***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*

Afbeelding met Landvoertuig, band, voertuig, buitenshuis

Door AI gegenereerde inhoud is mogelijk onjuist.

**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](#_Toc198323038)

[Voorwoord 4](#_Toc198323039)

[Samenvatting 5](#_Toc198323040)

[Inhoudsopgave 6](#_Toc198323041)

[1 Inleiding 8](#_Toc198323042)

[1.1 Aanleiding 8](#_Toc198323043)

[1.2 Doelstelling 8](#_Toc198323044)

[1.3 Structuur van het rapport 9](#_Toc198323045)

[2 Probleemanalyse & Ontwerpkeuze 10](#_Toc198323046)

[2.1 Opdrachtomschrijving 10](#_Toc198323047)

[2.2 Kernprobleem 10](#_Toc198323048)

[2.3 Analyse van mogelijke oplossingen 10](#_Toc198323049)

[2.4 Gekozen oplossing en onderbouwing 10](#_Toc198323050)

[3 Functioneel ontwerp 11](#_Toc198323051)

[3.1 Functionele specificaties 11](#_Toc198323052)

[3.2 Functionele eisen 11](#_Toc198323053)

[3.3 Technische eisen 14](#_Toc198323054)

[3.4 User interface 15](#_Toc198323055)

[4 Technisch ontwerp 17](#_Toc198323056)

[4.1 Architectuur 17](#_Toc198323057)

[4.2 Interfaces 18](#_Toc198323058)

[Microcontroller – Communicatie – PC driver – App 25](#_Toc198323059)

[4.3 Software 25](#_Toc198323060)

[5 Realisatie 26](#_Toc198323061)

[5.1 Hardware 26](#_Toc198323062)

[5.2 Software 26](#_Toc198323063)

[6 Testen 27](#_Toc198323064)

[7 Conclusies en aanbevelingen 28](#_Toc198323065)

[8 Verwijzingen 29](#_Toc198323066)

[Bijlage A 30](#_Toc198323067)

[Bijlage B 31](#_Toc198323068)

[Bijlagen 32](#_Toc198323069)

# 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 (1 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 worden de probleemanalyse en ontwerpkeuze behandeld. In dit hoofdstuk wordt de opdrachtomschrijving toegelicht en het kernprobleem geformuleerd. Vervolgens wordt het probleem geanalyseerd en de gekozen oplossing gepresenteerd, inclusief een onderbouwing van de gemaakte ontwerpkeuzes.
* In hoofdstuk 3 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 4 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 5 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 6 worden de testresultaten beschreven.
* In hoofdstuk 7 wordt afgesloten met het eindresultaat en enkele aanbevelingen. Er wordt teruggekeken naar het project en daarmee wordt er een conclusie beschreven.
* In hoofdstuk 8 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.

# Probleemanalyse & Ontwerpkeuze

## Opdrachtomschrijving

Het team Forensische Opsporing van de politie eenheid Zeeland-West-Brabant gebruikt een datalogger om tijdstempels van verkeersregelinstallaties (VRI’s) te valideren. Bij verkeersongevallen waarbij vermoedelijk sprake is van roodlichtnegatie of een snelheidsovertreding, worden de loggegevens van de detectielussen onderzocht. Deze detectielussen registreren verstoringen die veroorzaakt worden door passerende voertuigen en slaan bijbehorende tijdstempels op. De bestaande datalogger die momenteel gebruikt wordt voor de validatieproeven is sterk verouderd en niet flexibel genoeg voor innovatie. De opdracht is om een nieuwe, nauwkeurige en uitbreidbare datalogger te ontwikkelen, gebaseerd op moderne hard- en software.

## Kernprobleem

De VRI registreert tijdstempels met een resolutie van 0,1 seconde (10 Hz), wat te grof is voor een betrouwbare en nauwkeurige snelheidsberekening. Een verschil van 0,1 seconde kan leiden tot een aanzienlijke afwijking in de berekende snelheid. De bestaande datalogger is sterk verouderd en mist gebruiksgemak. Er is bij de politie behoefte aan een compact systeem dat eenvoudig te bedienen is en tijdstempels met millisecondenprecisie registreert.

## Analyse van mogelijke oplossingen

Tijdens de analyse zijn voor- en nadelen van drie oplossingsrichtingen nader onderzocht:

De eerste optie betreft het herschrijven van de software op de bestaande hardware. Dit is te realiseren en vereist geen nieuwe onderdelen, maar het effect is beperkt en de oude hardware is lastig te vervangen.

De tweede optie betreft de aanschaf van een commerciële datalogger. Deze systemen hebben een professioneel karakter en zijn vaak gecertificeerd, maar duur, beperkt aanpasbaar en vereisen een langdurige aanbestedingstraject.

De derde optie is het ontwikkelen van een nieuwe datalogger. Dit biedt volledige controle over hard- en software, is aanpasbaar aan de wenste functies en goedkoper dan commerciële alternatieven. Nadeel is dat dit meer ontwikkeltijd en tests vraagt.

Vanwege het belang van maatwerk, uitbreidbaarheid en de lage kosten is gekozen voor de derde optie.

## Gekozen oplossing en onderbouwing

De nieuwe datalogger wordt gebouwd rond een Arduino Nano ESP32-S3 vanwege hoge snelheid (240 Mhz), nauwkeurige timingmogelijkheden (1 kHz), uitbreidbaarheid (WiFi, Bluetooth) en ondersteuning voor randapparatuur.

Belangrijke componenten:

* DS3231 RTC - voor betrouwbare tijdwaarneming;
* SD-kaartmodule - voor lokale opslag (CSV-formaat);
* Seeed Studio Grove – Air530Z - voor de plaatsbepaling en aanvullende tijdsbron;
* OLED-scherm + knoppen - voor eenvoudige gebruikersinteractie;
* Modulaire software-architectuur - voor toekomstgerichte uitbreiding en onderhoud.

Zie bijlage 1 voor het volledige componentenonderzoek.

Deze opzet biedt een robuuste, kostenefficiënte en nauwkeurige oplossing voor het valideren van tijdregistraties in VRI-systemen.

# 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 reflecterend 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.

## 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 ritten 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 kort na het beëindigen van de huidige rit. | Helpt de gebruiker bij het beheren van metingen en ontdekken van mogelijke valse detecties. |
| F1.7 | Must (M) | Opslagbehoud | De opgeslagen data moeten 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 wordt het huidige aantal ritten weergegeven. | Helpt bij het monitoren van voltooide metingen. |
| F1.11 | Must (M) | Bediening via drukknoppen | Aan de onderzijde van het OLED-scherm komen 4 drukknoppen, boven de drukknoppen worden op het scherm de betekenis van de drukknoppen 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 drukknop “START” wordt het frame gestart waarbij de datalogger tijdstempels registreert na detecties van de afstandssensor. Met het bedienen van de STOP- drukknop 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) | Lokale tijd synchroniseren | De drukknop “TIJD” opent een menu om de actuele datum en tijd in te stellen. | Tijdsynchronisatie |
| F1.14 | Should (S) | Tijdsynchronisatie aan de hand van GPS | De drukknop “GPS” opent een menu om de actuele datum en tijd te synchroniseren met GPS. | Actuele GPS datum en tijd. |
| F1.15 | Must (M) | Datalog exporteren | De meetgegevens worden opgeslagen op een SD-kaart in CSV-formaat. | Handig voor verdere analyse. |
| F1.16 | Should (S) | GPS-ondersteuning | De datalogger moet real-time GPS-coördinaten opslaan bij elke meting. | Belangrijk voor locatiegebonden data. |
| F1.17 | Should (S) | Aantal Satellieten | Rechtsboven op het display moet het aantal verbonden satellieten worden weergegeven. | Biedt inzicht in signaalsterkte. |
| F1.18 | Could (C) | Energiebeheer | De datalogger moet een energiezuinige modus hebben die automatisch wordt geactiveerd na X minuten inactiviteit. | Verlengt eventueel de batterijduur. |
| F1.19 | Should (S) | Draadloze communicatie | Optioneel kan Bluetooth of WiFi worden toegevoegd voor draadloze data-export. | Handig voor realtime monitoring zonder kabels. |
| F1.20 | Wont’t (W) | Ai-analyse | Het systeem zal geen automatische snelheidsberekening of patroonherkenning implementeren. | Goede toevoeging om efficiënt en effectief te werken. |

## 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++.** |
| 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 wordt beveiligd tegen spanningspieken. |
| T8 | Must (M) | |  | | --- | | De microcontroller moet een RTOS (Real-Time Operating System) ondersteunen. |  |  | | --- | |  | |
| T9 | Must (M) | De externe RTC moet een nauwkeurigheid hebben van maximaal 2 PPM. |
| T10 | Must (M) | |  | | --- | |  |  |  | | --- | | Het OLED-scherm moet minimaal 128x64 pixels zijn en dimfunctie hebben. | |
| T11 | Must (M) | |  | | --- | |  |  |  | | --- | | De sensor voor reflectorpaaltjes moet een max. reactietijd van 1 ms en schakelfrequentie van 1 kHz hebben. | |
| T12 | Must (M) | |  | | --- | | De SD-kaartmodule moet een minimale schrijfsnelheid van Class 10 ondersteunen. |  |  | | --- | |  | |
| T13 | Could (C) | De RTC moet automatisch kunnen synchroniseren met een **GPS-signaal**. |
| T14 | Won’t (W) | Een 3-assige gyroscoop/versnellingsmeter, om de acceleratie van de datalogger in x, y en z richting te meten. |

## 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:

* Drukknop “Start” – Start een nieuwe meting.
* Drukknop “Klok” – Opent een menu om de lokale datum en tijd in te stellen.
* Drukknop “GPS” – Opent een menu voor GPS.
* Drukknop ”Stop” – Stopt de huidige meting.

Afbeelding met tekst, schermopname, diagram, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 1 User interface schets

**Fysieke In- en Output**

Inputs (zie figuur 1):

* Drukknoppen voor gebruikersbediening.
* Infraroodsensor voor het detecteren van objecten, X = detectie en 0 = geen detectie.
* SD-kaartsleuf voor gegevensopslag, X = sd-kaart gedetecteerd en 0 = geen detectie.

Outputs op het OLED-scherm toont (zie figuur 1):

* Aantal opgeslagen ritten.
* Meetwaarden (aantal detecties).
* Systeemstatus (bijv. bevestiging van acties en foutmeldingen zoals ontbreken van de SD-kaart).
* Visuele feedback op het OLED-scherm (bijvoorbeeld of er een detectie is waargenomen door de lichtpoortsensor en/of de SD-kaart is geplaatst).
* Huidige lokale systeemtijd en datum.

**Dynamiek tussen Input en Output**

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

* De visuele indicator op het scherm geeft de detectie weer met een “X”.

**Wanneer de gebruiker drukknop "Klok" bediend (zie figuur 2):**

* Wordt een menu geopend die de huidige datum en tijd weergeeft;
* Met een marker krijgt het actieve veld een contrasterende achtergrond;
* De vier drukknoppen hebben een nieuwe functie om de datum en tijd naar de gewenste waarde in te stellen;
* Met de drukknoppen “+” en “-“ kan de waarde worden opgehoogd of verlaagd;
* Met de drukknop “>” kan het volgende karakter worden ingesteld;
* Met de drukknop “OK” wordt de waarde opgeslagen, het menu beëindigd en hoofdmenu geopend.

**Afbeelding met tekst, schermopname, diagram, cirkel

Door AI gegenereerde inhoud is mogelijk onjuist.**

Figuur 2: Userinterface van klok-menu.

**Wanneer de gebruiker op "Start" drukt (zie figuur 3):**

* Een nieuwe meting wordt gestart.
* De teller “rit” wordt met één opgehoogd;
* De userinfo wordt leeg gemaakt;
* Bovenaan de userinfo verschijnt “Meting gestart”.

Afbeelding met tekst, schermopname, scherm, diagram

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 3: Userinterface na starten meting.

**Wanneer de gebruiker op "Stop" drukt (zie figuur 4):**

* De huidige meting wordt beëindigd;
* De userinfo toont “Meting beëindigd’;
* De userinfo toont het aantal detecties;
* De userinfo toont of de rit is opgeslagen op de SD-kaart;

Afbeelding met tekst, schermopname, Lettertype, nummer

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 4: Userinterface na beëindigen meting.

# 

# Technisch ontwerp

## Architectuur

**Overzicht**

De architectuur van de datalogger (zie figuur 5) bestaat uit een modulaire opzet waarin verschillende deelsystemen met elkaar samenwerken. Het hart van het systeem wordt gevormd door de Arduino Nano ESP32-S3, die communiceert met randcomponenten zoals de afstandssensor, RTC-module, SD-kaartmodule, OLED-scherm en eventueel een GPS-module. Daarnaast is het systeem voorzien van vier fysieke knoppen voor gebruikersinteractie.

Afbeelding met tekst, schermopname, software, Multimediasoftware

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 5. Blokschema datalogger.

**Deelsystemen**

De architectuur is opgebouwd uit de volgende deelsystemen:

* **Inputmodules:**
* Afstandssensor (Sick WL250-2P2431) detecteert object en genereert een triggersignaal.
* Drukknoppen (4 stuks): Start, Stop, Klok, GPS.
* **Outputmodules:**
* OLED-scherm: toont userinfo onder andere de systeemstatus, huidige aantal opgeslagen ritten, detectie van object door afstandssensor, aanwezigheid van sd-kaart en eventueel de GPS-info.
* SD-kaartmodule: Slaat meetgegevens op in CSV-formaat.
* **Tijdregistratie:**
* RTC-module (DS3132): biedt nauwkeurige tijdregistratie (max 2 PPM afwijking).
* **Locatiebepaling:**
* GPS-module (air530Z): geeft de exacte locatie en tijd.
* **Voeding en regeling:**
* 12 Volt voeding: met stepdown naar 3.3 V.
* **Centrale verwerkingseenheid:**
* Arduino Nano ESP32-S3: verwerkt inkomende signalen, regelt logging van inkomende signalen en aansturing van interfaces.

**Architectuurprincipe**

Het systeem is ontworpen volgens het event-driven principe met interruptgestuurde triggers voor het loggen van detecties. De software is opgebouwd met een modulaire structuur, waarbij elk deelsysteem wordt aangestuurd via een aparte driver. Het ontwerp houdt rekening met uitbreidbaarheid en herbruikbaarheid.

## Interfaces

De datalogger maakt gebruik van diversen digitale en elektrische interfaces tussen de microcontroller en de aangesloten modules. De eigenschappen en de toegepaste communicatieprotocollen van de interfaces van de datalogger, worden in deze paragraaf beschreven. Ook kan er een verwezen worden naar de bijbehorende softwaredrivers.

**Voedingsspanning**

Specificatie

De datalogger wordt gevoed door een externe 12V voeding. De afstandssensor krijgt 12 Volt voeding. Een step-down module reduceert de 12V spanning naar 3.3V voor de microcontroller en de rest van de aangesloten randapparatuur.

* Ingangsspanning: 12V DC.
* Ingangsaansluiting: DC-barrel jack.
* Uitgangsspanning: 3.3V gereguleerd.
* Maximale stroomverbruik: ~150 mA (gemeten piek in bedrijfstoestand)

**Microcontroller – Sensor (afstandssensor)**

Specificaties

De afstandssensor (Sick WL250-2P2431) betreft een digitale sensor die een object detecteert binnen een bereik van 1 cm en 11 meter dat een digitaal hoog signaal aanbiedt op de GPIO-ingang.

* Interface: Digital input + interrupt.
* Signaaltype: digitale pulsen bij detectie (1 = object aanwezig).
* Signaalspanning: 12V met spanningsdeler en zenerdiode gereduceerd naar 3.3 Volt.
* Triggergevoeligheid: binnen 1 milliseconde.
* Werkspanning sensor: 12V.
* Uitlezing: via externe interrupt op GPIO pin3.
* Driverfuncties (zie figuur 6):
  + - SensorTrigger.begin(D3) – Initialiseert de sensor door GPIO-pin als input te configureren en de interrupt aan deze pin te koppelen.
    - SensorTrigger::handleInterrupt() – Wordt automatisch aangeroepen bij een interrupt. Voert softwarematige debounce uit en registreert een geldige trigger.
    - SensorTrigger::wasTriggered() – Wordt aangeroepen vanuit de hoofdloop. Controleert of er sinds de vorige aanroep een nieuwe detectie is geweest.
    - TransferPending() – Verplaatst gedetecteerde triggers vanuit een tijdelijke ISR-buffer naar een hoofdbuffer voor verwerking buiten de interruptcontext.
    - ProcessNext(SDManager& sd) – Verwerkt de oudste trigger en schrijft deze als tijdstempel naar een CSV-bestand via de SDManager.

Afbeelding met tekst, schermopname, Lettertype, diagram

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 6: Aansturing van de afstandssensor.

*“De diagrammen in de figuren 3 tot en met 7 zijn opgesteld op systeem- en module-niveau. Per driver is aangegeven welke functie beschikbaar zijn en hoe de microcontroller deze aanroept. Gedetailleerde implementatielogica (zoals loop() structuren) is buiten beschouwing gelaten om de systeemwerking overzichtelijk te houden.”*

**Microcontroller – Actuator (OLED-scherm)**

Specificaties

De gebruikersinterface bestaat uit een OLED-scherm (128X64, driver:SSD1306) dat via I2C communiceert met de microcontroller.

* Interface : I2C.
* Adres : 0x3C.
* Snelheid : 100 kHz (standard mode).
* Spanning : 3.3V.
* Schermformaat : 128 x 64 pixels.
* Besturingschip : SSD1306
* Geheugenstructuur : 8 pagina’s van elk 128 bytes (1 KB totaal)
* Bibliotheek : Adafruit\_SSD1306.h (of U8g2lib.h)
* Driverfuncties (zie diagram figuur 7):
  + displayManager.begin() – Initialiseert het OLED-scherm en stelt de I2C-verbinding in.
  + displayManager.setState(DisplayState::Menu) – Zet de weergavemodus op ‘Menu’, zodat het hoofdmenu getoond wordt.
  + displayManager.update(time zone, rtcNow) – Ververst de informatie op het scherm op basis van de actuele tijd en ingestelde tijdzone.
  + displayManager.showIntro() / showMenu() – Toont het opstartlogo of het hoofdmenu afhankelijk van de systemstatus.
  + displayManager.showMessage(”logging gestart”) – Geeft een tijdelijke statusmelding weer aan de gebruiker.
  + displayManager.addUserMessage(“SD-kaart geplaatst”) – Voegt een gebruikersmelding toe aan het informatiescherm.

Afbeelding met tekst, schermopname, Lettertype, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 7 Aansturing van het OLED-scherm via de DisplayManager-klasse.

**Microcontroller – Actuator (SD-kaartmodule)**

Specificaties

De SD-kaartmodule wordt gebruikt voor het opslaan van meetgegevens in CSV-formaat. De module maakt gebruik van SPI-communicatie.

* Interface : SPI.
* SPI-pinnen : MOSI, MISO, SCK, CS.
* Spanning : 3.3V.
* Vereiste SD-klasse : Class 10 of hoger (voor voldoende schrijfsnelheid).
* Bestandsformaat : CSV (Comma Separated Values)
* Besturingsbibliotheek : Arduino SD.h of aangepaste SDManager-klasse
* Driverfuncties (zie diagram figuur 8):
  + SDManager.begin() – Initialiseert de SPI-verbinding en controleert of de SD-kaart is geplaatst en retourneert als deze actie geslaagd is true terug.
  + SDManager.writeLine(csvData) – Schrijft een regel meetgegevens naar het actieve CSV-bestand op de SD-kaart en retourneert als deze actie geslaagd is true terug. De string csvData bevat een tijdstempel.
  + SDManager.fileExists(filename) – Controleert of een bepaald logbestand zich op de SD-kaart bevind en retourneert true terug als het bestand bestaat.

Afbeelding met tekst, schermopname, lijn, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 8: Aansturing van de SD-kaartmodule via de SDManager-klasse.

**Microcontroller – Actuator (RTC-module)**

Specificaties

De RTC-kaartmodule (Real-Time Clock) zorgt voor een stabiele en nauwkeurige tijdbron, geheel onafhankelijk van de systeemklok van de microcontroller. De tijd wordt onder andere gebruikt voor het tijdstempels van metingen op de SD-kaart.

* Type Module : DS3132
* Interface : I2C.
* Adres : 0x68.
* Nauwkeurigheid : ± 2 (ppm) Parts Per Million.
* Spanning : 3.3V.
* Back-up voeding : Ja, via knoopcel (CR2032)
* Gebruik : dag/maand/jaar uur:minuut:seconde
* Bibliotheek : Arduino RTClib.h of aangepaste RTCManager-klasse
* Driverfuncties (zie diagram figuur 9):
  + RTCManager.begin() – Initialiseert de RTC en controleert of deze correct functioneert en retourneert als deze actie geslaagd is true terug.
  + RTCManager.getNow() – Haalt de huidige datum en tijd op als DateTime-object (van RTClib).
  + RTCManager.setDateTime(day, month, year, hour, minute) – Stelt de datum en de tijd handmatig in.
  + RTCManager.getTimeString() – Retourneert een string met de huidige tijd in HH:MM:SS-formaat.
  + RTCManager.getDateString() – Retourneert een string met de huidige datum in DD-MM-YYYY-formaat.

De RTC wordt gebruikt om:

Afbeelding met tekst, schermopname, Lettertype, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 9: Aansturing van de RTC-module via de RTCManager-klasse.

**Microcontroller – Actuator (GPS-module)**

Specificaties

De GPS-module is momenteel nog niet actief en ook niet hardwarematig aangesloten. De module maakt gebruik van de UART-communicatie.

* Interface : UART (seriële communicatie).
* Baudrate : 9600.
* Spanning : 3.3V.
* Satellietnetwerk : GPS, GLONASS, GALILEO, BEIDOU, QZSS.
* NMEA-output : Ja, o.a. GPRMC en GPGGA.
* Datatypes : Positie (lat, lng) datum en tijd.
* Tijdzone : Instelbaar via constructorparameter.
* Bibliotheek : TintGPSPlus (TinyGPS++.h).
* Driverfuncties (zie diagram figuur 10):
  + GpsHandler::GpsHandler(rxPin, txPin, timeZone) – Constructor waarmee de RX- en TX-pinnen en tijdzone-offset worden ingesteld.
  + GpsHandler::begin()– Initialiseert de UART-verbinding met de GPS-module (via HardwareSerial).
  + GpsHandler::update()– Leest de seriële data uit en voedt de parser van TinyGPS++ (meestal aangeroepen in loop()).
  + GpsHandler::getGps– Retourneert een verwijzing naar het interne TinyGPSplus-object, waarmee functies als gps.location.lat() en gps.time.value() beschikbaar zijn.

Afbeelding met tekst, schermopname, lijn, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 10: Aansturing van de GPS-module via de GPSHandler-klasse.

**Toekomstig gebruik**

De GPS-module is op dit moment nog niet operationeel, maar wel klaar voor integratie. Mogelijke toepassingen:

* Tijdcorrectie op basis van satellietdata (als vervanging of aanvulling op de RTC).
* Locatielogging bij metingen.
* Snelheidsbepaling bij metingen op basis van satellietdata.

**Microcontroller – Drukknoppen**

Specificaties

De drukknoppen worden aangesloten op vier digitale input pinnen. Alle vier de drukknoppen zijn laag actief.

* Wanneer een drukknop niet ingedrukt is, is de ingangsspanning van de GPIO-pin logisch hoog (1) ≈ 3.3V.
* Wanneer een drukknop wel is ingedrukt, is de ingangsspaning logisch laag (0) ≈ GND.

De vier drukknoppen worden gebruikt voor menu-invoer via het OLED-scherm. De functionaliteit is als volgt verdeeld:

|  |  |  |
| --- | --- | --- |
| Drukknop | Functie binnen hoofdmenu | Functie binnen klokmenu |
| **1** | Startmeting (start) | Verhoogde waarde ( + ) |
| **2** | Klokmenu (klok) | Volgende veld ( > ) |
| **3** | GPS (GPS) | Bevestigen ( OK ) |
| **4** | Stopmeting (stop) | Verlagen waarde ( - ) |

* Tabel 1: Functie van de drukknoppen.

Er is een ButtonManager-klasse geïmplementeerd als softwaredriver, die verantwoordelijk is voor het configureren van de vier GPIO-pinnen als digitale input. En voor het lezen van de momentane status van elke afzonderlijke drukknop. De status van de knoppen worden gepold (periodiek uitgelezen), en niet interrupt-gestuurd. Dit houdt in dat iedere 100 milliseconden de vier drukknoppen worden gecontroleerd.

Driverfuncties (zie diagram figuur 11):

* + ButtonManager.begin() – Initialisatie van de inputpinnen (éénmalig in setup()).
  + ButtonManager.readButtons() – Periodieke polling van hoofdmenu drukknoppen.
  + ButtonManager.readSecondButtons() – Periodieke polling van klokmenu drukknoppen.

De functies readButtons() en readSecondButtons() maken gebruik van debouncing-logica. Hierbij wordt voor elke knop de stabiele toestand bepaald op basis van een ingesteld vertraging (20 ms). Interne arrays houden bij wat de vorige en huidige status van elke knop is. Op basis hiervan wordt per knop één bijbehorende actie (ButtonAction) teruggeven. Dit voorkomt dat een knop per ongeluk meerdere keren geteld wordt bij één druk.

Afbeelding met tekst, schermopname, lijn, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 11: Aansturing van de drukknoppen via de ButtonManager-klasse.

**Microcontroller**

De gekozen microcontroller is een Arduino Nano ESP32-S3, gebaseerd op de Espressif ESP32-S3-chip. Deze microcontroller vormt het centrale rekenhart van het systeem en verzorgt zowel de communicatie met de sensoren als de verwerking van tijdstempels en opslag op SD-kaart.

Specificaties:

* Model: Arduino Nano ESP32-S3.
* Core: Dual-core Tensillica LX7, 240 Mhz.
* Werkspanning 3,3 V.
* Ingangsspaning: (Via VIN): 5V.
* RAM: 512 KB SRAM + 8 MB extern PSRAM.
* Flash: 8 MB.
* GPIO-pinnen: 23 beschikbaar.
* Communicatieprotocollen:
  + I2C (voor OLED-scherm en RTC)
  + UART (voor GPS-module)
  + SPI (Voor SD-kaartmodule)
* USB-communicatie via USB-C (data en voeding).
* Extra’s: hardwarematig ondersteuning voor interrupts en nauwkeurige timing.

Ingezette functionaliteit:

* GPIO: D3 t/m D9 gebruikt voor 4 drukknoppen en een interruptsignaal van een afstandssensor (Sick WL250-2P2431).
* I2C-bus: Communicatie met een SSD1306 OLED en DS3132 RTC.
* UART: Ontvangst van NMEA-berichten via GPS-module (AIR530Z).
* SPI: Communicatie met SD-kaartmodule (MISO,MOSI,SCK, CS).
* USB: Voor debuggen en/of data-overdracht met PC.

De microcontroller verwerkt inkomende pulsgebaseerde signalen van een sensor via hardware-interrupts, registreert tijdstempels, en stuurt deze naar een SD-kaart met behulp van een interne buffer. De ESP32-S3 wordt gekozen vanwege de hoge rekencapaciteit, ruime geheugenvoorziening, FREE RTOS en een dual core.

Afbeelding met tekst, diagram, schermopname, Parallel

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 12: Diagram van de interactie tussen de microcontroller en aangesloten componenten.

De pijlen geven methodes weer die door de hoofdapplicatie worden aangeroepen tijdens de werking van het systeem.

## Software

De software van de datalogger is gebaseerd op een event-driven architectuur. De microcontroller reageert op externe gebeurtenissen zoals drukknoppen en sensorinterrupts. Het systeem maakt gebruik van hardwarematige interrupts voor nauwkeurige registratie van sensordetecties, terwijl de overige onderdelen periodiek worden uitgelezen in de hoofdloop. Wanneer er een interrupt gedetecteerd wordt er een tijdstempel opgeslagen in een tijdelijke buffer.

Er is géén gebruik gemaakt van een RTOS. In plaats daarvan is gekozen voor een modulaire structuur waarbij elke component een eigen init()- en update()- functie heeft. Events (zoals drukknoppen of detecties) worden door specifieke handlers verwerkt.

Toestandsmachine

Het systeem werkt op basis van een eenvoudige toestandsmachine met de volgende hoofdmodi:

* “Startup: toont logo en systeeminformatie”
* “Menu: standby-toestand, wacht op knopbediening”
* “Logging: registreert detecties via interrupts, slaat deze op”
* “Samenvatting: toont aantal registraties na meting”
* “Klokinstellingen (optioneel): stelt RTC handmatig in via knoppen”

Onderstaande diagrammen tonen de softwarewerking op hoofdlijnen.

# 

# 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