaciteit | De datalogger moet minimaal 20 metingen kunnen opslaan op een extern geheugen, bij voorkeur een SD-kaart. | Voldoende opslagruimte voor veldmetingen. |
| F1.5 | Must (M) | Detectie | De datalogger moet signalen van een afstandsdetector kunnen verwerken, met een detectiebereik van tenminste 100 cm en binnen 1 ms een meetwaarde registreren. | Moet een reflector kunnen detecteren die een detectielusverandering simuleert. |
| F1.6 | Must (M) | Status opgeslagen metingen | Het scherm toont het aantal opgeslagen detecties. | Helpt de gebruiker bij het beheren van metingen en ontdekken van mogelijke valse detecties. |
| 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 tijdstempels worden in lokale tijden geregistreerd. | Dit maakt het makkelijker om de tijdstempels van de VRI met elkaar te vergelijken. |
| F1.9 | Must (M) | Live weergave | Aan de rechterzijde op het OLED-scherm wordt weergegeven of de SD-kaart aanwezig is en of de sensor een detectie heeft geregistreerd. | Real-time visualisatie van metingen. |
| F1.10 | Must (M) | Aantal opgeslagen ritten | Aan de rechterzijde worden het aantal ritten weergegeven dat is opgeslagen, | Helpt bij het monitoren van voltooide metingen. |
| F1.11 | Must (M) | Bediening via buttons | Onderin het scherm komen 4 buttons, boven de buttons worden op het scherm de betekenis van de buttons op het OLED-scherm weergegeven START, KLOK, GPS, STOP. | Fysieke knoppen zorgen voor snelle bediening. |
| F1.12 | Must (M) | Meting starten en stoppen | Met de button “START” wordt het frame gestart waarbij de datalogger tijdstempels registreert na detecties van de afstandssensor. Met het bedienen van de STOP-button wordt het frame gesloten en de detecties opgeslagen. Tevens wordt er één rit opgeteld. | Zelf moment bepalen wanneer de datalogger reageert op de sensor. |
| F1.13 | Must (M) |  |  |  |
| F1.12 | Must (M) | Datalog exporteren | De meetgegevens worden opgeslagen op een SD-kaart in csv-formaat. | Handig voor verdere analyse. |
| F1.13 | 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 | Should (S) | Draadloze communicatie | Optioneel kan Bluetooth of WiFi worden toegevoegd voor draadloze data-export. | Handig voor realtime monitoring zonder kabels. |
| F1.15 | Wont’t (W) | Ai-analyse | Het systeem zal geen automatische snelheidsberekening of patroonherkenning implementeren. | Goede toevoeging om efficiënt en effectief te werken. |

### 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 Arduino Nano ESP32-S3 microcontroller. |
| T2 | Must (M) | De hardware wordt ontworpen als een uitbreidingsmodule (shield) voor de Arduino Nano ESP32-S3. |
| T3 | Must (M) | De firmware wordt geschreven in **C++ en C** (C11-standaard voor RTOS). |
| T4 | Must (M) | Het systeem moet via UART, SPI en I2C of USB data kunnen exporteren. |
| T5 | Must (M) | De datalogger moet worden gevoed door een 12V USB-voeding of batterijpack. |
| T6 | Must (M) | De voeding wordt gereguleerd via een step-down converter om een stabiele 3,3V aan de microcontroller te leveren. |
| T7 | Must (M) | De datalogger moet bij stroomuitval minimaal 5 seconden noodstroom behouden. |
| T8 | Must (M) | De datalogger wordt beveiligd tegen spanningspieken. |
| T9 | Must (M) | |  | | --- | |  |  |  | | --- | | De datalogger moet ESD-bescherming bevatten om schade door elektrostatische ontladingen te voorkomen. | |
| T10 | Must (M) | |  | | --- | | De microcontroller moet een RTOS (Real-Time Operating System) ondersteunen. |  |  | | --- | |  | |
| T11 | Must (M) | De externe RTC moet een nauwkeurigheid hebben van maximaal 2 PPM. |
| T12 | Must (M) | |  | | --- | |  |  |  | | --- | | Het OLED-scherm moet minimaal 128x64 pixels zijn en dimfunctie hebben. | |
| T13 | Must (M) | |  | | --- | |  |  |  | | --- | | De sensor voor reflectorpaaltjes moet een max. reactietijd van 1 ms en schakelfrequentie van 1 kHz hebben. | |
| T14 | Must (M) | |  | | --- | | De SD-kaartmodule moet een minimale schrijfsnelheid van Class 10 ondersteunen. |  |  | | --- | |  | |
| T15 | Could (C) | De RTC moet automatisch kunnen synchroniseren met een **GPS-signaal**. |

## User interface

Op de voorzijde van de datalogger (zie figuur 1) bevindt zich in het midden een OLED-scherm (128x64 pixels) waarop real-time metingen en statusinformatie worden weergegeven.

Direct onder het scherm bevinden zich vier drukknoppen, die corresponderen met de tekst erboven. Deze knoppen stellen de gebruiker in staat om instellingen aan te passen en functionaliteiten te bedienen:

* Stop – Stopt de huidige meting.
* Erase – Wis de laatst opgeslagen meting.
* Next – Ga naar de volgende meting.
* Start – Start een nieuwe meting.

Aan de rechterzijde van het OLED-scherm bevindt zich een potmeter, waarmee de helderheid van het scherm kan worden aangepast (scherm dimmen).

**Fysieke In- en Output**

Inputs:

* Drukknoppen voor gebruikersbediening.
* Infraroodsensor voor het detecteren van objecten.
* SD-kaartsleuf voor gegevensopslag.

Outputs op het OLED-scherm toont:

* Aantal opgeslagen samples.
* Meetwaarden (tijdstempels, aantal detecties en aantal geslaagde metingen).
* Systeemstatus (bijv. foutmeldingen, bevestiging van acties).
* Visuele feedback op het OLED-scherm bij interactie (bijvoorbeeld een wijziging van het aantal samples bij een meting).

**Dynamiek tussen Input en Output**

**Wanneer een object wordt gedetecteerd (via de afstandssensor):**

* De tijdstempel en bijbehorende meetwaarde worden weergegeven op het OLED-scherm.
* Het aantal geregistreerde samples neemt toe.
* De visuele indicator op het scherm geeft de detectie weer.

**Wanneer de gebruiker op "Erase" drukt:**

* De laatst opgeslagen meting wordt verwijderd.
* De waarde van het aantal geregistreerde samples neemt af.

**Wanneer de gebruiker op "Start" drukt:**

* Een nieuwe meting wordt gestart.
* De interface toont de actuele metingen.

**Wanneer de gebruiker de schermhelderheid aanpast:**

* De helderheid van het OLED-scherm verandert.

*.*

*Afbeelding met tekst, schets, diagram, tekening

Door AI gegenereerde inhoud is mogelijk onjuist.*

Figuur User interface schets

# 

# 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.*

## Onderzoek & Analyse

**Inleiding**

Om een geschikte datalogger te ontwikkelen voor de validatie van tijdwaarneming bij detectielusverstoringen, is een gedegen onderzoek uitgevoerd. Dit onderzoek richt zich op de keuze van hardware en software, waarbij nauwkeurigheid, betrouwbaarheid en efficiëntie centraal staan. Zie voor het volledige overzicht van het onderzoek de bijlage.

**De hoofdonderzoeksvraag luidt:**  
"Welke hard- en software is het meest efficiënt en effectief voor een datalogger die nauwkeurig de tijdwaarneming kan loggen na een inkomend signaal?"

Om deze vraag te beantwoorden, zijn verschillende aspecten onderzocht:

* Vergelijking van bestaande dataloggers die momenteel worden gebruikt binnen de politie.
* Keuze van hardwarecomponenten, waaronder microcontrollers, afstandssensoren en opslagmedia.
* Software-architectuur en verwerking van tijdstempels.
* Prestatie-eisen voor een nauwkeurige tijdsregistratie en logging.

### Vergelijking van bestaande dataloggers

Momenteel worden binnen de politie twee systemen gebruikt:

* Data Translation USB Data Acquisition (DAQ) Module
* Movilog BM22

Deze systemen zijn respectievelijk bedoeld voor validatie van verkeersregelinstallaties en remvertragingmetingen met GPS.

Evaluatie van de huidige systemen:

* De DAQ-module biedt hoge precisie, maar is beperkt in mobiliteit en gebruiksvriendelijkheid.
* De Movilog BM22 heeft een compact ontwerp, maar kent softwarefouten, een onpraktische montagelocatie en een onhandige voedingsaansluiting.

Conclusie:  
De nieuwe datalogger moet de sterke punten van de BM22 behouden, maar verbeterd worden op het gebied van betrouwbaarheid, voeding en gebruiksgemak.

### 3.1.2 Keuze van hardwarecomponenten

**Microcontroller**  
Om de juiste microcontroller te kiezen, zijn de volgende criteria opgesteld:

* Ondersteuning voor RTOS en real-time verwerking.
* Hoge CPU-snelheid en voldoende RAM/Flash-geheugen.
* Ondersteuning voor SPI, I²C en UART voor communicatie met externe modules.

| ***Microcontroller*** | ***CPU*** | ***RAM/Flash*** | ***RTOS-ondersteuning*** | ***Opmerking*** |
| --- | --- | --- | --- | --- |
| ***Arduino Nano ESP32-S3*** | *2x 240 MHz* | *512KB RAM / 8MB Flash* | *Ja* | ***Gekozen*** *vanwege Arduino-ecosysteem en dual-core.* |
| *Teensy 4.1* | *600 MHz* | *1024KB RAM / 8MB Flash* | *Ja* | *Krachtig, maar minder ondersteuning.* |
| *STM32F4* | *180 MHz* | *192KB RAM / 512KB Flash* | *Ja* | *Betrouwbaar, maar complexer in implementatie.* |
| *NXP KL25Z* | *48 MHz* | *32KB RAM / 128KB Flash* | *Ja* | *Niet krachtig genoeg.* |

Tabel Vergelijking van de microcontrollers.

**Gekozen microcontroller:**  
**De Arduino Nano ESP32-S3** is gekozen vanwege de balans tussen prestaties, ondersteuning en eenvoudige implementatie.

**Afstandssensor**

Voor de detectie van objecten is een **roodlichtsensor** gekozen vanwege de betrouwbaarheid onder verschillende lichtomstandigheden.

| ***Sensor*** | ***Schakelfrequentie*** | ***Aanspreektijd*** | ***Voedingsspanning*** | ***Opmerking*** |
| --- | --- | --- | --- | --- |
| ***Leuze PRK25C.D1/4P-200-M12*** | ***1,5 kHz*** | ***0,33 ms*** | *10-30V* | ***Gekozen vanwege betrouw-baarheid en compatibiliteit.*** |
| *SICK WL12G-3B2531* | *1 kHz* | *0,5 ms* | *12-24V* | *Te traag.* |
| *ifm O5D150* | *1 kHz* | *0,5 ms* | *10-30V* | *Goede optie, maar minder nauwkeurig.* |

Tabel : Vergelijking van de afstandssensoren.

**Gekozen sensor:**  
De **Leuze PRK25C.D1/4P-200-M12** voldoet het beste aan de prestatie-eisen.

**Opslagmedium**  
Er is gekozen voor een **SD-kaartmodule** vanwege de hoge opslagcapaciteit en compatibiliteit met embedded systemen.

| ***Opslagtype*** | ***Max. schrijfsnelheid*** | ***Levensduur*** | ***Opmerking*** |
| --- | --- | --- | --- |
| ***SD-kaart (Class 10, SPI)*** | *10 MB/s* | *100.000 cycli* | ***Gekozen vanwege snelheid en eenvoudige implementatie.*** |
| *EEPROM* | *1 MB/s* | *1 miljoen cycli* | *Geschikt voor kleine datasets, maar beperkt in capaciteit.* |
| *FRAM* | *5 MB/s* | *Onbeperkt* | *Ideaal als buffer, maar duurder.* |

Tabel : Vergelijking van opslagmediums.

**Gekozen opslagmedium:**  
**Een Class 10 SD-kaart** wordt gebruikt voor hoge snelheid en compatibiliteit met de ESP32-S3.

### 3.1.3 Software-architectuur

De software van de datalogger bestaat uit vier lagen:

* Sensorinterface – Verwerking van inkomende data via interrupts.
* Tijdstempelbeheer – Nauwkeurige registratie met RTC en GPS-synchronisatie.
* Opslagbeheer – Data wordt tijdelijk in RAM opgeslagen en in batches naar de SD-kaart geschreven.
* Uitvoer & uitlezing – OLED-display en seriële communicatie voor monitoring.

Optimalisaties:

* FIFO-buffering voorkomt data-verlies bij hoge meetfrequenties.
* Batch-schrijfoperaties naar de SD-kaart verminderen slijtage.
* Interrupt-gebaseerde verwerking zorgt voor minimale CPU-belasting.

### 3.1.4 Prestatie-eisen voor tijdsregistratie

Voor een nauwkeurige tijdsvergelijking moeten de volgende eisen worden gehaald:

| ***Stap*** | ***Max. vertraging*** | ***Optimalisatie*** |
| --- | --- | --- |
| *Interrupt detectie* | *< 10 µs* | *Hardware interrupts.* |
| *Timestamp opslag* | *< 50 µs* | *Gebruik van micros() of RTC.* |
| *Tijdsvergelijking* | *< 500 µs* | *Integer-math in plaats van floating point.* |
| *Schrijven naar SD-kaart* | *< 10 ms* | *Buffer en batch-opslag.* |

Tabel tijdsvergelijking prestatie-eisen.

### 3.1.6 Keuze van de software-omgeving

De software van de datalogger moet voldoen aan de volgende eisen:

* Real-time verwerking van inkomende signalen zonder vertraging.
* Efficiënte opslag van tijdstempels en metingen.
* Compatibiliteit met de gekozen hardware (ESP32-S3, SD-kaart, GPS).

**Keuze van het besturingssysteem**

Voor de software-implementatie is een afweging gemaakt tussen bare-metal programming en een RTOS (Real-Time Operating System).

| **Softwarebenadering** | **Voordelen** | **Nadelen** |
| --- | --- | --- |
| **Bare-metal (Arduino Core)** | Directe hardwarecontrole, minder overhead | Moeilijk schaalbaar, minder gestructureerd |
| **FreeRTOS (ESP-IDF)** | Ondersteuning voor multitasking, betere timingcontrole | Hogere complexiteit |
| **Zephyr RTOS** | Extreem flexibel, krachtige scheduling | Steile leercurve, minder community-ondersteuning |

Tabel Vergelijking software omgeving.

**Gekozen oplossing:**  
Omdat real-time verwerking en betrouwbare multitasking nodig zijn, is gekozen voor FreeRTOS binnen het ESP-IDF-framework. Dit zorgt ervoor dat:

* Data-acquisitie en opslag parallel kunnen verlopen.
* Tijdsregistratie via interrupts verwerkt wordt zonder vertraging.
* GPS-synchronisatie en SD-kaartopslag efficiënt worden beheerd.

### 3.1.7 Keuze van het GPS-systeem

Om de tijdssynchronisatie van de datalogger te verbeteren, wordt een GPS-module gebruikt.

De module moet:

* NMEA 0183-compatibel zijn voor eenvoudige integratie.
* Hoge nauwkeurigheid bieden (≤1 ms afwijking).
* Een snelle TTFF (Time To First Fix) hebben voor snelle initialisatie.

| **GPS-module** | **Tijdsafwijking** | **TTFF (koude start)** | **Interface** | **Opmerking** |
| --- | --- | --- | --- | --- |
| **u-blox NEO-M8N** | ±30 ns | 26 s | UART, I2C | **Gekozen vanwege hoge nauwkeurigheid en betrouwbaarheid.** |
| u-blox NEO-6M | ±60 ns | 40 s | UART | Langzamere TTFF, oudere chip. |
| Quectel L86 | ±50 ns | 35 s | UART | Goede optie, maar minder documentatie. |

Tabel Vergelijking van GPS-modules.

**Gekozen oplossing:**  
De u-blox NEO-M8N is gekozen vanwege:

* Zeer hoge tijdsnauwkeurigheid (~30 nanoseconden).
* Snelle fix-tijd en robuuste prestaties onder verschillende omstandigheden.
* Ondersteuning voor I²C en UART, wat flexibiliteit biedt voor communicatie met de ESP32-S3.

De datalogger zal de **UTC-tijd van de GPS-module** gebruiken als referentie en de interne RTC kalibreren om afrondingsfouten te minimaliseren.

### 3.1.5 Validatie & Testplan

Om de nauwkeurigheid van de datalogger te testen, wordt deze vergeleken met een referentie-GPS-logger.

**Testprocedure:**

1. Monteer de datalogger en een GPS-logger op een testvoertuig.
2. Rij een vaste route met meerdere detectiepunten.
3. Vergelijk de tijdstempels van beide systemen.

Maximale afwijking moet binnen de foutmarge van de RTC/GPS blijven.

**Conclusie**

Het onderzoek heeft geleid tot de keuze voor **de Arduino Nano ESP32-S3**, **de Leuze PRK25C.D1/4P-200-M12 sensor** en een **Class 10 SD-kaart**. De software-architectuur is geoptimaliseerd voor snelheid en betrouwbaarheid, en de prestatie-eisen waarborgen een nauwkeurige tijdsregistratie.

Met deze keuzes voldoet de datalogger aan de eisen en biedt het een betrouwbare oplossing voor de validatie van tijdwaarneming bij detectielusverstoringen.

| ***Component*** | ***Gekozen optie*** | ***Reden*** |
| --- | --- | --- |
| ***Microcontroller*** | *Arduino Nano ESP32-S3* | *Balans tussen prestaties en ondersteuning* |
| ***Afstandssensor*** | *Leuze PRK25C.D1/4P-200-M12* | *Betrouwbare detectie, snelle responstijd* |
| ***Opslag*** | *Class 10 SD-kaart* | *Snelle schrijfsnelheid, eenvoudige implementatie* |
| ***Besturingssysteem*** | *FreeRTOS (ESP-IDF)* | *Real-time verwerking, efficiënte multitasking* |
| ***GPS-module*** | *u-blox NEO-M8N* | *Hoge tijdsnauwkeurigheid, snelle fix-tijd* |

Tabel Overzicht gekozen hard- en software.

## 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.*

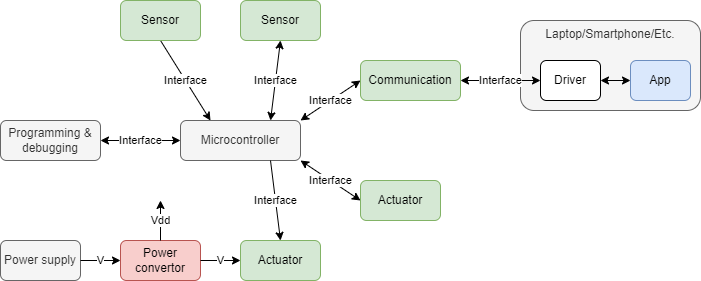
*In Figuur 5 wordt een architectuur getoond die algemeen toepasbaar is voor embedded systemen. Het hart is de microcontroller die communiceert via verschillende interfaces met de deelsystemen. Deze interfaces moeten ondubbelzinnig worden vastgelegd in deze fase van het project.*

*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 2. 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