

Embedded Systems Engineering

Embedded Datalogger voor de validatie van tijdwaarneming van verkeersregelinstallaties

Productrapport

Ontwerp en ontwikkeling van een systeem voor efficiënte validatie van tijdwaarneming door verkeersregelinstallaties



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

Auteur	: Maarten van Riel
Studentnr.	: 633992
Bedrijfsbegeleider	: Mark Bos
Afstudeerbegeleider	: Peter Bijl
Bedrijf	: Nationale politie
Periode	: februari 2025 / juni 2025
Versie	: 0.2

Revisies

Versie	Wanneer	Wie	Wat
0.1	10 feb	Maarten	Eerste initiële aanzet
0.1	10 maart	Maarten	Functioneel- en technisch ontwerp aangevuld
0.1	10 mei	Maarten	Productrapport afgerond
0.2	30 mei	Maarten	Aanvullingen na commentaar Mark Bos

Voorwoord

Voor u ligt het afstudeerverslag van mijn project, uitgevoerd in opdracht van de Nationale Politie. Met dit project rond ik mijn deeltijd Associate-Degree opleiding Embedded System Engineering aan de Hogeschool Arnhem en Nijmegen af. In dit project heb ik een prototype ontwikkeld van een datalogger die nauwkeurig tijdmetingen kan uitvoeren tijdens validatieproeven van verkeersregelinstallaties.

De aanleiding voor dit project was de behoefte aan een gebruiksvriendelijke en toekomstbestendig alternatief voor de huidige, verouderde datalogger die in de eenheid Zeeland – West – Brabant wordt gebruikt. Het gaf mij de unieke kans om mijn technische kennis en praktijkervaring te combineren in een concreet en maatschappelijk relevant product.

Tijdens dit traject heb ik veel geleerd over embedded systemen, projectplanning en het belang van iteratief werken. Tegelijkertijd werd ik geconfronteerd met uitdagingen zoals tijdsdruk, softwarebugs en het bewaken van de realistische doelen. Deze ervaringen hebben mij professioneel als persoonlijk doen groeien.

Ik wil mijn werkgever de Nationale Politie bedanken voor het faciliteren van dit project in tijd, budget en middelen. In het bijzonder dank ik mijn praktijkbegeleider, die mij regelmatig van waardevolle input voorzag, heeft geholpen met het assembleren van de LogiTrack MR1 en altijd voor mij klaar stond. Ook mijn studiebegeleider wil ik bedanken voor zijn advies en begeleiding, evenals mijn collega's van de Forensische Opsporing (eenheid Zeeland – West – Brabant) en de Digitale Opsporing (eenheid Oost – Brabant), die mij de ruimte en de gelegenheid hebben geboden om dit project volledig af te ronden.

Tot slot wil ik mijn gezinsleden bedanken voor hun geduld, begrip en steun gedurende deze intensieve periode.

Gorinchem – Maarten van Riel, mei 2025

Samenvatting

De ongevalsanalisten van de Forensische Opsporing van politie worden regelmatig geconfronteerd met complexe aanrijdingen op kruispunten die worden geregeld door verkeerslichten. Voor de analyse wordt gebruik gemaakt van loggegevens van detectielussen uit de verkeersregelinstallatie, onder andere om roodlichtnegatie of de gereden snelheid te berekenen. Doordat bijna alle installaties slechts in 10 Hz loggen, is de tijdresolutie onvoldoende voor betrouwbare snelheidsberekening.

Om de nauwkeurigheid van de loggegevens te valideren, maakt de politie gebruik van dataloggers die onafhankelijk meetgegevens vastleggen met een frequentie van 1 kHz. Deze dataloggers registreren het moment waarop een referentievoertuig een detectielus aanrijdt, aan de hand van reflectorpalen. Door deze detectiemomenten te vergelijken met de logtijden van de verkeersregelinstallatie, kan het afrondingsverschil worden vastgesteld.

Bij de Nationale Politie zijn momenteel meerdere dataloggers in gebruik. De datalogger van de eenheid Zeeland – West – Brabant is echter verouderd en gebruiksvriendelijk. Dit project beschrijft het ontwerp en de realisatie van een nieuw prototype: de LogiTrack MR1, ontwikkeld op basis van wensen van de eindgebruikers.

De LogiTrack MR1 is opgebouwd rond een Arduino Nano ESP32-S3 en maakt gebruik van een Sick WL250-2P2431 afstandssensor. De logger registreert detecties nauwkeurig met een DS3231 RTC en slaat tijdstempels op in CSV-formaat via een SD-kaartmodule. Een OLED-display biedt gebruikersfeedback, en bediening verloopt via vier fysieke drukknoppen. Vanwege prioritering is de GPS-module buiten de scope van deze versie gelaten.

De LogiTrack is getest in combinatie met de Mov1Log BM22 en een DAQ-systeem. Door een softwarefout kon de data van de Mov1Log niet worden geduid. Daarnaast werd bij de LogiTrack een fout in de seondenweergave vastgesteld: deze liep incidenteel één seconde voor of achter. Ondanks deze bug bleef de millisecondenregistratie nauwkeurig. Uit 73 metingen bleek de gemiddelde afwijking ten opzichte van de DAQ minder dan 9 ms bedroeg, met een maximale afwijking van 26 ms. De LogiTrack MR1 registreerde 68 van de 73 metingen eerder dan de DAQ.

Daarmee zijn de hoofddoelen van dit project behaald: het realiseren van een werkend prototype van een datalogger die nauwkeurige tijdmetingen uitvoert met een gebruiksvriendelijk ontwerp. De LogiTrack MR1 biedt een solide basis voor verdere optimalisatie en validatie, en brengt het team Forensische Opsporing een stap dichter bij een vervanging van hun verouderde meetsysteem.

Inhoudsopgave

Revisies.....	2
Voorwoord.....	3
Samenvatting	4
Inhoudsopgave.....	5
1 Inleiding.....	7
1.1 Aanleiding	7
1.2 Doelstelling	7
1.3 Structuur van het rapport	8
2 Probleemanalyse & Ontwerpkeuze.....	9
2.1 Opdrachtomschrijving	9
2.2 Kernprobleem	9
2.3 Analyse van mogelijke oplossingen.....	9
2.4 Gekozen oplossing en onderbouwing	9
3 Functioneel ontwerp	10
3.1 Functionele specificaties	10
3.2 Functionele eisen.....	10
3.3 Technische eisen.....	13
3.4 User interface	14
4 Technisch ontwerp	16
4.1 Architectuur.....	16
4.2 Interfaces	17
4.2.1 Voedingsspanning	17
4.2.2 Microcontroller – Sensor (afstandssensor).....	18
4.2.3 Microcontroller – Actuator (OLED-scherm).....	19
4.2.4 Microcontroller – Actuator (SD-kaartmodule)	20
4.2.5 Microcontroller – Actuator (RTC-module).....	21
4.2.6 Microcontroller – Actuator (GPS-module).....	22
4.2.7 Microcontroller – Drukknoppen	23
4.2.8 Microcontroller – Arduino Nano ESP32-S3	24
4.3 Software.....	25
5 Realisatie	26
5.1 Hardware	26
5.2 Pinout Overzicht.....	26
5.3 PCB	28

5.4	Software.....	30
5.4.2	Interruptbuffering.....	31
5.4.3	Verwerking van de pending buffer.....	31
5.4.4	RTC-integratie (DS3132 via RTCManager).....	31
6	Testen en evaluatie LogiTrack MR1	32
6.1	Real-life loggingtest metritten op circuit.....	32
6.2	Acceptatietesten	35
7	Uren- en kostenoverzicht.....	36
7.1	Urenbesteding.....	36
7.2	Kostenoverzicht	37
8	Conclusies en aanbevelingen.....	38
8.1	Reflectie op projectdoelen.....	38
8.2	Resultaten en wat is gerealiseerd	38
8.3	Wat is (nog) niet gerealiseerd	38
8.4	Aanbevelingen en uitbreidingen	39
9	Verwijzingen Producdocument	40
	Bijlage A Plan van Aanpak	41
	Bijlage B Reflectieverslag	56
	Bijlage C Analyse, Onderzoek en datasheets.....	59
	Bijlage C1 Datasheet Sick WL250-2P2431.....	78
	Bijlage C2 Datasheet OLED-Scherm.....	85
	Bijlage C3 Datasheet SD-kaartmodule	88
	Bijlage C4 Datasheet RTC DS3231	89
	Bijlage C6 Datasheet Tactile Momentary Push Button White.....	109
	Bijlage C7 Datasheet Arduino Nano ESP32-S3.....	110
	Bijlage C8 Datasheet GROVE – GPS (AIR530) Seeed Studio	133
	Bijlage C9 Verwijzingen Onderzoek & analyse	138
	Bijlage D Elektrisch schema	140
	Bijlage D1 Pinout Arduino Nano ESP32-S3.....	141
	Bijlage E Plan van aanpak testen prototype LogiTrack MR1.....	142
	Bijlage F Testuitslagen	144
	Bijlage G Acceptatietesten	151
	Bijlage H Handleiding LogiTrack MR1	155

1 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.

1.1 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 metingstabbel 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.

1.2 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 meetstelsel;
- 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.

1.3 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 worden de uren en kosten berekend.
- In hoofdstuk 8 wordt afgesloten met het eindresultaat en enkele aanbevelingen. Er wordt teruggekeken naar het project en daarmee wordt er een conclusie beschreven.
- In hoofdstuk 9 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.

2 Probleemanalyse & Ontwerpkeuze

2.1 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.

2.2 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.

2.3 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.

2.4 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 (zie [bijlage C](#) voor de datasheet);.

Belangrijke componenten:

- DS3231 RTC - voor betrouwbare tijdwaarneming (zie [bijlage C4](#) voor de datasheet);
- SD-kaartmodule - voor lokale opslag (zie [bijlage C3](#) voor de datasheet);
- Seeed Studio Grove – Air530Z - voor de plaatsbepaling en aanvullende tijdsbron (zie [bijlage C8](#) voor de datasheet);
- OLED-scherm + knoppen - voor eenvoudige gebruikersinteractie (zie respectievelijk [bijlage C2](#) en [bijlage C6](#) voor de datasheets);
- Sick WL250-2P2431 – voor detectie van de reflectorpalen (zie [bijlage C1](#) voor de datasheet).

Zie [bijlage C](#) voor het volledige componentenonderzoek en de bijbehorende datasheets.

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

3 Functioneel ontwerp

3.1 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.

3.2 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	Moet een reflector kunnen detecteren die een detectielusverandering simuleert.

#	Prioriteit (MoSCoW)	Specificatie	Details	Opmerkingen
			binnen 1 ms een meetwaarde registreren.	
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 druknop "START" wordt het frame gestart waarbij de datalogger tijdstempels registreert na detecties van de afstandssensor. Met het	Zelf moment bepalen wanneer de datalogger reageert op de sensor.

#	Prioriteit (MoSCoW)	Specificatie	Details	Opmerkingen
			bedienen van de STOP-druknop wordt het frame gesloten en de detecties opgeslagen. Tevens wordt er één rit opgeteld.	
F1.13	Must (M)	Lokale tijd synchroniseren	De druknop "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 druknop "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	Won't (W)	Ai-analyse	Het systeem zal geen automatische snelheidsberekening of patroonherkenning implementeren.	Goede toevoeging om efficiënt en effectief te werken.

3.3 Technische eisen

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

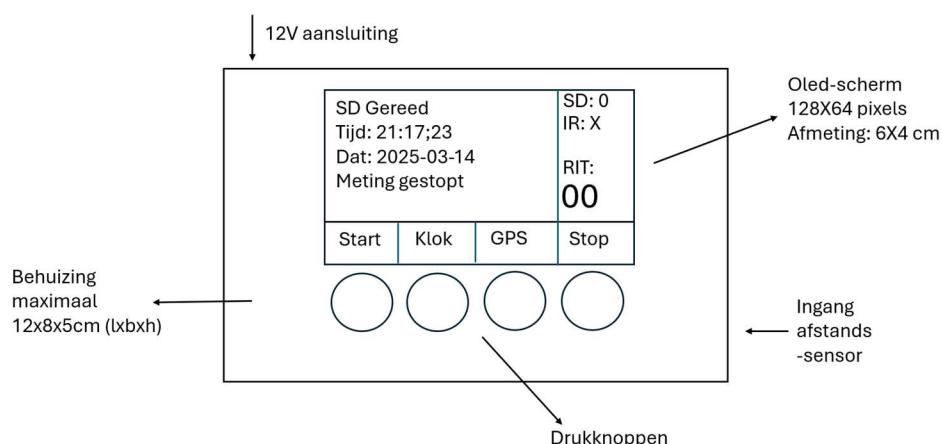
#	Prioriteit (MoSCoW)	Omschrijving
T1	Must (M)	De datalogger moet worden gebaseerd op de Arduino Nano ESP32-S3 microcontroller.
T2	Must (M)	De hardware wordt ontworpen als een uitbreidingsmodule (shield) voor de Arduino Nano ESP32-S3.
T3	Must (M)	De firmware wordt geschreven in C++ .
T4	Must (M)	Het systeem moet via UART, SPI en I ₂ C 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-signalen.
T14	Won't (W)	Een 3-assige gyroscoop/versnellingsmeter, om de acceleratie van de datalogger in x, y en z richting te meten.

3.4 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" – Stop de huidige meting.



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

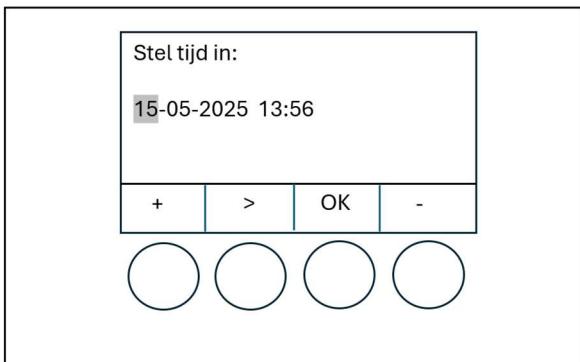
Wanneer een object wordt gedetecteerd (via de afstandssensor):

- De visuele indicator op het scherm geeft de detectie weer met een "X".

Wanneer de gebruiker druknop "Klok" bedient (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;

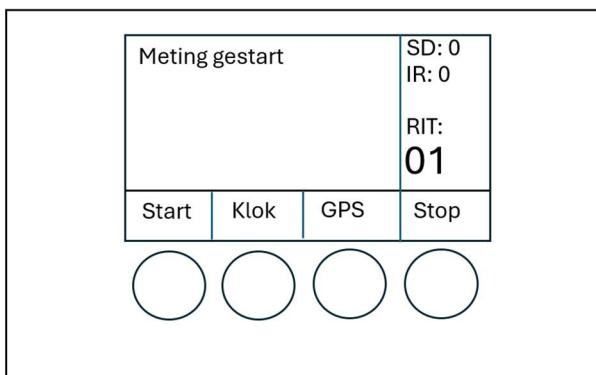
- Met de drukknop “>” kan het volgende karakter worden ingesteld;
- Met de drukknop “OK” wordt de waarde opgeslagen, het menu beëindigd en hoofdmenu geopend.



Figuur 2: Userinterface van klokmenu.

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

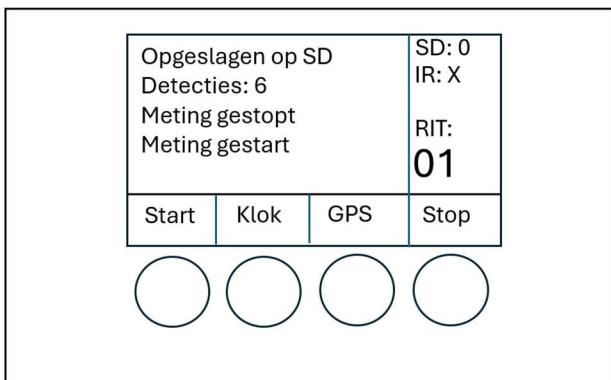
- Een nieuwe meting wordt gestart.
- De teller “rit” wordt met één opgehoogd;
- De userinfo wordt leeg gemaakt;
- Bovenaan de userinfo verschijnt “Meting gestart”.



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;



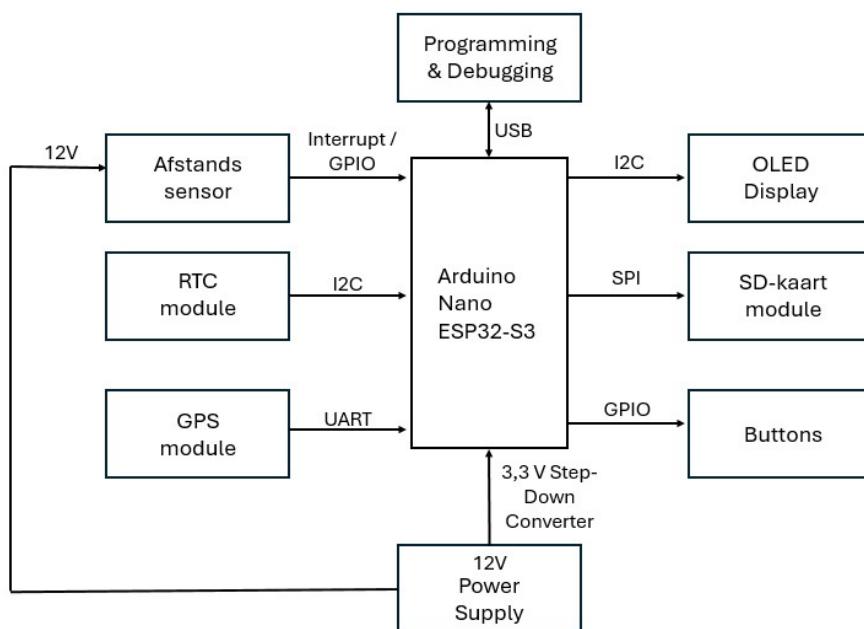
Figuur 4: Userinterface na beëindigen meting.

4 Technisch ontwerp

4.1 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.



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 verwerkingsseenheid:**
 - 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.

4.2 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 verwijzen worden naar de bijbehorende softwaredrivers.

4.2.1 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.

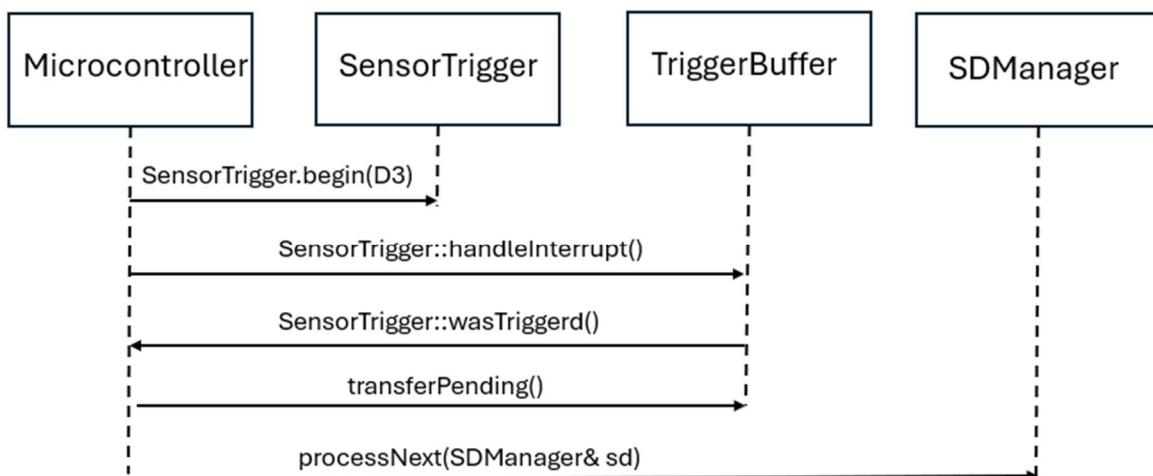
- Ingangsspanning: 12V DC.
- Ingangsaansluiting: DC-barrel jack.
- Uitgangsspanning: 3.3V gereguleerd.
- Maximale stroomverbruik: ~150 mA (gemeten piek in bedrijfstoestand)

4.2.2 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. (Zie bijlage C1 voor de datasheet)

- 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.



Figuur 6: Aansturing van de afstandssensor.

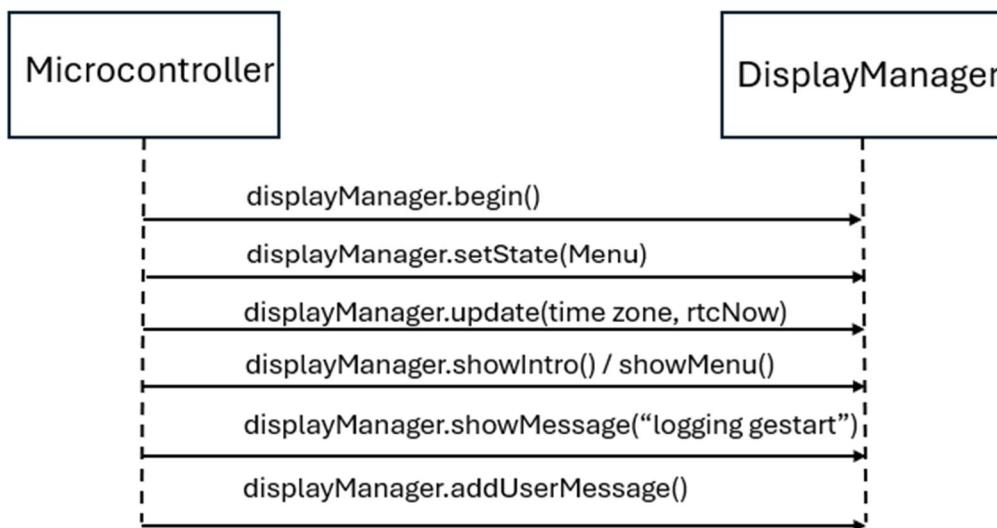
“De diagrammen in de figuren 6 tot en met 12 zijn opgesteld op systeem- en module-niveau. Per driver is aangegeven welke functies beschikbaar zijn en hoe de microcontroller deze aanroeft. Gedetailleerde implementatielogica (zoals loop() structuren) is buiten beschouwing gelaten om de systeemwerking overzichtelijk te houden.”

4.2.3 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.



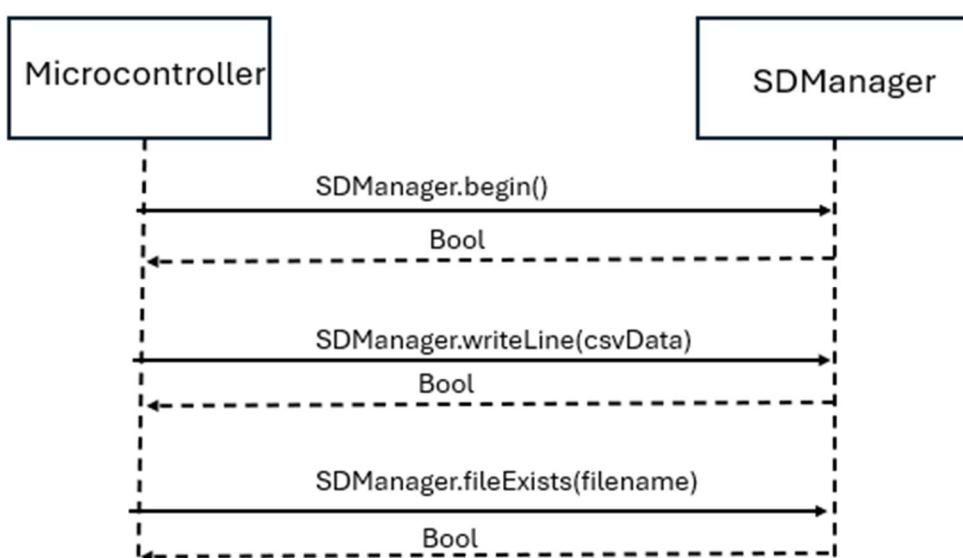
Figuur 7 Aansturing van het OLED-scherm via de DisplayManager-kLASSE.

4.2.4 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 bevindt en retourneert true terug als het bestand bestaat.



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

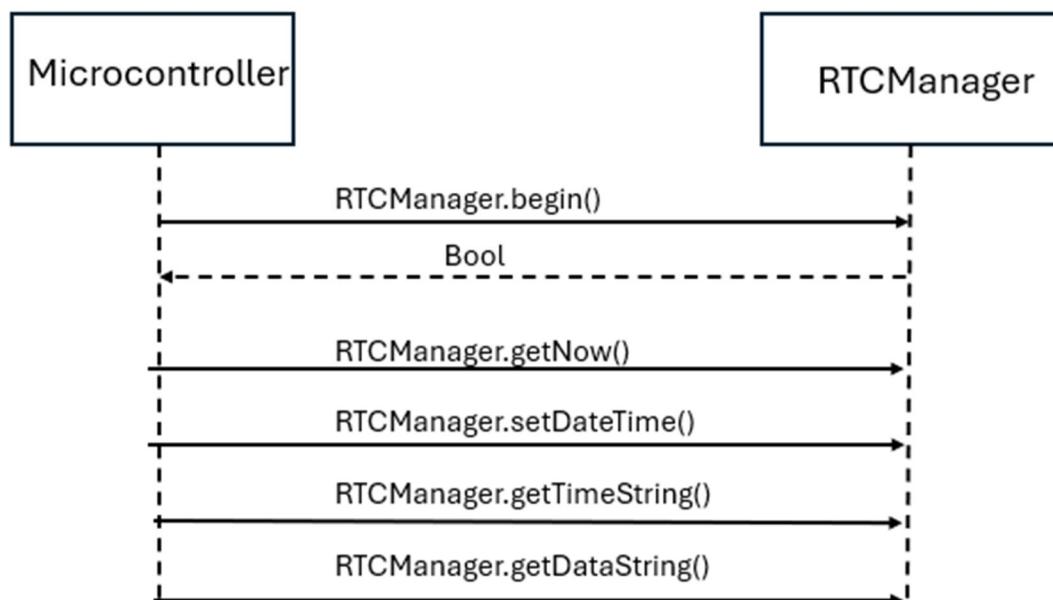
4.2.5 Microcontroller – Actuator (RTC-module)

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:



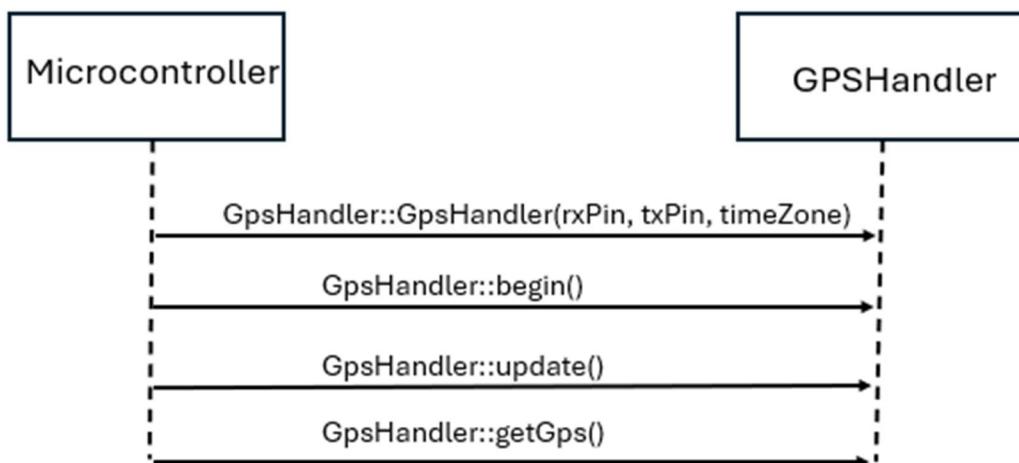
Figuur 9: Aansturing van de RTC-module via de RTCManager-klasse.

4.2.6 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, lon) datum en tijd.
 - Tijdzone : Instelbaar via constructorparameter.
 - Bibliotheek : TinyGPSPlus (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.



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.

4.2.7 Microcontroller – Drukknoppen

Specificaties

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

- Wanneer een druknop niet ingedrukt is, is de ingangsspanning van de GPIO-pin logisch hoog (1) \approx 3.3V.
- Wanneer een druknop wel is ingedrukt, is de ingangsspanning logisch laag (0) \approx 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)	Verhogen waarde (+)
2	Klokmenu (klok)	Volgende veld (>)
3	GPS (GPS)	Bevestigen (OK)
4	Stopmeting (stop)	Verlagen waarde (-)

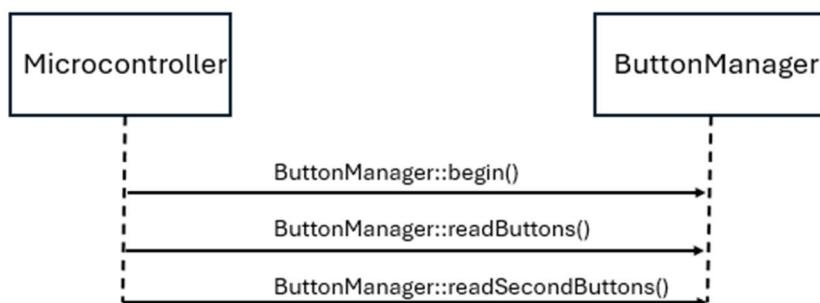
Tabel 1: Functie van de drukknoppen.

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 knopen 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) teruggegeven. Dit voorkomt dat een knop per ongeluk meerdere keren geteld wordt bij één druk.



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

4.2.8 Microcontroller – Arduino Nano ESP32-S3

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.

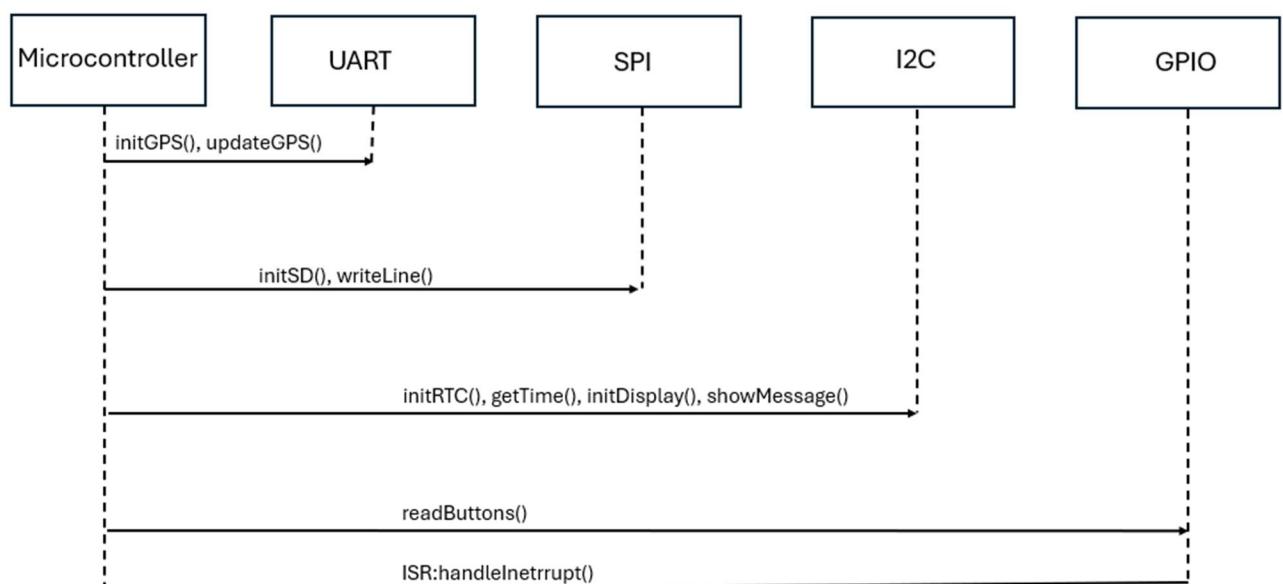
Specificaties:

- Model: Arduino Nano ESP32-S3.
- Core: Dual-core Tensilica LX7, 240 Mhz.
- Werkspanning 3,3 V.
- Ingangsspanning: (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.



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.

4.3 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 (de TriggerBuffer), waarna het event in de hoofdloop verder verwerkt wordt.

Er is bewust 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 sensorinterrupts) worden door specifieke handlers verwerkt.

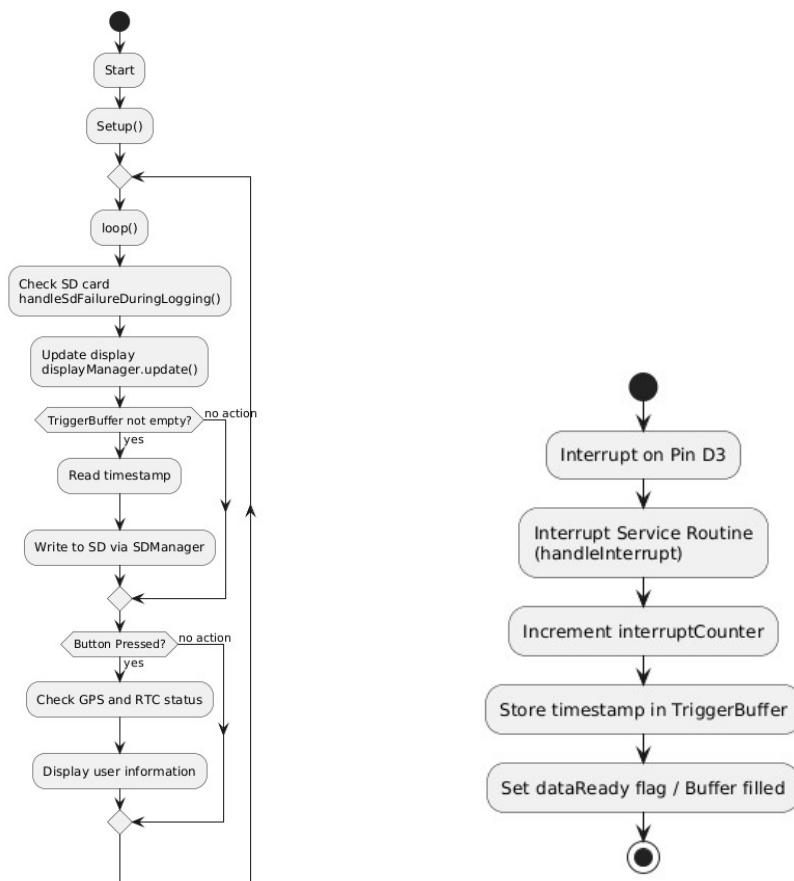
Toestandsmachine

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

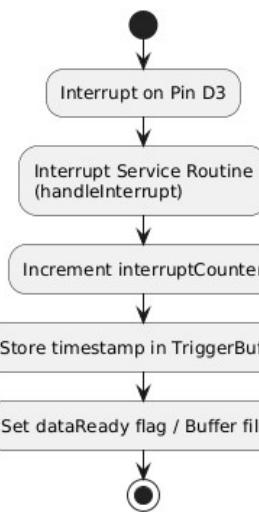
- “Startup: toont logo en systeeminformatie”
- “Menu: standby-toestand, wacht op knopbediening”
- “Logging: registreert detecties via interrupts, slaat deze op”
- “Samenvatting: toont aantal registraties na meting”
- “Klokinstellingen: stelt RTC handmatig in via knoppen”

Onderstaande diagrammen tonen de softwarewerking op hoofdlijnen.

- Figuur 13 toont de main-loop met eventverwerking en SD-opslag.
- Figuur 13a toont de interrupt-logica, waarbij een detectie leidt tot bufferen van de timestamp.



Figuur 13: Flowchart van de main-loop.



Figuur 13a: Flowchart van de interrupt.

5 Realisatie

5.1 Hardware

De hardware van de datalogger wordt beschreven aan de hand van het pinout overzicht en het elektrische schema.

5.2 Pinout Overzicht

Omdat in dit project gebruik gemaakt wordt van de Arduino Nano ESP32-S3, is het aantal beschikbare I/O afhankelijk van de fysieke pinnen op de headers van de microcontroller. Bij het opstellen van het pinout-overzicht is uitgegaan van de officiële documentatie van Espressif (ESP32-S3 datasheet en pinout).

De tabel in [bijlage D](#) en [bijlage D1](#) geven een overzicht weer welke pinnen welke functies vervullen binnen het systeem. Hierin is opgenomen welke pinnen worden gebruikt voor digitale en analoge I/O, communicatieprotocollen (zoals I²C, UART en SPI), en welke externe modules (zoals het OLED-scherm, SD-kaartlezer, GPS-module en afstandssensor) aan welke pin zijn gekoppeld.

5.2.1 Elektrisch schema

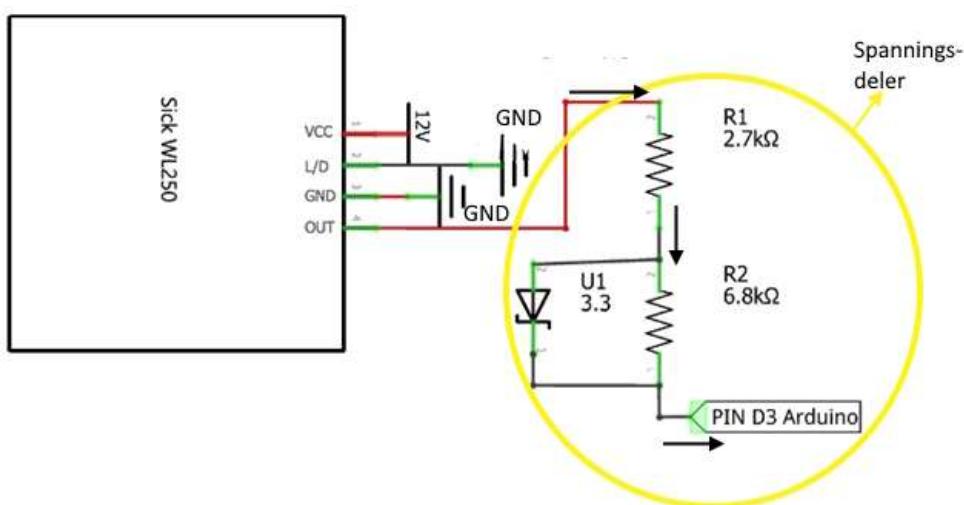
Het volledige elektrische schema van de datalogger is opgenomen als [bijlage D](#). In deze paragraaf volgt een toelichting op de meest belangrijke deelschema's.

5.2.2 Afstandssensor

Voor nauwkeurige detectie van objecten maakt dit project gebruik van de SICK WL250-2P2431, een compacte retroreflectieve fotocel met een PNP-schakeluitgang en een licht/donker-schakelmodus. De sensor detecteert objecten door middel van zichtbaar rood licht en vereist reflecterend materiaal (bijvoorbeeld de meegeleverde reflector P250) om te functioneren.

De sensor wordt aangesloten via een M12 4-pins connector, met de volgende pinconfiguratie (zie figuur 14):

- Pin 1 (BN – bruin): 10-30 V DC voedingsspanning
- Pin 2 (WH – wit): Schakelmodusselectie (naar GND voor detectie bij object)
- Pin 3 (BU – blauw): 0 V Massa (GND)
- Pin 4 (BK - zwart): PNP schakeluitgang naar Pin D3 Arduino Nano Esp32-S3



Figuur 14: Aansluitschema afstandssensor met spanningsdeler en zenerbeveiliging voor 3.3V GPIO-ingang.

Deze sensor werkt op een voeding van 12 V DC en is voorzien van een PNP-schakeling. Wanneer een object wordt gedetecteerd, schakelt de uitgang naar hoog (12 V), en in rusttoestand is de uitgang laag (0 V).

Omdat de GPIO-pinnen van de Arduino ESP32-S3 werken met een logicaniveau van 3,3 V, mag de uitgang van de sensor niet direct worden aangesloten. Om de 12 V van de sensoruitgang veilig terug te brengen naar circa 3,3 V, zijn een spanningsdeler en een zenerdiode (zie gele inzet figuur 14) opgenomen in de schakeling, bestaand uit:

- R1 = weerstand van 2,7 kΩ
- R2 = weerstand van 6,8 kΩ
- U1 = zenerdiode van 3,3 V (parallel over R2)

De spanningsdeler reduceert de 12 V van de sensor tot ongeveer:

$$V_{uit} = 12V \times \frac{6,8}{2,7 + 6,8} \approx 3,29V$$

Deze spanning wordt over R2 gezet (zie figuur 13). Parallel aan R2 is een zenerdiode (U1) van 3,3 V geplaatst, met kathode richting de GPIO. Deze diode beperkt de spanning tot maximaal 3,3 V, bij spanningspieken zoals variaties in voeding.

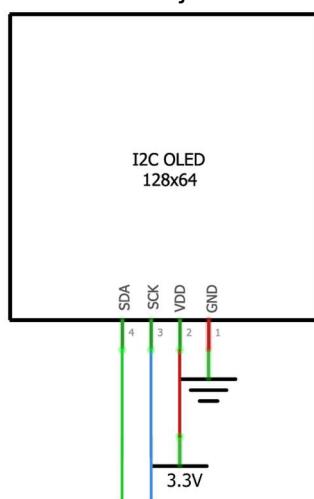
De begrenende spanning op het knooppunt tussen R2 en de zenerdiode wordt verbonden met GPIO D3 van de Arduino Nano ESP32-S3. Bij een hoog signaal van de sensor (detectie) zal de ingang circa 3,3 V lezen (logisch '1'), en bij geen detectie 0 V (logisch '0').

Opgemerkt dient te worden dat een optocoupler een beter alternatief is en zal in een volgend model worden toegepast.

5.2.3 OLED-scherm

Voor het weergeven van systeemstatus, menuschermen en meetgegevens wordt gebruik gemaakt van een 128x64 pixels OLED-scherm gebaseerd op de SSD1306-controller. Het display gebruikt I²C-interface en bevat een 4-pins male header met de volgende aansluitingen (zie figuur 15):

- VCC – Voedingsspanning (3,3 V)
- GND – Massa
- SCL - I²C-kloklijn
- SDA - I²C-datalijn



Figuur 15: Het aansluitschema van het OLED-scherm.

De module is aangesloten op de Arduino Nano ESP32 via de standaard I²C pinnen D6 (SCL) en D5 (SDA).

De module bevat reeds interne pull-up weerstanden op zowel de SCL- als SDA-lijn, waardoor er in dit ontwerp geen externe pull-up weerstanden zijn toegevoegd. De voedingsspanning wordt direct via de ESP32 geleverd (3,3 V), omdat de module compatible is met 3,3 V logica.

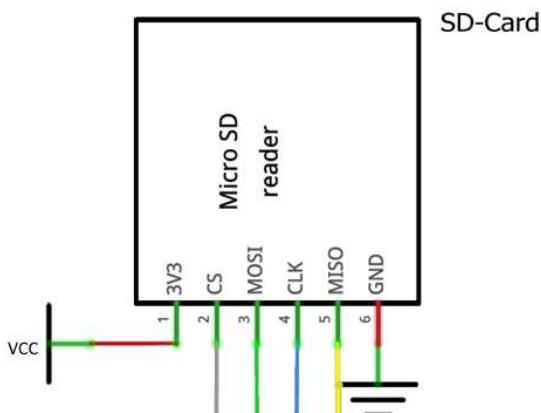
Het display wordt in de firmware aangestuurd met behulp van de Adafruit SSD1306-bibliotheek in combinatie met Adafruit GFX. Deze bibliotheken voorzien in tekenen van tekst, grafieken, menu's en gebruikersinformatie op het scherm.

5.2.4 SD-kaartmodule

Voor het opslaan van meetdata en timestamps wordt gebruik gemaakt van een SD-kaartmodule gebaseerd op een SPI-interface. De module maakt het mogelijk loggegevens in CSV-formaat weg te schrijven op een verwisselbare geheugenkaart.

De module bevat een 6-pins aansluiting met de volgende signalen (zie figuur 16):

- 3V3 – voedingsspanning VCC (3,3 V)
- CS (Chip Select) – GPIO D4 (Selecteert het SD-apparaat op de SPI_bus)
- MOSI (Master Out Slave In) – GPIO 11 (data van microcontroller naar SD)
- CLK (Clock / SCK) – GPIO D13 (kloksignaal)
- MISO – (Master In Slave Out) – GPIO D12 (data van SD naar microcontroller)
- GND - Massa



Figuur 16: Aansluitschema SD-kaartmodule (SPI).

De communicatie verloopt via de SPI-bus, waarbij het SD.begin(D4) commando in de code gebruikt wordt om de kaart te initialiseren. Bij een foutmelding (bijvoorbeeld kaart niet aanwezig) wordt via het OLED-scherm een foutstatus weergegeven.

5.3 PCB

Voor de PCB is geen eigen ontwerptekening opgenomen, omdat het ontwerpen en assembleren van een printplaat buiten de scope van deze opdracht bevond viel, mede vanwege een gebrek aan kennis en ervaring op dit gebied. Daarom zijn deze werkzaamheden uitbesteed.

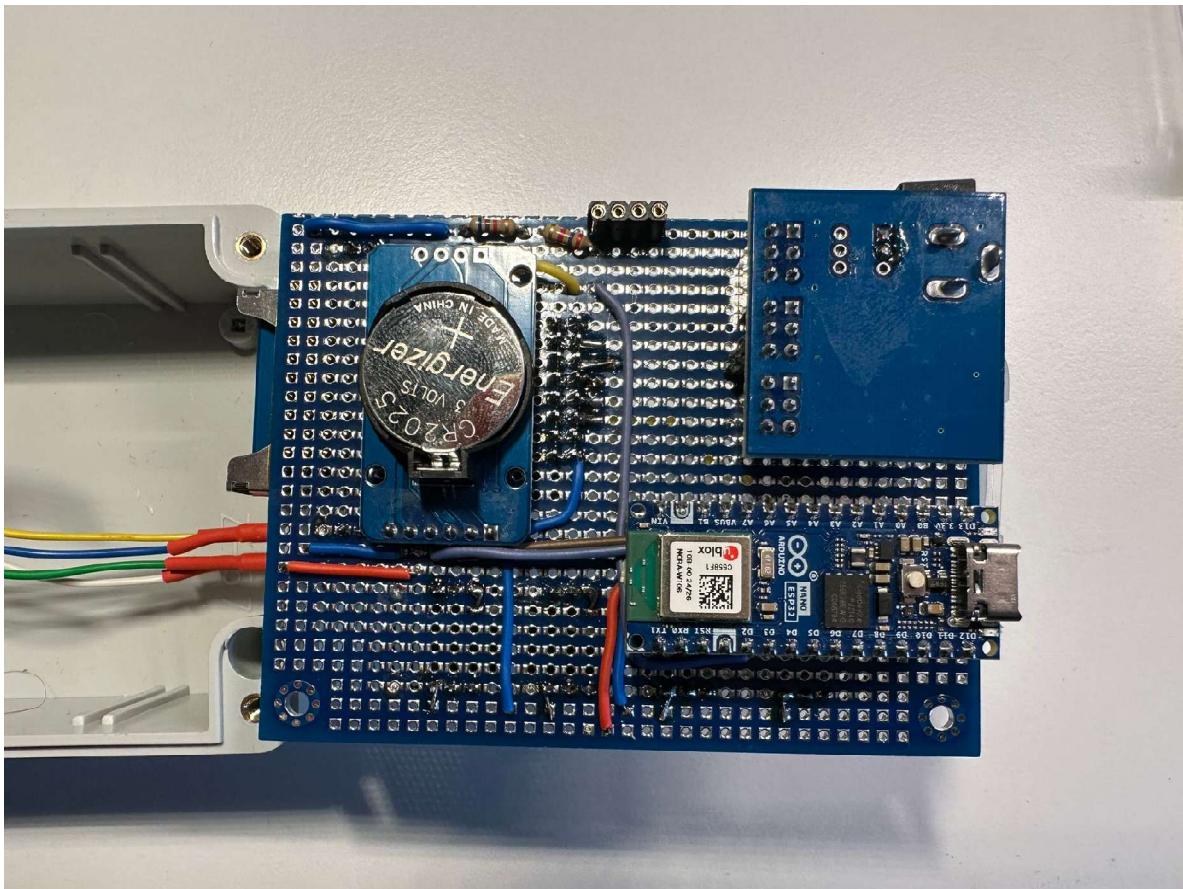
De modules - waaronder de ESP32, de SD-kaartlezer, het OLED-scherm en de externe RTC - zijn grotendeels door een externe partij op de printplaat gesoldeerd. Zelf heb ik de benodigde aansluitingen en de pinout bepaald. In de figuren 17, 18 en 19 zijn de fotoprints van het resultaat van de PCB-assemblage weergegeven.



Figuur 17: Bovenaanzicht printplaat



Figuur 18: Bovenaanzicht zonder OLED-display.



Figuur 19: Onderzijde printplaat.

5.4 Software

De volledige gerealiseerde software is als zip-bestand bij het digitale productdocument gevoegd. De bijbehorende broncode is tevens voorzien van Doxygen-documentatie, die gegenereerd is met behulp van [DoxiWizard](#).

Daarnaast is de volledige broncode beschikbaar gesteld via de interne Git-repository van het Ministerie van Justitie en Veiligheid, zodat deze ook in teamverband of voor doorontwikkeling geraadpleegd kan worden.

In deze paragraaf worden de belangrijkste onderdelen toegelicht aan de hand van de code snippets.

5.4.1 Arduino IDE

Voor de ontwikkeling van de software is gebruikgemaakt van Arduino IDE 2.3.5 (Arduino, 2025), in combinatie met de Arduino Nano ESP32-S3 boarddefinities. De hoofdcode is geschreven in C++ en in meerdere modules (klassen) voor o.a. tijdregistratie, displaybeheer, triggerhandling en datalogging.

Het project is op gebouwd uit meerdere .cpp- en .h-bestanden en volgt een modulaire structuur (zie figuur 20 voor de projectstructuur). De gebruikte bibliotheken zijn:

- TinyGPS++ versie 1.0.3. voor het parsen van NMEA-data;
- Adafruit_SSD1306 versie 2.5.14 voor het OLED-scherm;
- RTClib versie 2.1.4 voor de DS3231 RTC-module;
- SD.h versie 1.3.0. voor bestandopslag op SD-kaart.

Alle gebruikte libraries zijn beschikbaar via de Arduino Library Manager of GitHub.

```
#include <TinyGPS++.h> // Lib for translating NMEA data
#include "DisplayManager.h" // Header-file for OLED
#include "GpsHandler.h" // GPS-module
#include "Utils.h" // Helpfunction (printing digits)
#include "ButtonManager.h" // Buttonpress
#include "SensorTrigger.h" // React on trigger interrupt sensor
#include "RTC.h"
#include "SDManager.h" // SD card
#include "TriggerBuffer.h"
#include <SPI.h>
#include <SD.h>
#include "SdFailureHandler.h"
```

Figuur 20: Projectstructuur en modules in Arduino IDE 2.3.5.

Voor het uploaden van de code naar de Arduino Nano ESP32-S3 is het belangrijk dat de microcontroller correct is aangesloten via een USB-C-kabel. In de Arduino IDE wordt het juiste boardtype geselecteerd via het dropdownmenu bovenin (in dit geval de Arduino Nano ESP32-S3). Vervolgens wordt via het knopje met de pijl naar rechts (Upload) de gecompliceerde code via USB naar de microcontroller geschreven.

Bij het eerste gebruik moet mogelijk het juiste poortnummer worden geselecteerd via het menu: Hulpmiddelen > Poort. Als de upload correct is uitgevoerd, verschijnt in de seriële monitor of via het OLED-scherm feedback van het systeem.

5.4.2 Interruptbuffering

Onderstaande code (zie figuur 21) wordt aangeroepen vanuit interruptservice-routine (ISR) wanneer de afstandssensor een object detecteert. De tijdstempel (RTC + milliseconden) wordt in een tijdelijke buffer geplaatst. Dit voorkomt dat de interrupt handler te veel tijd kost.

```
17 void TriggerBuffer::addFromISR(DateTime time, unsigned long ms) {  
18     if (isrIndex < MAX_ISR_BUFFER) {  
19         isrBuffer[isrIndex].time = time;  
20         isrBuffer[isrIndex].ms = ms;  
21         isrIndex++;  
22         pending = true; // Signal that data is ready to be transferred  
23     }  
24 }
```

Figuur 21: Snippet code van de TriggerBuffer.

5.4.3 Verwerking van de pending buffer

Deze code (zie figuur 22) wordt cyclisch aangeroepen in de hoofdloop van het programma. Zodra er tijdens een interrupt een nieuwe detectie is toegevoegd aan het interrupt-buffer, zorgt transferPending() ervoor dat deze naar de hoofdbuffer wordt overgezet. Vervolgens controleert hasPending() of er iets te verwerken is. Met processNext(sd) wordt de oudste detectie gelogd naar de SD-kaart.

```
93 triggerBuffer.transferPending();  
94 if (triggerBuffer.hasPending()) {  
95     triggerBuffer.processNext(sd);  
96 }  
97 }
```

Figuur 22: Snippet code van de verwerking van de pending buffer.

5.4.4 RTC-integratie (DS3132 via RTCManager)

Deze functie (zie figuur 23) wordt aangeroepen wanneer een meting of logging wordt gestart. De huidige datum en tijd worden vastgelegd via de DS3231 RTC (rtc.now()), en tegelijkertijd wordt de systeemtijd in milliseconden (millis()) opgeslagen. Dit maakt het mogelijk om nauwkeurig relatieve en absolute tijdstempels te berekenen tijdens en na de meetcyclus.

```
43 void RTCManager::start() {  
44     running = true;  
45     startTime = rtc.now(); // Capture real time at start  
46     startMillis = millis(); // Save system uptime at start  
47 }
```

Figuur 23: Snippet code van de RTC-integratie.

6 Testen en evaluatie LogiTrack MR1

Doele van de testen

Het doel van het testen is om vast te stellen of de LogiTrack MR1 correct functioneert als datalogger voor snelheidsafhankelijke detectie van reflectorpaaltjes, en hoe de prestaties zich verhouden tot bestaande systemen (Mov1Log BM22 en DAQ).

De LogiTrack MR1 is op twee verschillende manieren getest:

1. Praktijktest op het circuit van de politieacademie te Lelystad. Hierbij zijn de betrouwbaarheid en nauwkeurigheid van het systeem beoordeeld onder realistische omstandigheden.
2. Acceptatietesten in een laboratoriumomgeving. Deze testen waren uitsluitend bedoeld om te controleren of de functies van het systeem naar behoren werken. Ze geven geen uitspraak over de betrouwbaarheid of nauwkeurigheid van het systeem in praktijksituaties.

6.1 Real-life loggingtest met ritten op circuit

Doele:

Nagaan of de LogiTrack MR1 sensortriggers betrouwbaar detecteert en correct opslaat tijdens realistische rijsituaties, en hoe de tijdwaarnemingen zich verhoudt tot vergelijkbare systemen. (*zie voor de testopstelling en methode het plan van aanpak voor de testen dat als bijlage E in dit document is opgenomen*)

Resultaten:

- De software van de Mov1Log kon door een softwarefout geen meetgegevens produceren.
- In het algemeen registreerde de LogiTrack de meeste triggers correct, met tijdstempels die grotendeels overeenkwamen met het DAQ-systeem.
- Bij zes ritten traden enkele problemen op:
 - De secondenwaarde van de RTC DS3231 bleef soms gelijk, ondanks dat de milliseconden wel veranderden.
 - In twee tests werden twee van de vier detectiebordjes niet geregistreerd. Deze bordjes stonden op 1 meter afstand van elkaar.

Beoordeling:

Detectie en logging werken in normale omstandigheden conform de functionele eisen.

Het probleem met de seconde is mogelijk softwarematig op te lossen door alleen gebruik te maken van de RTC-timer van de Arduino in plaats van DateTime.

Evaluatie door stakeholders:

Het apparaat voldoet aan de eisen.

Er is een voorkeur uitgesproken voor tijdsregistratie op basis van tussentijden, in plaats van absolute tijdstempels in Real-Time, omdat de Real-Time data alsnog moet worden verrekend.

Workarounds:

De huidige implementatie registreert tijdstempels in Real-Time.

Aanbevolen aanpassing: registreer alleen tussentijden tussen detectiepalen, bijvoorbeeld op basis van systeemuptime.

Dit verbetert waarschijnlijk ook de consistentie van de tijdregistratie.

Vergelijking met DAQ:

Tijdens de testritten is de LogiTrack vergeleken met de DAQ-systeem (zie voor het complete overzicht van de meetdata [bijlage F](#)). De testritten waarbij de seconde waarneming niet klopte werd handmatig aangepast. Na analyse van **73 metingen** blijkt:

Aspect	Waarde
Gemiddelde Afwijking	9 ms
Grootste afwijking	26 ms
Totale tijd gemeten door LogiTrack	42,204 seconden
Totale tijd gemeten door DAQ	42,257 seconden
Totale afwijking (over 73 metingen)	51 ms

Tabel 2: Overzicht testresultaten.

Analyse vergelijking meetverschillen:

Tijdens de analyse werd, zoals benoemd, vastgesteld dat bij 11 metingen (verspreid over 6 ritten) de secondewaarde één seconde te hoog of te laag was. Deze foutieve waarden zijn handmatig gecorrigeerd voor ritten 2, 6, 8, 9, 11 en 19. Daarnaast zijn ritten 21 en 22 uitgesloten van analyse, omdat de LogiTrack MR1 bij deze ritten slechts twee van de vier reflecties detecteerde. Hierdoor kon niet worden vastgesteld welke reflectiepalen wel of niet werden gedetecteerd.

Het is belangrijk te vermelden dat deze metingen bewust waren opgezet om de grenzen van het systeem te testen. De proefopstelling betrof reflectorpalen met slechts één meter tussenruimte, waarbij de beginsnelheid van het voertuig 100 km/u betrof.

Het gemiddelde verschil tussen de DAQ en de LogiTrack MR1 kwam uit op 9 ms, met een maximaal verschil van 26 ms (gemeten bij rit 4). Het verschil werd berekend als:
[waarde LogiTrack MR1] – [waarde DAQ].

Opvallend is dat in 68 van de 73 metingen de LogiTrack MR1 een kortere tijd registreerde dan het DAQ-systeem (zie de grafiek van [bijlage F](#)).

Om de invloed van de geconstateerde tijdsafwijkingen (gemiddeld 9 ms, max 26 ms) te duiden, is berekend hoe deze afwijkingen zouden doorwerken in realistisch situaties met verschillende afstanden. Hiervoor zijn parameters uit een praktijkcasus gebruikt. Een voertuig passeerde drie detectielussen: de afstand tussen eerste en de tweede lus 30,07 meter, de derde lus lag 17,02 meter verder, en de totale afstand was 47,09 meter.

Aan de hand van de gemeten detectietijden is een indicatieve snelheid berekend (tweede kolom tabel 3). Vervolgens is bekeken welke snelheidsverschillen zouden ontstaan indien deze tijden met 9 ms of 26 ms zouden afwijken.

Afstand (m)	Tijd (s)	Snelheid (km/u)	Snelheid + 9ms	Snelheid - 9ms	Snelheid + 26 ms	Snelheid - 26 ms
17,02	0,5	122,38	120,38	124,8	116,4	129,2
30,07	0,8	135,32	133,81	136,9	131,1	139,9
47,09	1,3	130,4	129,51	131,3	127,8	133,1

Tabel 3: Invloed van tijdsafwijking op snelheid bij verschillende afstanden.

Tabel 3 laat zien hoe klein de tijdsafwijkingen van 9 ms of 26 ms in de praktijk kunnen leiden tot snelheidsverschillen van enkele km/h. Opgemerkt moet worden dat deze casus representatief is voor een vrij extreme situatie: door de hoge snelheid zijn de detectietijden relatief kort, waardoor zelfs minimale tijdsverschillen een relatief grote impact hebben op de brekende snelheid. In praktijksituaties met lagere snelheden of grotere afstanden tussen detectielussen zal het effect van dergelijke tijdsafwijkingen naar verwachting nog kleiner zijn.

Conclusie betrouwbaarheid

Op basis van de uitgevoerde testen kan geconcludeerd worden dat de LogiTrack MR1 in staat is om sensortriggers betrouwbaar te registreren en te loggen tijdens realistische rijsituaties. De gemeten afwijking ten opzichte van het DAQ-systeem is gering (gemiddeld 9 ms), en blijft daarmee zeer waarschijnlijk binnen de acceptabele toleranties voor dit type toepassing.

Hoewel formele tollerantieregels of richtlijnen hiervoor ontbreken, wijzen de resultaten op voldoende precisie. De waargenomen problemen met de secondesynchronisatie hebben geen invloed op de milliseconden nauwkeurigheid en kunnen softwarematig worden opgelost.

De forensische opsporing werkt in de praktijk met een bandbreedte, waarbij per meting zowel een minimale als maximale snelheid wordt aangehouden op basis van meetnauwkeurigheden. Tabel 3 toont aan dat tijdsafwijkingen van 9 ms of zelfs 26 ms in realistische situaties slechts leiden tot snelheidsverschillen van enkele kilometers per uur.

6.2 Acceptatietesten

In totaal zijn er vier testscenario's uitgevoerd (zie tabel 4). Elk scenario is bedoeld om één of meerdere functionele specificaties te toetsen op correcte werking. Per scenario kan de uitkomst van een functionele specificatie als volgt worden beoordeeld:

- Functionele specificatie is volledig gerealiseerd
- Functionele specificatie is deels gerealiseerd
- Functionele specificatie is niet gerealiseerd

In tabel 4 wordt het testresultaat per scenario in één oogopslag weergegeven.

De gedetailleerde beschrijving van ieder testscenario is opgenomen in [Bijlage G](#). Daarin staat vermeld welke voorbereidingen vereist zijn, welke stappen zijn uitgevoerd, en welke uitvoer het systeem per stap zou moeten geven. Komt de werkelijke uitvoer overeen met de verwachte uitvoer, dan wordt de test als succesvol aangemerkt. Eventuele afwijkingen of opmerkingen worden aan het eind van elk scenario vermeld.

Test-nummer	Testscenario 1	Testscenario 2	Testscenario 3	Testscenario 4
F1				
F2				
F3				
F4				

Tabel 4: Testscenario's van de acceptatietesten.

7 Uren- en kostenoverzicht

7.1 Urenbesteding

In deze paragraaf wordt een overzicht gegeven van de tijdsbesteding (zie tabel 5) tijdens het afstudeerproject LogiTrack MR1. De totale beschikbare studielast voor dit project bedroeg 420 uren, zoals vastgesteld in de afstudeergids Associate Degree-opleidingen Engineering (HAN, 2025).

De uren zijn verdeeld over de verschillende projectfasen, zoals weergegeven in de planning (zie hoofdstuk 7 van het plan van aanpak in [bijlage A](#)). Het overzicht geeft inzicht in de verhouding tussen analyse, ontwerp, realisatie, testen en rapportage.

Activiteit	Aantal uren	Toelichting
Plan van aanpak	50 uur	Opstellen projectplan, opzet van doel.
Analyseren	20 uur	Analyse van bestaande systemen (Mov1Log BM22, DAQ), gebruikersbehoeften
Onderzoek	40 uur	Oriëntatie op hardware, technieken, en datalogging.
Functioneel ontwerp	20 uur	Uitwerken gebruikerswensen en eisen naar systeemfunctionaliteit.
Technisch ontwerp	20 uur	Schemaschets, hardware selectie.
Prototype	50 uur	Opbouw, montage en eerste werkende versie.
Testopstellingen voorbereiden.	40 uur	Gereedmaken opstellingen voor lab- en praktijktest (circuit Lelystad)
Realiseren	80 uur	Softwareontwikkeling, debuggen, Doxygen documentatie.
Testen	40 uur	Acceptatietesten (laboratorium) en praktijktesten (circuit) en de analyse daarvan.
Rapportage	100 uur	Opstellen en afronden verlag.
Totaal	460 uur	40 uur meer dan vastgesteld.

Tabel 5: Urenoverzicht.

In werkelijkheid is er circa 460 uur besteed aan het project. De extra tijd is besteed aan debugging en de uitgebreide documentatie. Hierdoor is het eindproduct volledig geworden.

7.2 Kostenoverzicht

In deze paragraaf wordt een overzicht gegeven van de totale kosten. Deze kosten bestaan voornamelijk uit hardwarecomponenten voor de LogiTrack. In tabel 6 zijn de kosten gespecificeerd.

Component	Aantal	Prijs per stuk (€)	Totaal
Arduino Nano ESP32	1	20,50	20,50
SD-kaartmodule + SD-kaart	1	4,25	4,25
GPS-module (Seeed AIR530)	1	11	11
OLED-scherm (128x64 I ² C)	1	8	8
DS3231 RTC-module	1	4,50	4,50
SICK WLP250 + kabel	1	285,22	285,22
Drukknoppen + behuizing	4	0,95	3,80
Hardcase	1	12	12
Zuignap + statief (Manfrotto)	1	101	101
Overige (kabels, breadboard, printplaat)	1	25	25
Totaal			475,27

Tabel 6: Kostenoverzicht hardware LogiTrack MR1

De in tabel 5 weergegeven kosten zijn de netto materiaalkosten van het prototype van de LogiTrack MR1. Deze hebben uitsluitend betrekking op de aangeschafte hardwarecomponenten.

De ontwikkelkosten, zoals ontwerp, programmering, testen en documentatie, zijn buiten beschouwing gelaten. Deze bestonden uit arbeidsuren binnen het afstudeerproject en vormen in een eventuele doorontwikkeling bij de politie de voornaamste kostendrager.

Hiermee laat het overzicht zien dat een functionele datalogger met relatief beperkte materiaalkosten gerealiseerd kan worden.

8 Conclusies en aanbevelingen

8.1 Reflectie op projectdoelen

Het doel van dit project was het ontwikkelen van een robuuste, modulaire en uitbreidbare datalogger waarmee nauwkeurige tijdmetingen tussen sensorgebeurtenissen (zoals reflectorpalen) kunnen worden vastgelegd. Belangrijke functionele en technische eisen waren:

- Het gebruik van een externe klok (RTC) voor nauwkeurige tijdwaarneming;
- Betrouwbaar verwerken van sensorinterruptie;
- Opslaan van meetgegevens op een SD-kaart;
- Eenvoudige bediening via knoppen;
- Terugkoppeling aan de gebruiker via het OLED-scherm over de status van metingen en opslag.

Deze doelen zijn grotendeels gerealiseerd. Het systeem detecteert sensorpulsen betrouwbaar, verwerkt deze via een interruptbuffer en slaat tijdstempels op in een CSV-bestand. De bediening is intuitief via knoppen, en de gebruikersinterface op het OLED-scherm geeft duidelijke feedback. De opbouw in losse modules (o.a. DisplayManager, RTCManager, TriggerBuffer) maakt het project goed onderhoudbaar en uitbreidbaar.

8.2 Resultaten en wat is gerealiseerd

- Nauwkeurige tijdregistratie met behulp van de DS3231 RTC en systick-timer
- Real-Time interruptafhandeling zonder gemiste detecties dankzij buffering
- Betrouwbare opslag op SD-kaart met foutafhandeling
- Gebruiksvriendelijke gebruikersinterface op OLED-scherm
- Keuzemenu voor het instellen van de lokale tijd
- Modulaire opgebouwde code met eigen .cpp en .h-bestanden
- Succesvolle testmetingen met herhaalbare resultaten
- Handleiding voor de gebruiker (zie [bijlage H](#))

8.3 Wat is (nog) niet gerealiseerd

- Er is geen draadloze overdracht voor data gerealiseerd (bijvoorbeeld Wi-Fi of Bluetooth), hoewel de Esp32 dat wel ondersteund.
- De GPS-module is nog niet volledig geïntegreerd. De pinout is voorbereid en de driver is grotendeels geïmplementeerd, maar door tijdsgebrek en prioritering in overleg met de stakeholders is de module voor nu buiten de scoop gevallen.
- De meetfrequentie is aangepast naar 1 kHz in plaats van de eerder beoogde 10 kHz, omdat uit onderzoek is gebleken dat bestaande datalogger in de praktijk ook op 1 kHz werken.
- Er is nog geen actieve preventie tegen valse detecties door de afstandssensor. Dit kan in de toekomst hardwarematig worden opgelost door het gebruik van sensoren met instelbare detectieafstand.
- Het RTOS van de Arduino Nano ESP32-S3 is niet gebruikt. De gekozen event-driven architectuur met interrupts bleek voldoende voor de gewenste functionaliteit.
- Er is geen op maat gemaakte, waterdichte behuizing ontwikkeld. Door gebrek aan ervaring is gekozen voor een bestaande behuizing, maar voor een professionele toepassing is een custom, spatwaterdichte behuizing wenselijk.

8.4 Aanbevelingen en uitbreidingen

1. **Herontwerp van de PCB**

Een eigen printplaat (ontworpen met KiCad of EasyEDA) en het vervangen van modules door losse componenten kunnen het systeem compacter, goedkoper en professioneler maken.

2. **Ontwerpen van een waterbestendige behuizing**

Een op maat gemaakte, robuuste en spatwaterdichte behuizing verhoogt de duurzaamheid en maakte de datalogger geschikt voor gebruik in het veld.

3. **Gebruik van relatieve tijdstempels**

De software kan worden aangepast zodat alleen de tijdverschillen tussen detectiepalen worden gelogd. De RTC wordt dan enkel gebruikt voor het registreren van de starttijd als referentie.

4. **Integreren van 3-assige versnellingssensor**

Door een versnellingsopnemer toe te voegen kan de datalogger worden ingezet voor aanvullende toepassingen zoals remvertragingsmetingen of acceleratiemetingen.

5. **Volledige integratie van de GPS-module**

Met een GPS-module kan de nauwkeurigheid van de tijdwaarneming verder worden verhoogd (via PPS-signalen), kan de actuele rijsnelheid worden gelogd en kunnen locatiegegevens aan de meetdata worden toegevoegd.

6. **Toevoegen van een batterij**

Een oplaadbare batterij maakt de datalogger onafhankelijk van voertuigvoeding, verhoogt het gebruiksgemak en voorkomt spanningspieken die de datalogger kunnen beschadigen.

Slotopmerking:

De LogiTrack MR1 heeft bewezen dat het mogelijk is om met beperkte middelen en lage materiaalkosten een betrouwbare en uitbreidbare datalogger te ontwikkelen op basis van een ESP32. Uit de uitgevoerde testen blijkt dat het systeem sensortriggers registreert met een nauwkeurigheid van gemiddeld 9 milliseconden, wat acceptabel is voor het beoogde forensische gebruik.

Hoewel formele tolerantiekaders ontbreken, is de meetnauwkeurigheid voor Forensisch onderzoek veelbelovend. De bug in de secondenweergave beïnvloed de millisecondenregistratie niet en is softwarematig oplosbaar.

De huidige versie vormt daarmee een stabiele en functionele basis voor verdere doorontwikkeling. Aanbevolen wordt om het systeem na afronding van de softwarecorrecties te valideren door paralleltesten met bestaande dataloggers uit de praktijk.

9 Verwijzingen Producdocument

- Arduino. (2025). Arduino IDE. *Software Arduino IDE downloads*.
- 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>
- HAN. (2025). <https://han.onderwijsonline.nl/elearning/lesson/Vq32z9kD>. Opgehaald van <https://han.onderwijsonline.nl>.
- <https://www.logmore.com/post/what-is-data-logging>. (sd). Opgehaald van
<https://www.logmore.com/>: <https://www.logmore.com/post/what-is-data-logging#:~:text=Data%20logging%20is%20the%20process,regardless%20of%20the%20method%20used>.
- 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 Plan van Aanpak



Plan van Aanpak

Datalogger voor het valideren van tijdwaarneming
verkeersregelinstallatie

Student : Maarten van Riel
Versie : 0.1
Datum : 30 januari 2025

Eenheid Zeeland – West Brabant
Dienst Regionale Recherche
Afdeling Specialistische Ondersteuning
Team Forensische Opsporing

Postbus 8050
5004 GB Tilburg
Tel. +31 88 9635060

« waakzaam en dienstbaar »



Algemeen

Opdrachtgever en betrokkenen

Opdrachtgever	: Nationale Politie
Contactpersoon	: Mark Bos, (Mark.Bos@politie.nl)
Bezoekadres	: Hoofdstraat 54
Postcode / Plaats	: 3972 LB Driebergen-Rijsenburg
Telefoon	: 06 18144638
Opdrachtnemer	: Politie Zeeland West Brabant
Contactpersoon	: Maarten van Riel (Maarten.van.Riel@politie.nl)
Bezoekadres	: Ringbaan West 232
Postcode / Plaats	: 5038 KE Tilburg
Telefoon	: 088-9635060
Docentbegeleider	: Hogeschool Arnhem & Nijmegen (HAN)
Contact persoon	: Peter Bijl (Peter.Bijl@han.nl)
Bezoekadres	: Ruitenberglaan 29
Postcode	: 6826 CC, Arnhem (Engineering)
Telefoon	: 06 55208764

Mutatieoverzicht

Datum	Versie	Auteur	Mutaties
30 januari 2025	0.1	M. van Riel	Eerste versie
18 februari 2025	0.2	M. van Riel	Aanpassingen na commentaar M Bos

Tabel 1: Mutatieoverzicht.



Inhoud

ALGEMEEN.....	2
OPDRACHTGEVER EN BETROKKENEN	2
MUTATIEOVERZICHT	2
1 INLEIDING	4
1.1 BEKNOPTE BESCHRIJVING.....	4
1.2 AANLEIDING	4
1.3 DOEL DOCUMENT.....	4
2 PROBLEEMANALYSE EN OPDRACHTOMSCHRIJVING.....	6
2.1 PROBLEEMSTELLING	6
2.2 DOELSTELLING	7
2.3 EINDRESULTAAT	7
3 PROJECTACTIVITEITEN EN FASERING.....	8
3.1 OVERZICHT ACTIVITEITEN	8
4 KWALITEITSWAARBORGING EN METHODOLOGIE.....	9
4.1 KWALITEITSEISEN.....	9
4.2 WERKWIJZE EN METHODES	10
4.3 RISICOMANAGEMENT	10
5 PROJECTGRENZEN	11
5.1 BINNEN SCOPE	11
5.2 BUITEN SCOPE.....	11
6 ORGANISATIE EN COMMUNICATIE.....	12
6.1 STAKEHOLDERANALYSE.....	12
6.2 COMMUNICATIE.....	12
7 PLANNING.....	13
7.1 TIJDSPANNING EN MIJLPALEN	13
7.2 TIJDSPANNING EN MIJLPALEN	13
8 COMPETENTIES EN ACTIVITEITEN	14
8.1 SMART COMPETENTIEDOELEN.....	14
8.2 ACTIVITEITEN EN REFLECTIE.....	15
9 BIBLIOGRAFIE	15
10 CONCLUSIE.....	15
10.1 GOEDKEURING EN HANDTEKENINGEN	15



1 Inleiding

1.1 Beknopte beschrijving

Dit plan van aanpak (PvA) beschrijft de aanpak voor het afstudeerproject waarin een embedded apparaat wordt ontworpen en ontwikkeld. Het doel is om een technisch probleem te analyseren, een oplossing te ontwerpen en deze te implementeren. Dit omvat zowel de hardware- als softwareontwikkeling van een datalogger die wordt ingezet bij forensisch onderzoek naar verkeersongevallen. Het PvA dient als leidraad voor het project en wordt gebruikt om de voortgang te monitoren en afwijkingen te onderbouwen.

1.2 Aanleiding

Bij ernstige verkeersongevallen onderzoekt het team Forensische Opsporing de toedracht. Op kruispunten dat wordt geregeld met verkeerslichten, slaat de verkeersregelinstallatie (VRI) relevante data op. Wanneer een voertuig het kruispunt nadert, registreert een detectielus in het wegdek verstoringen in het magnetische veld, die worden gelogd. Deze data kan inzicht geven in snelheid en roodlichtnegatie.

Een cruciaal aspect hierbij is de nauwkeurigheid van de tijdstempel. De VRI-computer registreert tijdswaarden 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. Van de meting wordt een tabel gemaakt, in de tabel is te zien wat de tijdsnemer was van de VRI, de datalogger en wat het verschil is (zie rode inzet tabel 2). Een afwijking van 0,1 seconde kan een verschil tot wel tot 30 km/h veroorzaken.

De huidige datalogger voldoet echter niet meer aan de nu gestelde eisen. De tekortkomingen liggen onder andere op het gebied van verouderde hard- en software en het missen van functionaliteiten. Daarom krijg ik de mogelijkheid om een nieuw en verbeterd model te ontwikkelen, met een focus op betrouwbaarheid, gebruiksgemak en toekomstbestendigheid.

1.3 Doel document

Dit plan van aanpak dient als leidraad voor de afstudeerperiode en beschrijft de opdracht, gebruikte methoden en verwachte resultaten. Daarnaast biedt het een toetsingskader voor het bewaken van de voortgang en kwaliteit van het project en fungeert het als communicatiemiddel tussen de opdrachtgever, begeleider en andere stakeholders.



Titel	Lusnummers D05.6 - D05.4			Lusnummers D05.4 - D05.2			Gehele Traject		
	logbestand	meting	verschil	logbestand	meting	verschil	logbestand	meting	verschil
Rijproef 1	0,7	0,6940	-0,0060	1,3	1,2367	-0,0633	2,0	1,9307	-0,0693
Rijproef 2	0,4	0,4792	0,0792	0,9	0,8547	-0,0453	1,3	1,3339	0,0339
Rijproef 3	0,5	0,5207	0,0207	1,1	1,1101	0,0101	1,6	1,6308	0,0308
Rijproef 4	0,5	0,4346	-0,0654	0,7	0,7792	0,0792	1,2	1,2138	0,0138
Rijproef 5	0,5	0,5649	0,0649	1,1	1,0087	-0,0913	1,6	1,5736	-0,0264
Rijproef 6	0,4	0,4003	0,0003	0,7	0,7180	0,0180	1,1	1,1183	0,0183
Rijproef 7	0,5	0,5043	0,0043	1,0	0,9007	-0,0993	1,5	1,4050	-0,0950
Rijproef 8	0,4	0,4156	0,0156	0,7	0,7431	0,0431	1,1	1,1587	0,0587
Rijproef 9	0,6			0,7			1,3		
Rijproef 10	0,3	0,4050	0,1050	0,8	0,7250	-0,0750	1,1	1,1300	0,0300
Rijproef 11	0,7	0,6837	-0,0163	1,2	1,2206	0,0206	1,9	1,9043	0,0043
Rijproef 12	0,5	0,4684	-0,0316	0,8	0,8351	0,0351	1,3	1,3035	0,0035
Rijproef 13	0,6	0,6090	0,0090	1,0	1,0890	0,0890	1,6	1,6980	0,0980
Rijproef 14	0,4	0,4369	0,0369	0,8	0,7788	-0,0212	1,2	1,2157	0,0157
Rijproef 15	0,5	0,5551	0,0551	1,0	0,9917	-0,0083	1,5	1,5468	0,0468
Rijproef 16	0,4			0,7			1,1		
Rijproef 17	0,5	0,5103	0,0103	1,0	0,9106	-0,0894	1,5	1,4209	-0,0791
Rijproef 18	0,4	0,4055	0,0055	0,7	0,7253	0,0253	1,1	1,1308	0,0308
Rijproef 19	0,4	0,4705	0,0705	0,9	0,8390	-0,0610	1,3	1,3095	0,0095
Rijproef 20	0,4			0,8			1,2		
Afstand detectiepunten (PD onderzoek)	17,02			30,07			47,09		
Afstand detectiepunten (nader onderzoek)	16,73			29,85			46,58		
Mectom nauwkeurigheid (m)	0,29			0,22			0,51		
Ondergrens voor afstand (m)	16,73			29,85			46,58		
Bovengrens voor afstand (m)	17,31			30,29			47,6		
Tijd bij ongeval in log-bestand (s)	0,5			0,8			1,3		
Gemiddeld verschil (s)	0,0211			-0,0137			0,0073		
Standaarddeviatie v/h verschil (s)	0,0432			0,0600			0,0500		
Indicatie voor afwijking?	nee			Nee			Nee		
Centrale waarde voor ondergrens (s)	0,521			0,786			1,307		
Centrale waarde voor boven grens (s)	0,521			0,786			1,307		
Aantal rijproeven	17			17			17		
Aantal vrijheidsgraden T-verdeling	16			16			16		
Grens	Lusnummers D05.6 - D05.4			Lusnummers D05.4 - D05.2			Gehele Traject		
	Niveau	Tijd	Snelheid	Niveau	Tijd	Snelheid	Niveau	Tijd	Snelheid
Ondergrens	99%	0,6326	95	99%	0,9414	114	99%	1,4364	117
Bovengrens	99%	0,4095	152	99%	0,6311	173	99%	1,1782	145

Tabel 2: Rekenblad van validatieonderzoek (Bron: rechtspraak ECLI:NL:RBZWB:2023:8013).



2 Probleemanalyse en opdrachtomschrijving

2.1 Probleemstelling

Binnen de nationale politie zijn er 11 teams Forensische Opsporing, waarvoor meetapparatuur bij voorkeur centraal wordt ingekocht. Dit gebeurt meestal via een landelijke aanbesteding, waarbij alle teams inspraak hebben in de functionaliteiten. Omdat de politie vaak specifieke metingen uitvoert, worden er compromissen gesloten om een apparaat te kiezen dat zo breed mogelijk inzetbaar is. Maatwerk wordt zelden overwogen vanwege de hoge kosten en lange levertermijnen.

Meetapparaten gaan gemiddeld tien jaar mee, in combinatie met een onderhoudscontract. Momenteel zijn er verschillende meetinstrumenten in omloop, waaronder een datalogger die wordt gebruikt voor validatieonderzoek bij verkeersregelinstallaties (zie afbeelding 1). Dit apparaat is echter uitgefaseerd; het onderhoudscontract is verlopen en reparaties zijn niet meer mogelijk, want reserveonderdelen zijn slecht verkrijgbaar. De software van de datalogger is niet compatible met de huidige versie van Windows. Hierdoor ontstaat de noodzaak voor een vervangend meetinstrument.



Afbeelding 1: De huidige datalogger voor de VRI.



2.2 Doelstelling

Het doel van dit project is het ontwikkelen van een functionerende prototype van de nieuwe datalogger die tijdens rijproeven nauwkeurig de tijdwaarneming kan vastleggen. Dit apparaat moet de huidige, verouderde hardware en software van de huidige datalogger vervangen en specifiek worden ingezet voor de validatie van verkeersregelinstallaties (VRI).

Omdat de afstudeerperiode beperkt is, richt het project zich op het ontwerpen en implementeren van één werkend prototype met een functionele module voor VRI-validatie. De focus ligt op:

- Ontwikkeling van een embedded hardware-oplossing met de juiste aansluitingen en interfaces.
- Implementatie van software/firmware voor het vastleggen en verwerken van meetdata.
- Betrouwbare tijdwaarneming en synchronisatie voor forensisch gebruik.
- Modulariteit, zodat toekomstige functionaliteiten eenvoudig kunnen worden toegevoegd.

Het eindproduct van dit project is een werkend prototype dat voldoet aan de functionele eisen voor VRI-validatie. Volledige implementatie en integratie in politieprocessen vallen buiten de scope.

2.3 Eindresultaat

Het eindresultaat van dit project is een functionerende datalogger die tijdens referentieritten betrouwbare meetgegevens registreert en opslaat. De datalogger wordt gebruikt in een referentievoertuig voor de validatie van detectielussen. Aan de buitenzijde van het voertuig wordt een optische sensor geplaatst om referentiepunten te detecteren. Met het referentievoertuig worden meerdere referentieritten gereden, met verschillende snelheden. Na afloop worden de tijdwaarnemingen van de datalogger met elkaar vergeleken om de nauwkeurigheid vast te stellen over meerdere snelheden. De datalogger zal een handzaam formaat krijgen.

De datalogger zal de volgende functionaliteiten bevatten:

- 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.



Afbeelding 2: Referentievoertuig tijdens positioneren lusdetectie.



Om te valideren of de ontwikkelde datalogger correct functioneert, zal een testsysteem worden opgezet waarmee meetresultaten geverifieerd kunnen worden met de huidige VRI. Dit testsysteem valt binnen de scope van dit project. Dit project richt zich uitsluitend op de ontwikkeling en initiële testfase van de datalogger. Volledige implementatie en integratie in het politieproces vallen buiten de projectomvang.

3 Projectactiviteiten en Fasering

Dit project bestaat uit verschillende stappen die ik ga nemen om het project en de afstudeeropdracht te kunnen voltooien. Door mij zijn in dit hoofdstuk een overzicht van de stappen weergegeven en hoe deze in de tijd worden gepland.

3.1 Overzicht Activiteiten

Oriëntatie & Analyse

- Probleemanalyse.
- Wat is de huidige vorm van valideren?
- Gebruikerseisen en randvoorwaarden specificeren.
- Projectplanning.
- Plan van aanpak;

Onderzoek

Hoofdonderzoeksvraag:

- Welke hard- en software is het meest efficiënt en effectief voor een datalogger, die nauwkeurig de tijdwaarneming kan loggen na een inkomend signaal?

Subvragen:

- Welke hardwarecomponenten zijn geschikt voor het detecteren en verwerken van een inkomend signaal met een hoge nauwkeurigheid?
- Welke software-architectuur zorgt voor een efficiënte verwerking en opslag van tijdstempels?
- Wat zijn de prestatie-eisen voor een nauwkeurige tijdsvergelijking?
- Welke opslag- en verwerkingsmogelijkheden zijn nodig voor logging?
- Hoe kan de datalogger worden getest en gevalideerd op nauwkeurigheid?

Ontwerp (functioneel en technisch)

- Systeemarchitectuur en interface.
- Gedetailleerd ontwerp van componenten.

Realisatie (prototypeontwikkeling)

- Code schrijven en prototype bouwen.

Documentatie en Oplevering

- Unit testen.
- Systeemtesten.
- Acceptatietesten.
- Systeemvalidatie door het Nederlands Meet Instituut of het Nederlands Forensisch Instituut.
- Oplevering eindproduct.
- Documenteren.



4 Kwaliteitswaarborging en Methodologie

Om de kwaliteit van het eindproduct en de uitvoering van het project te borgen, zijn door er kwaliteitseisen, werkwijze en methodes en vastgesteld in dit hoofdstuk.

4.1 Kwaliteitseisen

Het eindproduct moet voldoen aan de volgende kwaliteitseisen:

1. Bedieningsgemak & gebruiksvriendelijkheid

- De interface moet intuitief en eenvoudig te bedienen zijn.
- Installatie en configuratie moeten duidelijk en zonder specialistische kennis mogelijk zijn.
- Knoppen en menu's moeten logisch en toegankelijk zijn.

2. Betrouwbaarheid & robuustheid

- Storingen moeten eenvoudig op te lossen zijn, bijvoorbeeld door een foutdiagnosesysteem of logging.
- Het systeem moet langdurig kunnen werken zonder handmatige herstarten of resets.
- Hardware en behuizing moeten bestand zijn tegen trillingen en temperatuurschommelingen.

3. Nauwkeurigheid & prestaties

- De klok moet een frequentie van 10 kHz (1 ms nauwkeurigheid) halen.
- Latency van signaaldetectie moet minimaal zijn.
- Synchronisatie tussen verschillende componenten moet binnen een acceptabele foutmarge blijven (bijv. max. afwijking ± 1 ms).

4. Opslag & data-integriteit

- Gegevens moeten consistent en foutloos worden weggeschreven naar een USB-stick of SD-kaart.
- Bestandsgrootte en opslagformaat moeten geoptimaliseerd zijn voor snelle uitlezing en verwerking.

5. Sensorvalidatie & foutpreventie

- Valse detecties van de laserafstandssensor moeten worden geminimaliseerd door filtering of signaalverwerking.
- Omgevingsfactoren (lichtinval, stof, obstakels) mogen de werking niet beïnvloeden.



4.2 Werkwijze en Methodes

Ontwerpmethodologie

- Gevolgd wordt het V-model aanpak: eerst specificeren, ontwerpen, testen en daarna valideren.
- Componenten worden modulair ontworpen zodat aanpassingen per onderdeel mogelijk zijn.
- Documentatie van de software en hardware zodat latere uitbreidingen mogelijk zijn.

Testmethoden

- **Unit testing:** Elk hardware- en softwarecomponent wordt afzonderlijk getest.
- **Integratietesting:** De samenwerking tussen componenten wordt getest om communicatieproblemen te detecteren.
- **Systeemtesten:** De volledige datalogger wordt getest met real-world inputs.
- **Validatietesten:** De metingen worden vergeleken met een referentieklok om nauwkeurigheid te verifiëren.

Reviewprocessen & feedback

- Wekelijkse evaluaties met praktijkbegeleider.
- Maandelijkse review met externe stakeholders.
- Feedback wordt verwerkt via een versiebeheersysteem.

Versiebeheer

- Gebruik van GitHub.

4.3 Risicomanagement

Bij de ontwikkeling van de nieuwe datalogger zijn er verschillende risico's die de nauwkeurigheid en betrouwbaarheid kunnen beïnvloeden. In onderstaande tabel staan de belangrijkste risico's, hun impact en de maatregelen om ze te beheersen.

De focus ligt op een nauwkeurige tijdsregistratie, betrouwbare sensorwaarnemingen en correcte data-opslag. Daarnaast worden mogelijke vertragingen en hardwareproblemen meegenomen in de planning om het project soepel te laten verlopen.

Risico	Impact	Kans	Maatregelen
Nauwkeurigheid lager dan 10 kHz	Hoog	Middel	Testen met verschillende timers en klokken, optimaliseren code en signaalverwerking.
Sensor detecteert valse signalen	Hoog	Laag	Filtertechnieken implementeren, experimenteren met drempelwaarden.
Data wordt niet correct opgeslagen	Hoog	Middel	Opslagmethodes testen, redundante opslagopties inbouwen (bijv. checksum-controle).
Project loopt uit door technische complexiteit	Hoog	Middel	Strikte planning en prioritering van kritieke onderdelen.
Hardware-componenten niet op tijd beschikbaar	Middel	Hoog	Alternatieve componenten identificeren en testopstelling maken met vervangbare modules.
Lichte trillingen beïnvloeden metingen	Laag	Middel	Testen met bevestiging van sensor, trillingsdemping toevoegen indien nodig.
Display werkt niet optimaal onder fel zonlicht en te fel in het donker	Laag	Laag	Contrast en helderheid instellen, zonneschermje overwegen.

Tabel 3: Risico inventarisering.



5 Projectgrenzen

Het is belangrijk de grenzen van de werkzaamheden te definiëren, omdat de datalogger deel uitmaakt van een groter project. In dit hoofdstuk wordt beschreven welke activiteiten binnen en buiten de scope vallen.

5.1 Binnen scope

- **Opstellen van eisen:** Definitie van functionele en technische eisen.
- **Hardware-ontwerp:**
 - Selectie en ontwerp van de printplaat (indien nodig).
 - Aansluiting van de microcontroller op sensoren en interface.
 - Voeding en signaalintegriteit.
- **Softwareontwikkeling:**
 - Microcontroller-code in C++ voor data-acquisitie en opslag.
 - Timing en synchronisatie van metingen.
 - Weergave van de live-meetgegevens.
 - Opslag en uitlezing van data.
- **Testen en optimaliseren:**
 - Valideren van de nauwkeurigheid van de tijdsregistratie.
 - Controleren op storingen, foutafhandeling en een testplan.
 - Debuggen en optimaliseren van code en signaalverwerking.
- **Documentatie:**
 - Technische documentatie en gebruikershandleiding.
 - Mogelijkheden voor toekomstige uitbreidingen benoemen.

5.2 Buiten scope

- Geavanceerde dataverwerking en analyse: Het verwerken en interpreteren van de verzamelde data gebeurt extern.
- Mechanische behuizing: Geen ontwerp en productie van een specifieke behuizing.
- Ontwerpen van printplaat in Kicad.



6 Organisatie en Communicatie

Binnen dit project zijn verschillende belanghebbenden betrokken. Zij hebben een rol in de ontwikkeling, het testen en de uiteindelijke toepassing van de datalogger. Dit hoofdstuk beschrijft wie deze stakeholders zijn en hoe de communicatie met hen wordt vormgegeven.

6.1 Stakeholderanalyse

Men stakeholderanalyse helpt om te bepalen welke partijen belang hebben bij dit project en in welche mate zij invloed uitoefenen. De rollen van stakeholders kunnen worden onderverdeeld in vier categorieën:

- **Toeschouwer:** Heeft weinig invloed, maar wordt op de hoogte gehouden.
- **Geïnteresseerde:** Heeft belang bij de uitkomst van het project, maar geen directe invloed.
- **Beïnvloeder:** Kan de richting van het project bijsturen door advies of eisen.
- **Sleutelfiguur:** Heeft zowel grote invloed als een groot belang in het project.

In **Tabel 1** staat een overzicht van de betrokken stakeholders, hun functie en rol in het project.

Persoon/Organisatie	Functie	Rol
Nikolai Lieshout	Eindgebruiker van de datalogger	Sleutelfiguur/ Beïnvloeder
Forensisch onderzoekers	Gebruikers van de datalogger in het veld	Geïnteresseerde
Peter Bijl	Docentbegleider	Toeschouwer
Mark Bos	Digitaal Rechercheur	Sleutelfiguur/ Beïnvloeder
Aart Spek	Onderzoeker Nederlands Forensisch Instituut	Geïnteresseerde
Softwareontwikkelaar (ik)	Ontwerper en bouwer van het systeem	Sleutelfiguur

Tabel 4: Rolverdeling stakeholders

6.2 Communicatie

Een effectieve communicatie tussen de stakeholders is essentieel voor een succesvolle uitvoering van het project. De volgende communicatiemethoden worden gehanteerd:

- **Wekelijks overleg** met de opdrachtgever om de voortgang te bespreken en feedback te verwerken.
- **Periodieke korte rapportages** aan de forensisch onderzoekers over testresultaten en functionaliteit.
- **Begeleidingsgesprekken** met de docent om de afstudeereisen te bewaken (bijv. eens per drie weken).
- **Ad-hoc communicatie** met technisch onderlegde collega's via e-mail of Teams bij technische vraagstukken.

Door deze aanpak blijven alle betrokkenen goed geïnformeerd en kan het project efficiënt verlopen.



7 Planning

7.1 Tijdsplanning en Mijlpalen

Een goed gestructureerde planning is essentieel voor het succesvol afronden van dit project. Tabel 5 geeft een overzicht van de verschillende fasen, inclusief de start- en einddata van de activiteiten. Deze planning dient als richtlijn en is niet leidend. Tegenvalters en uitlopende activiteiten zijn reëel, waardoor sommige onderdelen later kunnen starten dan gepland.

De afstudeerperiode start in week 0 op maandag 3 februari 2025 en eindigt op maandag 2 juni 2025, de uiterste inleverdatum voor de eindrapportage.

Activiteit	3-feb	10-feb	17-feb	24-feb	3-mrt	10-mrt	17-mrt	24-mrt	31-mrt	7-apr	14-apr	21-apr	28-apr	5-mej	12-mej	19-mej	26-mej	2-jun	9-jun	16-jun
	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19
Plan van Aanpak																				
Analyseren																				
Onderzoek																				
Functioneel ontwerp																				
Technisch ontwerp																				
Prototype																				
Test systeem																				
Realiseren																				
Testen																				
Rapportage																				

Tabel 5: Gantt Chart tijdplanning.

De belangrijkste mijlpalen in dit project zijn (zie rode vlakken in tabel 5):

- Week 3, uiterlijk zondag 2 maart: Afronding Plan van Aanpak.
- Week 6, uiterlijk op zondag 23 maart: Eerste concept Functioneel Ontwerp.
- Week 10, uiterlijk op zondag 20 april: Prototype gereed voor eerste tests.
- Week 13, uiterlijk op zondag 11 mei: Volledige testresultaten beschikbaar.
- Week 14, uiterlijk op zondag 18 mei: Concept Eindrapport gereed.
- Week 16, uiterlijk op maandag 2 juni: Definitieve inlevering eindrapportage.

Omdat ik graag gericht naar deadlines toe werk, zijn mijlpalen in rood gemarkeerd. Zo is direct duidelijk wanneer een onderdeel afgerond moet zijn.

De deadline voor het eindrapport is maandag 2 juni 2025. Het conceptrapport dient uiterlijk zondag 18 mei gereed te zijn, zodat er tijd is voor optimalisatie en feedback te verwerken.

7.2 Tijdsplanning en Mijlpalen

Het project is opgedeeld in verschillende fasen, waarbij elke fase logisch voortbouwt op de vorige:

1. **Initiële onderzoeksfase** (Week 1-4): Definitie van eisen, onderzoek en hardwarekeuze.
2. **Ontwerpfase** (Week 5-9): Opstellen van functioneel en technisch ontwerp, softwareontwikkeling starten.
3. **Prototypefase** (Week 10-12): Fysiek bouwen en eerste tests uitvoeren.
4. **Test- en validatiefase** (Week 13-15): Optimalisatie en eindvalidatie van resultaten.
5. **Documentatie en afronding** (Week 16-17): Rapportage, reflectie en voorbereiden van de eindpresentatie.

Door deze fasering is er voldoende tijd voor ontwikkeling, testen en bijsturing, zodat het eindresultaat aan alle eisen voldoet.



8 Competenties en Activiteiten

Tijdens dit project werk ik aan verschillende competenties die aansluiten bij de eisen van de Bachelor Engineering Elektrotechniek, met een specialisatie in Embedded Systems Engineering. In dit hoofdstuk wordt beschreven hoe deze competenties worden toegepast en ontwikkeld. De competenties zijn gebaseerd op de Afstudeergids AD Engineering 2024-2025 (Steenbergen, 2024), waarbij het Domeincompetenties Elektrotechniek (zie Figuur 1) als leidraad is gebruikt.

De mate van competentiebeheersing wordt beoordeeld op vier niveaus, variërend van instroomniveau (0) tot zelfstandig functioneren in complexe, multidisciplinaire contexten (III). Het radarplot in Figuur 1 biedt een visuele weergave van de ontwikkeling van deze competenties binnen de opleiding. In de grafiek vertegenwoordigt de binnenste ring het instroomniveau (0), terwijl de buitenste ring het hoogste niveau (III) aangeeft.



Figuur 1: Domein competenties Engineering Elektrotechniek.

8.1 SMART Competentiedoelen

1. Analyseren

- Bepalen welke eisen de datalogger moet hebben.
- Controleren hoe nauwkeurig de metingen zijn en waar mogelijke fouten zitten.
- Overleggen met stakeholders om af te stemmen wat de datalogger moet kunnen.
- Onderzoeken welke bestaande systemen en methodes al worden gebruikt voor datalogging.

2. Ontwerpen

- Ontwikkelen van een functioneel en technisch ontwerp voor de datalogger.
- Opstellen van software- en hardware-architectuur.
- Bepalen van de noodzaak en opzet van een testomgeving.
- Keuze en implementatie van geschikte opslag- en communicatiemethoden.

3. Realiseren

- Implementeren van de firmware voor de datalogger.
- Bouwen van een werkend prototype en eerste functionele tests uitvoeren.
- Het systeem verbeteren op basis van testresultaten.

4. Beheren

- Opstellen en bewaken van de projectplanning en fasering.
- Documenteren van ontwerukeuzes, testresultaten en optimalisaties.
- Structureren van code en hardware keuzes voor toekomstige uitbreidingen.
- Periodieke rapportage aan begeleiders en stakeholders.



5. Managen

- Zorgen voor een tijdige en succesvolle afronding van het project.
- Prioriteren van taken en efficiënte tijdsindeling waarborgen.
- Problemen en tegenslagen signaleren en hierop anticiperen.
- Zorgen voor een goede afstemming met stakeholders en gebruikers.

6. Adviseren

- Stakeholders informeren over de mogelijkheden en beperkingen van de datalogger.
- Beoordelen of de gekozen oplossing voldoet aan de forensische eisen.

7. Onderzoeken

- Vergelijken van verschillende hardware- en softwareopties voor de datalogger.
- Onderzoeken hoe de timing van de meetgegevens geoptimaliseerd kan worden.
- Evaluieren van methoden voor foutdetectie en data-integriteit.
- Bepalen hoe de datalogger kan worden geïntegreerd in bestaande systemen.

8. Professionaliseren

- Effectief communiceren met stakeholders en teamleden.
- Zelfstandig werken en verantwoordelijkheid nemen voor projecttaken.
- Proactief problemen signaleren en oplossingen aandragen.
- Reflecteren op eigen werkwijze en feedback toepassen.

8.2 Activiteiten en Reflectie

Tijdens dit afstudeerproject voer ik verschillende activiteiten zelfstandig uit die bijdragen aan de ontwikkeling van de hierboven beschreven competenties. Deze activiteiten omvatten zowel technische taken, zoals het analyseren van eisen en het ontwikkelen van firmware, als organisatorische en communicatieve taken, zoals het beheren van de projectplanning en het afstemmen met stakeholders.

Om de voortgang van mijn competentieontwikkeling te bewaken, maak ik gebruik van tussentijdse evaluaties. Dit doe ik door mijn werkzaamheden periodiek te reflecteren, feedback te verzamelen en waar nodig mijn werkwijze aan te passen.

9 Bibliografie

Steenbergen, R. (2024). *Afstudeergids AD Engineering 2024-2025*. Arnhem.

10 Conclusie

Dit plan van aanpak dient als leidraad voor het uitvoeren van de afstudeeropdracht. Het biedt een gestructureerde werkwijze en duidelijke doelen om de opdracht succesvol af te ronden. Aanpassingen in het plan worden onderbouwd en besproken met de begeleiders.

10.1 Goedkeuring en Handtekeningen

Naam student: Maarten van Riel

Bedrijfsbegeleider: Mark Bos

Datum: dinsdag 18 februari 2025

Bijlage B Reflectieverslag

1. Met welke bagage startte ik?

In december 2024 had ik een afstudeeropdracht gevonden binnen de politiepraktijk. Er werd daar gewerkt met een gebruiksvriendelijke en verouderde datalogger. De kans om zelf een nieuwe, gebruiksvriendelijke versie te ontwerpen, sloot naadloos aan bij mijn interesse in techniek én praktijkgericht werken. Ik begon dit project met theoretische kennis vanuit de opleiding én praktijkervaring met bestaande dataloggers. Hoewel ik nog geen ervaring had met embedded systemen, zag ik dit juist als leerzame uitdaging. Mijn houding ten opzichten van het vak was positief en nieuwsgierig. Ik wilde mezelf verdiepen (onderdompelen) in embedded development en was gedreven om een functioneel en toekomstigbestendig product op te leveren.

2. Wat waren mijn persoonlijke leerdoelen?

Mijn leerdoelen heb ik gebaseerd op competenties uit het plan van aanpak (pagina 14 [bijlage A](#)). Ik wilde onder andere leren:

- Hoe ik zelfstandig een embedded systeem ontwerp en ontwikkel (analyseren, ontwerpen, onderzoek en realiseren)
- Hoe ik technische beslissingen kan onderbouwen en hierover communiceer met stakeholders (adviseren, communiceren)
- Hoe ik het overzicht houd over een project en mezelf aan een planning houd (managen)

Bereikt?

Grotendeels wel. De firmware is grotendeels zelfstandig ontworpen, geschreven en getest. Wel blijft het een aandachtspunt dat ik de neiging om pas hulp in te schakelen als ik er echt niet meer uitkom. Daardoor mis ik soms de kans om eerder te leren van anderen. Ik merk dat het bespreken cab code met anderen niet alleen tot verdieping leidt, maar ook mijn bewustzijn vergroot van mogelijke fouten of inefficiënties in mijn aanpak

Tijdens het project heb regelmatig overleg gehad met de stakeholders over specificaties en voortgang. Zo heb ik in een vroeg stadium het ontwerp van het OLED-display aangepast op basis van wensen. Ook heb ik in overleg besloten om GPS-functionaliteiten buiten de scoop te laten vallen, omdat dit niet tot de hoogste prioriteiten behoorde.

Mijn tijdsplanning heb ik gedurende het project meerdere keren bijgestuurd, vooral richting het einde toen het realiseren, testen en debuggen meer tijd vergden dan verwacht. Door deze aanpassingen. Heb ik geleerd welke keuzes cruciaal zijn tijdens het ontwerp- en realisatieproces, en hoe belangrijk het is om voortgang helder te documenteren.

3. Wat wilde ik verder bereiken?

Ik wilde laten zien dat ik een technisch complex project zelfstandig kan uitvoeren, met oog voor eindgebruikers. Ook wilde ik mijn zelfvertrouwen vergroten in het nemen van technische beslissingen.

Bereikt?

Ja, ik heb ervaren dat ik zelfstandig een functioneel prototype kan ontwerpen en realiseren. Zowel mijn begeleider als de gebruikers gaven mij positieve feedback op mijn aanpak en het eindproduct. Tegelijkertijd realiseerde ik me dat ik nog wil groeien in het onderbouwen van ontwerpkeuzes, bijvoorbeeld met behulp van literatuur, datasheets of vergelijkbare toepassingen.

Een uitdaging waar ik tegenaan liep, is mijn neiging om te verzanden in details. Het grondig onderzoeken van componenteneigenschappen is leerzaam, maar bleek ook zeer tijdrovend. Hierdoor verloor ik soms het overzicht.

Wel weet ik dat ik in de toekomst nog wil groeien in het onderbouwen van keuzes met literatuur en benchmarks. Ik vind het lastig om niet te verzanden in allerlei details. Het uitzoeken van de exacte specifieke eigenschappen van componenten is leerzaam, maar te tijdrovend. Verder moet ik mij beperken tot een kleine selectie van componenten.

4. Motivatie en concentratie

Mijn motivatie was hoog aan het begin en groeide naarmate ik meer overzicht kreeg. Een kantelpunt was het moment waarop het eerste prototype werkte; dat gaf voldoening. Wat me soms demotiveerde was wanneer een technische oplossing meer tijd kostte dan gepland, of als een bug hardnekkig bleef. Het had mij zeker geholpen als ik eerder hulp had ingeschakeld. Concentratie was meestal geen probleem, al moest ik mezelf soms dwingen tot pauzes om mijn focus te behouden.

5. Tijdsplanning en aanpak

Ik werkte met een globale planning, uitgewerkt op een whiteboard in mijn studeerkamer. Daarnaast maakt ik per week een dagindeling in tabelvorm, waarin ik specificeerde welke taken ik op welke dag wilde afronden. Deze combinatie gaf mij houvast en overzicht tijdens het project.

Gedurende het proces bleek echter dat met name het onderzoekswerk, de realisatiefase en het documenteren meer tijd vroegen dan vooraf ingeschat. Daardoor kwam ik richting het einde van het project in tijdnood. Door bewust prioriteiten te stellen en concessies te doen – zoals het tijdelijk schrappen van minder essentiële functies – kon ik de planning bijstellen en toch een werkend prototype opleveren binnen de afgesproken termijn. Het prototype bevatte nog een belangrijke bug: de secondeweergave is niet altijd consistent en kan incidenteel één seconde achter- of voorlopen. Hoewel dit geen directe impact had op de functionele werking binnen het testgebruik, is deze afwijking voor de forensische opsporing wel kritisch en behoeft verdere verfijning in een volgende ontwikkelfase.

6. Wat ging gemakkelijk, wat was lastig?

Gemakkelijk: het structureren van code, het bijhouden van technische documentatie, en het voeren van overleg met betrokkenen verliep soepel.

Lastig: Debuggen kostte veel tijd, vooral wanneer bugs zich onvoorspelbaar voordeden. Ook vond ik het lastig om vooraf goed in te schatten hoeveel tijd een taak daadwerkelijk zou kosten. Daarnaast had ik moeite om voldoende afstand te nemen van mijn werk om objectief te reflecteren. Ik twijfelde soms of ik mezelf tekort deed of juist overschatte, wat de zelfevaluatie bemoeilijkte.

7. Belemmeringen

Een belangrijke belemmering zat in mijn eigen perfectionisme. Ik wilde het systeem “af” hebben voordat ik feedback vroeg. Daardoor heb ik soms te laat geschakeld of iteraties uitgesteld. Ook merkte ik dat geneigd ben om te lang zelfstandig naar een oplossing te zoeken, vanuit de gedachte dat ik het eerst zelf moet kunnen oplossen. Daardoor heb ik waardevolle tijd verloren die ik had kunnen besparen door eerder te overleggen of hulp in te schakelen.

8. Feedback

- Van medestudenten: herkende ik vooral praktische tips “maak het niet te complex” en “test eerder met gebruikers, dan heb je meer tijd over voor aanpassing en finetunen”.

- Van mijn docent: het concept van het productdocument werd als goed en veelbelovend beoordeeld.
- Van gebruikers: positieve feedback over het gebruiksgemak en de duidelijke structuur van het OLED-scherm.
- Van mijn praktijkbegeleider: Complimenten over mijn aanpak, maar ook een kritische noot: het programmeren kostte te veel tijd, en het monteren van het prototype kon niet uitgesteld worden. Die feedback opende mijn ogen. Ik realiseerde me dat mijn manier van werken nog te veel gericht is op perfectionisme, wat ten koste ging van efficiëntie en voortgang.

Deze inzichten hielpen me om keuzes te maken. In overleg met de stakeholders heb ik besloten de GPS-module niet te implementeren en de op maat gemaakte behuizing los te laten. Hoewel dat in eerste instantie voelde als ‘inleveren op kwaliteit’, bleek het uiteindelijk een realisatie en functionele oplossing.

9. Resultaat en terugblik

Ik kijk met gemengde gevoelens terug op het eindresultaat. Technisch gezien werkt het prototype naar behoren en beantwoordt het grotendeels aan de praktijkvraag. Tegelijk zie ik duidelijke verbeterpunten, zoals de bug in de tijdsweergave en het ontbreken van een gebruikerstest in een vroeg stadium. Mijn eigen leerproces was echter waardevol: ik heb veel geleerd over embedded ontwikkeling, planning en het stellen van realistische doelen. Het documenteren bleek uitdagender dan gedacht en kostte meer tijd dan gepland. Daardoor bleef er minder ruimte over voor het afronden van details.

10. Wat wil ik anders? Nieuwe leerdoelen?

- Eerder en vaker feedbackmomenten inplannen tijdens het hele proces.
- Bij dreigende achterstand eerder in actie komen, en tijdig met stakeholders overleggen over consequenties en mogelijke oplossingen.
- Sneller knopen doorhakken, vooral bij het selecteren van componenten, om niet te verzanden in eindeloos vergelijken.
- Tijd effectiever indelen: minder perfectionistisch zijn in het begin. Mijn aanpak moet meer iteratief worden: ontwerp, test, verbeter.
- Meer tijd reserveren voor het documenteren, en dit verspreiden over het project aanpakken in plaats van aan het eind.

11. Wat laat ik los?

Ik accepteer dat ik niet alles in één project kan leren. Een volledig foutloos prototype of een perfect verslag is niet realistisch. Het belangrijkste is dat ik geleerd heb hoe ik problemen systematisch aanpak, leer van feedback en reflectie, en mijn werkwijze blijf verbeteren.

Bijlage C Analyse, Onderzoek en datasheets

Onderzoek naar een Datalogger voor Validatie van Tijdwaarneming

1. Inleiding

Voor dit onderzoek heb ik mij gericht op het ontwerpen en realiseren van een datalogger die geschikt is voor de validatie van de tijdwaarneming van verkeersregelinstallaties. Als leidraad heb ik onderzoeks vragen geformuleerd tijdens het opstellen van het plan van aanpak.

De eerste onderzoeks vraag was:

"Wat is een datalogger en hoe werkt datlogging?"

Om deze vraag te beantwoorden, heb ik verschillende bronnen geraadpleegd en mijn eigen ervaring meegenomen in de analyse.

2. Wat is een datalogger en hoe werkt datlogging?

Een datalogger is een elektronisch apparaat dat gegevens verzamelt en opslaat over een bepaalde periode, vaak met behulp van ingebouwde of externe sensoren. Deze apparaten zijn doorgaans klein en draagbaar en worden gebruikt voor het monitoren van omgevingscondities zoals temperatuur, luchtvuchtigheid, schokken, licht en kanteling. Door gegevens automatisch en continu te registreren, geven dataloggers een nauwkeurig beeld van de omstandigheden waarin goederen of processen zich bevinden, zoals bij transport of industriële toepassingen.

3. Onderzoeks vragen

Om te bepalen welke specificaties de nieuwe datalogger moet hebben, heb ik de volgende onderzoeks vragen opgesteld:

Hoofdonderzoeks vraag:

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

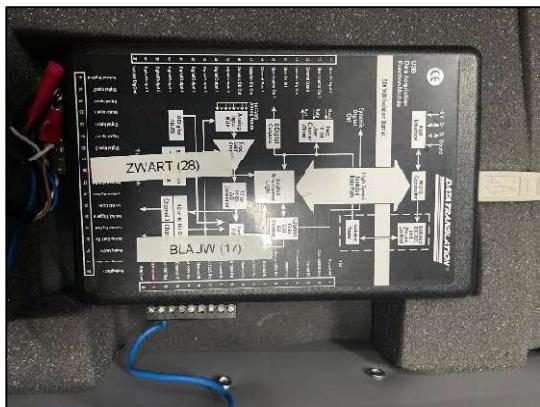
Subvragen:

- *Welke hardware componenten zijn geschikt voor het detecteren en verwerken van een inkomend signaal met hoge nauwkeurigheid?*
- *Welke software-architectuur zorgt voor een efficiënte verwerking en opslag van tijdstempels?*
- *Wat zijn de prestatie-eisen voor een nauwkeurige tijdsvergelijking?*
- *Welke opslag- en verwerkingsmogelijkheden zijn nodig voor logging?*
- *Hoe kan de datalogger worden getest en gevalideerd op nauwkeurigheid?*

4. Vergelijking van bestaande dataloggers bij de politie

Om een geschikte datalogger te ontwikkelen, heb ik onderzocht welke methodes en apparaten momenteel worden gebruikt binnen het team Forensische Opsporing van de politie Zeeland – West-Brabant. Momenteel zijn er twee systemen in gebruik:

1. Data Translation USB Data Acquisition (DAQ) Module (zie afbeelding 1)
2. Movilog BM22 (zie afbeelding 2)



Afbeelding 1:DAQ module



Afbeelding 2:Movilog BM22

De DAQ wordt gebruikt voor validatie van verkeersregelinstallaties, terwijl de BM22 ook wordt ingezet voor het meten van remvertraging van voertuigen, en specifiek voor het loggen van validatieritten met GPS.

Ik heb de specificaties van de beide dataloggers in een tabel gezet, zie hiervoor tabel 4.

Kenmerk	Movilog BM22	Data Translation DAQ
Type	Embedded apparaat	Embedded apparaat
Interface	TFT-scherm met knoppen	Geen eigen display, software-interface op laptop
Bediening	Start/stopknop, navigatieknoppen	Bediening via software
Voeding	12V-aansluiting (rechtstreeks op de accu)	12V-aansluiting
Opslag	SD-kaart	Txt-bestand op laptop
Sensortechnologie	3-assige gyroscope, GPS-module, RTC (100 Hz)	Digitale en analoge ingangen, RTC (10 kHz)
Aansluitingen	M-12 voor externe GPS-antenne, SD-kaartslot	USB naar laptop, M12 voor lichtpoortsensor
Mobiliteit & Montage	Handzaam, klein en draagbaar. Geplaatst met een zuignap aan de (voor)ruit.	Niet handzaam, vereist een laptop en bekabeling. Meestal in de kofferbak of op de achterbank geplaatst.
Soft- en hardware	Niet gespecificeerd.	Niet gespecificeerd.
Leeftijd	6 jaar	10 jaar
Onderhoud	Wordt iedere 2 jaar vergeleken met één of meerdere Mov1Log BM22.	Wordt nooit gekalibreerd.
Kosten*	± 3500 euro	± 2200 euro

Kenmerk	Movilog BM22	Data Translation DAQ
Betrouwbaarheid & Softwarefunctionaliteit	Over het algemeen goed, weinig storingen. Echter, een niet-oplosbare softwarefout vereist een arbeidsintensieve herberekening.	Goede betrouwbaarheid, functioneert zonder grote softwareproblemen. Software niet compatibel met nieuwste Windows-versies.

Tabel 7: Specificaties DAQ en Mov1Log BM22.

5. Evaluatie en Ontwikkelpunten van de huidige apparatuur

Uit eigen ervaring heb ik vastgesteld dat beide apparaten zich in de praktijk hebben bewezen, maar ook ontwikkelpunten hebben. Als basis voor de te ontwikkelen datalogger is de Mov1Log gebruikt, omdat die qua gebruik het dichts in de buurt komt van de wensen van de gebruikers/stakeholders.

Movilog BM22 - Sterk concept, maar met verbeterpunten:

- Ongunstige montagelocatie → De plaatsing op de (voor)ruit is onpraktisch, vooral omdat ruiten vaak beschadigd raken bij een aanrijding.
- Niet-oplosbare softwarefout → Een structurele softwarefout kon niet worden hersteld, waardoor een arbeidsintensieve herberekening noodzakelijk is.
- Onhandige voedingsaansluiting → De plus- en min-klemmen moeten direct op de accuklemmen worden bevestigd. Een interne batterij of alternatieve 12V-stekker zou gebruiksvriendelijker zijn.
- Geen terugkoppeling naar de gebruiker → het is niet direct zichtbaar of de gewenste data is opgeslagen. Hiervoor moeten de gegevens worden geïmporteerd in een speciaal geschreven softwareapplicatie.

6. Eisen datalogger

Gezien de sterke punten van de Movilog BM22, is dit apparaat in de basis geschikt, mits de aanpassingen van de ontwikkelpunten worden doorgevoerd.

Aangezien het huidige concept grotendeels functioneel en bewezen effectief is, zijn er geen redenen om hiervan af te wijken. Daarnaast ben ik gebonden aan een aantal landelijk vastgestelde normen (FO-norm), die voorschrijven dat validatie moeten worden uitgevoerd met een datalogger die tijdmetingen uitvoert op ten minste 1 kHz.

7. Beantwoording onderzoeks vragen

Ik heb als eerste de Subvragen beantwoord, want de Subvragen helpen bij de opbouw van het volledige antwoord op de hoofdvraag. Ik heb hiermee structuur aangebracht door van klein naar groot te werken. Eerst details (hardware/software), dan grotere lijnen (prestaties/logging) en uiteindelijk de validatie.

7.1 Hardware keuze

Eerste sub onderzoeks vrag:

Welke hardwarecomponenten zijn geschikt voor het detecteren en verwerken van een inkomen signaal met hoge nauwkeurigheid?

Microcontroller

Voor de selectie van geschikte microcontrollers heb ik eerst onderzocht welke merken en typen veel worden gebruikt in de automotive industrie en waarom juist deze microcontrollers worden gekozen.

In de automotive sector worden microcontrollers van Infineon, NXP, Texas Instruments en Renesas veel toegepast. Deze microcontrollers worden geselecteerd op basis van de volgende criteria: ondersteuning voor RTOS, deterministische reactietijd, ISO 26262 ASIL-B certificering, ondersteuning voor automotive communicatieprotocollen zoals CAN communicatie. Die toepassingen zijn voor mij niet allemaal noodzakelijk, op de

ondersteuning voor RTOS na. Hierdoor wijk ik af van de traditionele en verplichte eisen die gesteld worden aan de microcontrollers.

Ik heb aan de microcontroller de volgende voor het project belangrijke eisen gesteld: CPU snelheid en architectuur, geheugen (RAM & FLASH), RTOS, I/O, SPI/I²C/UART, DMA en voldoende ondersteuning van commodity's en standaard bibliotheken. Om het enigszins een beetje overzichtelijk te houden heb ik de volgende microcontrollers vergeleken op de bovengenoemde eisen:

- Arduino Nano ESP32-S3;
- Teensy 4.1;
- STM32F4;
- NXP KL25Z;
- Rasp Berry PI.

Definitieve keuze: Arduino Nano ESP32-S3

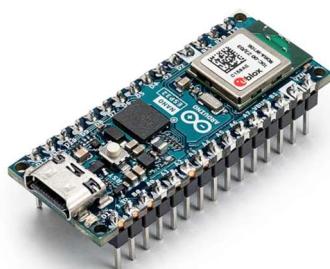
Omdat de microcontrollers nauwelijks voor elkaar onder doen heb ik uiteindelijk gekozen tussen twee veelzijdige microcontrollers: de Arduino Nano ESP32-S3 en de Teensy 4.1. De specificaties van beide microcontrollers liggen dicht bij elkaar, maar de doorslaggevende factoren waren mijn bekendheid met het Arduino-ecosysteem en de aanwezigheid van een dual-core processor. Daarom is mijn keuze gevallen op de Arduino Nano ESP32-S3 (zie afbeelding 3).

Hoewel de Teensy 4.1 een krachtige microcontroller is met uitgebreide mogelijkheden, vond ik de beschikbare tijd voor mijn afstudeerproject te beperkt om hiermee aan de slag te gaan. Daarom heb ik gekozen voor de optie waarbij ik sneller productief kan zijn.

De datasheet van de Arduino Nano ESP32-S3 is als bijlage toegevoegd.

Omdat de tijdwaarneming van de datalogger belangrijk is, heb ik uitgezocht wat de verwerkingstijd is door de Arduino Nano ESP32-S3. GPIO Interrupts (zoals een knop indrukken of een sensor-trigger) hebben meestal een latentie van 5-10 µs, afhankelijk van de systeembelasting. Deze vertraging is zo goed als te verwaarlozen.

De datasheet van de Arduino is als [bijlage C7](#) toegevoegd.



Afbeelding 3: De Arduino Nano ESP32.

Afstandssensor

Voor de afstandssensor heb ik gezocht naar een vergelijkbaar model als het exemplaar dat momenteel wordt gebruikt in de DAQ-module. Op basis daarvan heb ik de volgende minimale technische specificaties opgesteld:

- Aansluiting: M12, 4-polig.
- Beschermingsklasse: Minimaal IP67.
- Meetbereik: Minimaal 1 meter.
- Uitgangssignaal: digitaal (PNP of NPN).
- Montage: Moet in een bepaalde hoek geplaatst kunnen worden.
- Kosten: Maximaal 300 euro (om het enigszins betaalbaar te houden)
- Schakelfrequentie minimaal 10 kHz;
- Aanspreekijd minimaal 0,1 ms.

De huidige sensor Leuze PRK 96K/N-1380-46 heeft een schakelfrequentie van 500 Hz en een aanspreekijd van 1 ms. De DAQ logt met 10 kHz, wat betekent dat er twintig keer zo vaak wordt gemeten als de sensor kan schakelen. Dit is overbodig, omdat de sensor niet sneller dan 500 metingen per seconde kan verwerken. Hierdoor worden veel metingen opgeslagen zonder dat de sensorwaarde daadwerkelijk verandert. Door mij werd besloten om de collega's op de hoogte te brengen die zich met deze materie bezig houden. Na overleg met de stakeholders is besloten dat ik een datalogger ga bouwen die in 1 kHz meet, wat min of meer al werd gedaan.

De belangrijkste vereiste is dat de sensor betrouwbaar moet reageren op een reflectiepaaltje, ongeacht de lichtomstandigheden – zowel bij fel zonlicht als in volledige duisternis. Om de juiste technologie te selecteren, heb ik radar, rood licht, infrarood en ultrasoon met elkaar vergeleken. Hieruit blijkt dat radar de meest geschikte optie is. Radar is echter te complex en te duur, waardoor de keuze valt op infrarood en/of rood licht technologie als de meest efficiënte en haalbare oplossing.

Rood licht (620-750 nm) is zichtbaar en wordt gebruikt in optische sensoren waar visuele controle nuttig is. Infrarood licht (>750 nm) is onzichtbaar, minder gevoelig voor omgevingslicht en beter voor detectie van donkere of matte oppervlakken. Rood licht is ideaal voor nauwkeurige reflectiedetectie, terwijl infrarood beter werkt in omgevingen met veel zichtbaar licht of warmtebronnen. Praktisch gezien zijn beide opties goed. De afstandssensor die nu wordt gebruikt is een roodlichtsensor en er zijn nooit storingen of problemen geweest. Roodlicht is goedkoper dan infrarood en er is eer keuze als het gaat om sensoren die werken op roodlicht. Hierom heb ik gekozen voor roodlicht.

Ik heb gekozen voor een afstandssensor die werkt op basis van het Time-of-Flight (ToF)-principe, waarbij de sensor berekent hoelang een puls onderweg is van uitzending tot detectie. Hierdoor kan ik misschien een sensor kiezen waarbij de maximale detectieafstand kan instellen en valse detecties minimaliseren, een belangrijke eis van de stakeholders. Deze sensoren kunnen nauwkeurig worden afgesteld om alleen objecten binnen een bepaalde afstand te detecteren. Voor optimale meetnauwkeurigheid moeten zowel de sensor als het reflectieobject waterpas worden gemonteerd, zodat de laserstraal optimaal wordt teruggekaatst en meetfouten door verkeerde hoeken worden voorkomen.

Het aanbod aan afstandssensoren is zeer uitgebreid, waardoor een gerichte selectie noodzakelijk was om de keuze eenvoudiger en overzichtelijker te maken. Daarom heb ik een selectie van acht sensoren gemaakt en deze beoordeeld aan de hand van de gestelde eisen:

- Leuze PRK 96K/N-1380-46
- Leuze PRK25C.D1/4P-200-M12
- ifm O5D150
- sick wl250-2p2431
- Pepperl+Fuchs ML100-55/103/115
- Omron E3Z-R81
- Banner Q20FFLPQ
- Keyence PZ-G51N

Uit deze vergelijking komen de Leuze PRK25C.D1/4P-200-M12 en de SICK wl250-2p2431 als beste keuze naar voren. Deze sensoren voldoen bijna aan alle gestelde eisen en hebben bovendien een stevige, robuuste behuizing, wat bijdraagt aan de betrouwbaarheid en duurzaamheid in industriële toepassingen.

Keuze voor de Sick wl250-2p2431

De keuze is uiteindelijk gevallen op de sick wl250-2p2431. Omdat deze sensor een zeer hoge schakelfrequentie van 1 kHz en een zeer korte aanspreekijd van 0,5 ms. Dit betekent dat de sensor in maximaal 1 ms kan reageren en een detectiefrequentie tot 1 kHz kan halen.

De sick wl250-2p2431 vereist een voedingsspanning tussen 10 en 30V DC, wat hoger is dan de 3,3V die de Arduino Nano ESP32-S3 kan leveren. Daarom wordt de sensor rechtstreeks gevoed met 12V vanuit de boordspanning van het voertuig.

Daarnaast moet het digitale uitgangssignaal (NPN) worden aangepast om compatibel te zijn met de 3,3V logica van de Arduino. De uitgangsmodus van de sensor (NPN) moet een geschikte spanningsomzetter of pull-down weerstand worden toegepast om correcte signaalverwerking te garanderen.

De datasheet van de sick wl250-2p2431 is als [bijlage C1](#) toegevoegd.



Afbeelding 4: De sick wl250-2p2431.

Keuze van een externe RTC voor de Arduino Nano ESP32-S3

De Arduino Nano ESP32-S3 heeft een ingebouwde RTC, maar deze is niet bijzonder nauwkeurig voor lange metingen. Daarom is een externe RTC noodzakelijk voor toepassingen waar betrouwbare tijdregistratie vereist is.

Aangezien ik de opleiding Embedded Hardware niet heb gevolgd, is het voor mij geen optie om losse componenten te selecteren en op een PCB te solderen. Daarom heb ik gezocht naar kant-en-klare RTC-modules.

Belangrijke aspecten van een betrouwbare RTC

Uit verschillende wetenschappelijke papers blijkt dat de volgende elementen bijdragen aan een nauwkeurige en betrouwbare RTC:

- 32.768 kHz kristaloscillator met TCXO (Temperature Compensated Crystal Oscillator)
 - Dit betekent dat de frequentie stabiel blijft, zelfs bij temperatuurschommelingen.
 - RTC's zonder TCXO kunnen enkele seconden tot minuten per maand afwijken.
 - Een geïntegreerde oscillator vermindert afwijkingen en jitter ten opzichte van een extern kristal, dat gevoeliger is voor trillingen en temperatuurschommelingen.
- Back-up batterij (V_BAT)
 - Hiermee blijft de tijd behouden bij stroomuitval, waardoor de RTC altijd correct blijft lopen.

Op basis van deze criteria heb ik een selectie gemaakt van geschikte RTC-modules.

Gemaakte selectie

Gezien het brede aanbod aan RTC-modules was een gerichte selectie noodzakelijk. Daarom heb ik vijf RTC's beoordeeld op basis van de gestelde eisen:

- RV-3028-C7
- AB1805 (Abracon RTC)
- NXP PCF2129AT

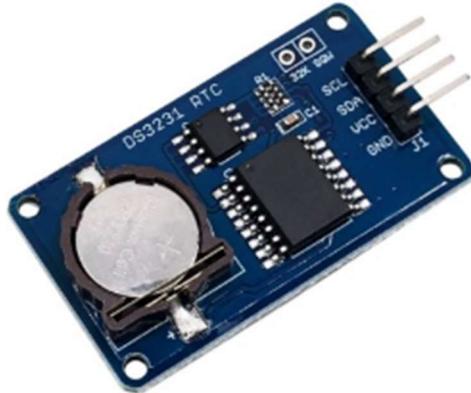
- MAX31341
- DS3231 (Open Smart)

Gekozen RTC: DS3231

Voor een probleemloze integratie en betrouwbare prestaties op de Arduino Nano ESP32-S3 zijn de DS3231 en de AB1805 uitstekende keuzes. Uiteindelijk heb ik gekozen voor de DS3231 (zie afbeelding 5), omdat:

- De nauwkeurigheid ± 2 PPM is, wat neerkomt op minder dan 1 minuut afwijking per jaar.
- De DS3231 volledig wordt ondersteund door de RTClib, die compatibel is met de ESP32-S3.
- Dit de implementatie eenvoudiger maakt en zorgt voor betrouwbare tijdregistratie.

De datasheet van de DS3231 is als [bijlage C4](#) toegevoegd.



Afbeelding 5: De RTC DS3231.

Displaykeuze voor de datalogger

Voor de datalogger wil ik graag een display waarmee een menustructuur kan worden weergegeven met enkele keuzemogelijkheden. Daarnaast moet de gebruiker via het menu kunnen communiceren met de hardware. Verder is het zinvol om op het display de status van het apparaat weer te geven, zoals of het in bedrijf is en of de meting loopt. Ook moet de gebruiker een indicatie krijgen van het aantal gedetecteerde reflectoren en of de metingen zijn geslaagd en opgeslagen.

Verschillen tussen TFT- en OLED-schermen

- TFT (Thin-Film Transistor):
 - Gebruikt in LCD's om individuele pixels aan te sturen.
 - TFT-LCD's vereisen een backlight, wat resulteert in lager contrast en beperktere kijkhoeken.
- OLED (Organic Light-Emitting Diode):
 - Elke pixel produceert zijn eigen licht, wat zorgt voor hoger contrast en bredere kijkhoeken.
 - OLED-schermen zijn dunner en flexibeler dan TFT-LCD's.

Op basis van deze eigenschappen heb ik gekozen voor een OLED-scherm, omdat dit meer mogelijkheden biedt en gebruiksvriendelijker is.

Belangrijke aspecten bij de aanschaf van een OLED-scherm

1. Schermgrootte en resolutie

- Fysieke afmetingen (inch-diagonaal): Voor embedded systemen zijn 0,96" tot 2,42" gangbaar.
- Resolutie (pixels): Hoe hoger de resolutie, hoe scherper het beeld. Typische waarden zijn 128×32, 128×64 of 256×64 pixels.

2. Interface en compatibiliteit

- I²C vs. SPI:
 - I²C: Minder draden nodig (SDA, SCL), eenvoudig te integreren.
 - SPI: Sneller dan I²C, beter geschikt voor snelle updates.
- Voedingsspanning:
 - OLED-schermen werken vaak op 3,3V, maar sommige ondersteunen ook 5V.
 - Bij een 5V-microcontroller is mogelijk een level shifter nodig.

3. Besturingschip (Controller)

OLED-schermen hebben een ingebouwde controllerchip die bepaalt hoe het scherm wordt aangestuurd. Veelgebruikte controllers zijn:

- SSD1306: Veel ondersteuning in Arduino en ESP32-projecten (I²C/SPI).
- SH1106: Vergelijkbaar met SSD1306, maar kan compatibiliteitsproblemen geven.
- SSD1322/SSD1327: Ondersteunt hogere resoluties en grijstinten.

4. Beeldkwaliteit en kleur

- Monochroom vs. kleuren-OLED:
 - Monochrome (wit, blauw, geel) zijn energiezuiniger.
 - Kleuren-OLED's (RGB) bieden betere visuals, maar verbruiken meer stroom.
- Contrast en helderheid:
 - OLED heeft oneindig contrast (zwarte pixels zijn echt uit).
 - Sommige OLED's ondersteunen grijstinten voor meer detail.

5. Energieverbruik

- OLED's verbruiken minder stroom dan TFT-schermen, maar het verbruik hangt af van het beeld:
 - Veel witte pixels → hoger verbruik.
 - Meer zwart → energiezuiniger.
- Voor batterijgevoede projecten is laag stroomverbruik belangrijk.

6. Software-ondersteuning

- Arduino, ESP32 en Raspberry Pi hebben uitstekende bibliotheekondersteuning voor OLED's.
- Controleer of drivers compatibel zijn met de gebruikte controllerchip.

Vergelijking van OLED-schermen

Ik heb drie OLED-displays met elkaar vergeleken:

- ZHONGJINGYUAN 2,42" OLED
- Waveshare 2,42" OLED
- Winstar 2,42" OLED



Afbeelding 6: De ZHONGJINGYUAN 2,42" OLED display White.

Omdat de schermen qua specificaties niet veel van elkaar verschillen en het display voornamelijk functioneel moet zijn, heb ik gekozen voor de ZHONGJINGYUAN 2,42" OLED. Dit scherm biedt voldoende ruimte en helderheid om alle benodigde informatie duidelijk weer te geven. De datasheet van het OLED-scherm is als [bijlage C7](#) toegevoegd.

geheugenmodule

Voor het opslaan van tijdstempels op de Arduino Nano ESP32-S3 is het selecteren van een geschikte (micro)SD-kaartlezer cruciaal. Zie tabel 5 met enkele aanbevolen modules, gevolgd door de criteria.

Aanbevolen microSD-kaartlezer modules:

Module	Interface	Spanningsniveau	Level Shifter	Prijs	Opmerking
DIYables Micro SD Card Adapter	SPI	3.3V/5V	Ja	6.99 voor 5 stuks	Geschikt voor Arduino, ESP32, Raspberry Pi
ANMBEST Micro SD TF Card Adapter	SPI	3.3V/5V	Ja	8.99 voor 10 stuks	Inclusief level conversie chip
Wishiot SD Card Module	SPI	3.3V/5V	Ja	7.98 voor 5 stuks	Inclusief Dupont kabels

Tabel 8: Aanbevolen modules met specificaties.

Belangrijke criteria bij het selecteren van een microSD-kaartlezer module:

1. Interface (Communicatieprotocol):

- De meeste microSD-kaartmodules communiceren via de SPI-interface, wat compatibel is met de Arduino Nano ESP32-S3.

2. Spanningsniveau:

- Controleer of de module werkt op 3.3V, 5V of beide. Aangezien de ESP32-S3 op 3.3V werkt, is compatibiliteit hiermee essentieel.

3. Level Shifter (Spanningsomzetter):

- Een ingebouwde level shifter zorgt voor compatibiliteit tussen de spanningsniveaus van de microcontroller en de (micro)SD-kaart. Dit voorkomt mogelijke schade aan de kaart of de controller.

4. Kwaliteit en Betrouwbaarheid:

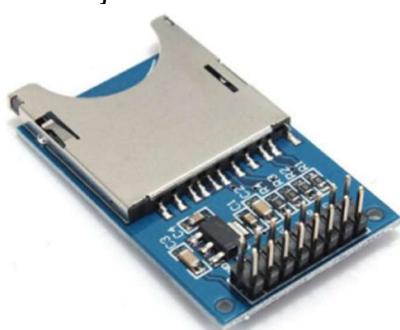
- Kies voor modules met positieve gebruikersrecensies en van gerenommeerde fabrikanten om betrouwbaarheid te waarborgen.

5. Beschikbaarheid van Documentatie en Ondersteuning:

- Modules met goede documentatie en community-ondersteuning kunnen het integratieproces aanzienlijk vereenvoudigen.

Conclusie: Bij het selecteren van een (micro)SD-kaartlezer module voor je Arduino Nano ESP32-S3 is het essentieel om te letten op compatibiliteit met de SPI-interface, het juiste spanningsniveau en de aanwezigheid van een level shifter. De hierboven genoemde modules voldoen allemaal aan deze criteria en zijn allemaal geschikt.

Door mij werd een van deze modules gekozen, zie afbeelding 7.



Afbeelding 7: SD Kaart Adapter Module.

GPS-module.

Belangrijke criteria bij het selecteren van een GPS-module.

Voor een zeer accurate tijdwaarneming en weergave van de actuele rijsnelheid, zijn de volgende eigenschappen van een GPS-module belangrijk:

1. GNSS-ondersteuning (GPS, GLONASS, Galileo, BeiDou)

- GPS (VS), GLONASS (Rusland), Galileo (EU), BeiDou (China) en QZSS (Japan) zijn de belangrijkste satellietsystemen.
- Multiconstellatie-ontvangers (GPS + Galileo + GLONASS + BeiDou) hebben een snellere fix en hogere nauwkeurigheid, vooral in stedelijke gebieden.

Aanbevolen: Een module die minimaal GPS en Galileo ondersteunt voor hoge nauwkeurigheid.

2. Nauwkeurigheid van tijdwaarneming

Voor zeer accurate tijdwaarneming:

- De PPS (Pulse Per Second) output geeft een zeer precieze tijdsreferentie met een nauwkeurigheid tot nanoseconden.
- Sommige geavanceerde modules ondersteunen TCXO (Temperature Compensated Crystal Oscillator) voor betere stabiliteit.

Aanbevolen: Een GPS-module met PPS-output en eventueel TCXO voor minimale drift.

3. Updatefrequentie (Refresh Rate)

- De standaard GPS-updatefrequentie is 1 Hz (1 meting per seconde), wat geschikt is voor veel toepassingen.
- Hogere updatefrequenties (5 Hz, 10 Hz, 25 Hz) verbeteren de weergave van actuele rijsnelheid, vooral bij snelle bewegingen (bv. in een voertuig).

Aanbevolen: Minimaal 5 Hz updatefrequentie voor realtime snelheid.

4. Gevoeligheid en Fix-tijd

- Gevoeligheid bepaalt hoe goed de module satellietsignalen kan ontvangen, zelfs bij slecht weer of in stedelijke gebieden.
- Expressie in dBm, bijvoorbeeld:
 - -165 dBm (zeer goed, zelfs binnenshuis).
 - -145 dBm (basisniveau, goed voor buitengebruik).
- TTFF (Time To First Fix) bepaalt hoe snel de module een eerste fix verkrijgt bij inschakeling.

Aanbevolen: -160 dBm of beter + snelle TTFF (<10 sec cold start, <1 sec warm start).

5. Interface (Communicatie met de Arduino Nano ESP32-S3)

De Arduino Nano ESP32-S3 ondersteunt de volgende interfaces:

- UART (Seriële communicatie, TX/RX): Meest gebruikelijk en compatibel met de meeste GPS-modules.
- I²C: Minder gebruikelijk voor GPS, maar sommige modules ondersteunen het.
- SPI: Sneller, maar minder frequent gebruikt bij GPS.
- USB: Meestal voor PC-gebruik, niet relevant voor ESP32-S3.

Aanbevolen: UART (TX/RX) GPS-module voor eenvoudige integratie.

6. Extra functionaliteiten

RTK (Real-Time Kinematics) voor cm-nauwkeurigheid?

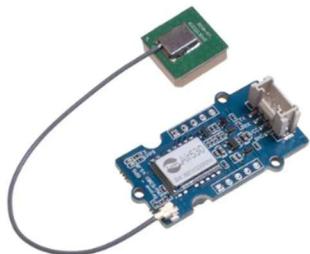
- Standaard GPS heeft een nauwkeurigheid van 1-3 meter.
- RTK-GPS (zoals u-blox ZED-F9P) kan nauwkeurigheid van ~1 cm bereiken, maar vereist een extra referentiestation.

Datalogging

- Sommige modules hebben een interne opslag om GPS-data direct op te slaan zonder SD-kaart.

Aanbevolen: RTK alleen als extreme nauwkeurigheid vereist is.

Door mij werden vijf verschillende GPS-ontvangers geselecteerd en in een tabel gezet. Van deze werd de: "Seeed Studio Grove GNSS-GPS Module - Air530", gekozen voor de datalogger. Omdat deze sensor relatief gezien het goedkoopst is en de meeste nauwkeurigheid geeft.



Afbeelding 8: Seeed Studio Grove GNSS-GPS Module - Air530

7.2 Softwarearchitectuur

Tweede subonderzoeks vraag:

Welke software-architectuur zorgt voor een efficiënte verwerking en opslag van tijdstempels?

Om een efficiënte software-architectuur te ontwerpen voor het verwerken en opslaan van tijdstempels in een datalogger, moeten we rekening houden met:

- Real-time verwerking van binnenkomende data
- Betrouwbare opslag van tijdstempels
- Energiezuinigheid (indien op een embedded systeem)
- Data-integriteit bij herstarts of stroomuitval
- Efficiënte opslag en uitlezing

Hieronder beschrijf ik een optimale software-architectuur die voldoet aan deze eisen.

1. Algemene Architectuurindeling

De software-architectuur kan worden opgesplitst in vier lagen:

1. Sensorinterface – Verwerking van binnenkomende data van detectieapparatuur (bijv. afstandssensor, GPS, RTC).
2. Tijdstempelbeheer – Correcte tijdregistratie en berekening van tussenliggende tijden.
3. Buffer- en opslagbeheer – Tijdelijke en permanente opslag van data.
4. Uitvoer en uitlezing – Weergave op een scherm of overdracht naar een pc/server.

2. Software-Architectuur in lagen

Sensorinterface

Doel: Data betrouwbaar ontvangen van de sensoren en tijdig verwerken.

- Interrupt-driven verwerking: Detectiepaal-data komt binnen via interrupt-handler.
- FIFO-buffering: Binnenkomende metingen worden eerst in een First In, First Out (FIFO) buffer geplaatst (voor toekomstige optie).
- Tijdstempel toewijzing: Zodra een voertuig wordt gedetecteerd, wordt een timestamp uit de RTC opgehaald.

Technieken:

- Interrupt Service Routines (ISR) voor snelle verwerking van detecties
- FIFO buffer om snelle pieken in metingen op te vangen zonder verlies (als toekomstige optie)

Tijdstempelbeheer

Doel: Correcte en nauwkeurige tijdstempels genereren en verwerken.

- Real-Time Clock (RTC) als tijdsbron (bijv. DS3231 of ingebouwde RTC in de ESP32).
- GPS als alternatieve tijdssynchronisatie, indien vereist.
- Berekening van tijd tussen detecties:
 - Zodra een voertuig een detectiepaal passeert, wordt de vorige timestamp opgezocht in de buffer.
 - De tijdsduur tussen detectiepunten wordt berekend en opgeslagen.

Technieken:

- RTC-bibliotheek voor nauwkeurige tijdstempels
- Tijdsdifferentiatie om de exacte reistijd te berekenen
- Failsafe-mechanisme: Als de RTC uitvalt, schakelt het systeem over naar GPS-tijd (als toekomstige optie)

Buffer- en opslagbeheer

Doel: Betrouwbare opslag van tijdstempels en tussenliggende tijdsintervallen, met bescherming tegen stroomuitval.

- Dubbele buffering:
 - Tijdstempels worden eerst tijdelijk in RAM opgeslagen (kleine buffer).
 - Na een bepaalde actie (bijv. wanneer een flag is gezet) worden ze permanent naar de SD-kaart geschreven.
- Periodieke opslag per rit:
 - Elke rit wordt in een apart opgeslagen in één CSV-bestand.
- Fail-safe mechanisme (**toekomstige uitbreiding**):
 - Flush()-functie na elke batch om dataverlies te voorkomen bij stroomuitval.
 - FRAM als optionele extra buffer, zodat gegevens bewaard blijven bij een crash.

Technieken:

- Ringbuffer voor RAM-opslag (geheugenefficiënt en snel)
- SD-kaart opslag met CSV-formaat (compatibel en makkelijk uit te lezen)

Uitvoer en uitlezing

Doel: Realtime monitoring en gemakkelijke data-uitlezing.

- OLED-display voor userinfo.
- Bestandsformaat keuze:
 - CSV-formaat voor compatibiliteit met Excel en Python-analyse.
 - JSON als alternatief voor draadloze transmissie naar een server (Toekomst).

Technieken:

- CSV-bestanden schrijven via SD-kaart bibliotheek
- Optioneel: WiFi/Bluetooth-integratie voor live monitoring
- Optioneel: OLED-update via timers (geen blocking delay())

3. Software-Flowchart

Hier is een overzicht van hoe de software zou kunnen werken:

1. **Start**
 - ↳ RTC controleren en synchroniseren
 - ↳ SD-kaart checken en openen
2. **Data-acquisitie**
 - ↳ Wachten op interrupt van detectiesensor
 - ↳ Tijdstempel ophalen van RTC (DS3132 of van de ESP32-S3 zelf)
 - ↳ Data opslaan in ISR RAM-buffer
3. **Gegevensverwerking**
 - ↳ Tijdsverschil berekenen tussen detecties
 - ↳ Als de flag is gezet buffer overschrijven naar SD-kaart

4. **Data-opslag**
 - ↳ Schrijven naar CSV-bestand (meting.csv)
 - ↳ Flush() oproepen (in toekomstige versie) om SD-kaart te updaten
5. **Einde rit**
 - ↳ OLED-display updaten met beëindigen rit
 - ↳ Vervolgens tonen van het aantal detecties
 - ↳ Weergeven indien de schrijfactie naar de sd-kaart succesvol was
 - ↳ Systeem resetten voor nieuwe rit

4. Efficiëntie-overwegingen

Om de architectuur zo efficiënt mogelijk te maken:

1. Gebruik interrupt-gebaseerde detectie in plaats van polling → Bespaart CPU-tijd en energie.
2. Optimaliseer SD-kaartgebruik:
 - o Gebruik grotere sector-schrijvingen om flash-slijtage te minimaliseren.
 - o Schrijf data in batches i.p.v. elke meting afzonderlijk.

5. Conclusie & Aanbevolen Software-Architectuur

Voor een efficiënte verwerking en opslag van tijdstempels in een datalogger is de beste software-architectuur:

- Interrupt-gebaseerde sensorinterface voor snelle detectie
- RTC-gestuurde tijdstempels met (eventuele) backup via GPS
- Batch-opslag naar SD-kaart om levensduur te verlengen
- CSV-formaat voor makkelijke uitlezing en analyse

Met deze architectuur krijg je:

- Nauwkeurige en efficiënte tijdstempellogging
- Minimale SD-kaartslijtage door batch-opslaan
- Data-integriteit bij stroomuitval
- Snelle verwerking en uitlezing via OLED of PC

Voor een embedded datalogger zijn deze de meest geschikte programmeertalen:

1. Programmeertalen en Embedded Frameworks voor Microcontrollers

Taal / Framework	Voordelen	Nadelen	Geschikt?
C++ (Arduino / ESP-IDF)	Direkte hardware-aansturing, efficiënt geheugenbeheer, brede ondersteuning voor microcontrollers	Minder abstract, vereist meer kennis	Ja, beste keuze met name voor ESP32/Arduino
C (Bare-metal of HAL)	Laagste overhead, volledige controle, breed ondersteund	Complex geheugenbeheer, handmatige optimalisatie nodig	Ja, voor maximale efficiëntie
MicroPython	Eenvoudig en snel te ontwikkelen, ideaal voor beginners	Langzamer dan C++, hogere geheugenbelasting	Kan, maar minder efficiënt
CircuitPython (Adafruit)	Gebruiksvriendelijker dan MicroPython, brede bibliotheekondersteuning	Minder geschikt voor real-time toepassingen	Kan, maar niet optimaal voor snelle logging

Taal / Framework	Voordelen	Nadelen	Geschikt?
Rust (voor ESP32, STM32, RISC-V, etc.)	Zeer veilig, geheugenveiligheid zonder garbage collector, efficiënt	Minder ondersteuning, complexer dan C++	Te experimenteel
Go (TinyGo)	Compact, efficiënt, goede ondersteuning voor RISC-V en ARM	Weinig ingebouwde bibliotheken voor microcontrollers	Niet aanbevolen
Embedded Java (LeJOS, JVM op ARM/ESP32)	Objectgeoriënteerd, Java-ecosysteem	Trager, hogere geheugenvereisten	Te zwaar voor embedded toepassingen
Embedded C# (.NET nanoFramework)	Gebruiksvriendelijk, moderne programmeerstijl	Beperkte ondersteuning, veel overhead	Niet efficiënt voor microcontrollers
Forth (Mecrisp, eForth, etc.)	Zeer compact, direct interactief testen mogelijk	Moeilijk te leren, beperkte community	Te niche voor moderne projecten
Lua (eLua, NodeMCU)	Compact, scripting mogelijk op ESP32 en STM32	Langzamer dan C++, niet real-time geschikt	Niet efficiënt genoeg
Assembly (ARM, AVR, RISC-V, etc.)	Maximale snelheid en controle	Zeer complex, niet schaalbaar	Niet praktisch voor grote projecten

Aanbevolen keuze: C/C++ met Arduino IDE, PlatformIO of ESP-IDF

- Arduino IDE is makkelijk voor snelle prototyping.
- PlatformIO is krachtiger en ondersteunt meer features.
- ESP-IDF is specifiek voor de ESP32 en biedt maximale controle en efficiëntie.

Conclusie:

Een efficiënte software-architectuur voor de verwerking en opslag van tijdstempels in een embedded datalogger bestaat uit vier lagen: sensorinterface, tijdstempelbeheer, opslagbeheer en data-uitlezing. De sensorinterface maakt gebruik van interrupt-gebaseerde detectie (UART/I²C) voor snelle en energieuze verwerking van binnenkomende data. In de tijdstempellaag worden RTC (bijv. DS3231) of GPS-data gebruikt om nauwkeurige tijdsregistraties te genereren en berekeningen uit te voeren met millis() of difftime(). De opslaglaag bevindt zich in RAM voor tijdelijke opslag en een SD-kaart met CSV-bestanden (via SdFat.h) voor permanente opslag. Tot slot verzorgt de uitleeslaag een OLED-scherm (Adafruit_SSD1306) voor real-time monitoring, seriële uitvoer via UART en optioneel WiFi/Bluetooth voor draadloze data-overdracht. Door gebruik te maken van C++ met Arduino IDE of PlatformIO, en indien nodig FreeRTOS voor multitasking, biedt deze architectuur een schaalbare, robuuste en energie-efficiënte oplossing voor betrouwbare tijdstempellogging.

7.3 Prestatie-eisen

Derde subonderzoeks vraag:

Wat zijn de prestatie-eisen voor een nauwkeurige tijdsvergelijking?

Voor een nauwkeurige tijdsvergelijking in een embedded datalogger zijn verschillende prestatie-eisen van belang, afhankelijk van de vereiste precisie en de gebruikte hardware. Hieronder bespreek ik de belangrijkste factoren en eisen.

1. Resolutie en Nauwkeurigheid van de Tijdbron

Om betrouwbare tijdstempels te genereren en te vergelijken, moet de tijdsbron nauwkeurig en stabiel zijn.

Tijdsbron	Resolutie	Nauwkeurigheid	Geschikt voor?
Interne millis() / micros() (ESP32, Arduino)	1 ms / 1 μ s	$\pm 0.1\%$ (afhankelijk van kristal)	Redelijke precisie voor korte metingen
RTC-module (DS3231, PCF8563)	1 s / 1 ms	$\pm 2 \text{ ppm}$ ($\pm 0.17 \text{ s}$ per dag)	Goede stabiliteit voor lange termijn
GPS PPS (Pulse Per Second)	1 μ s	$\pm 30 \text{ ns}$ (zeer nauwkeurig)	Hoge precisie en synchronisatie

Aanbevolen keuze:

Voor milliseconde-nauwkeurigheid is een RTC zoals de DS3231 geschikt, maar als tijdsvergelijkingen met hoge precisie nodig is, dan moet er gebruik gemaakt worden van GPS PPS. Maar voor korte metingen zoals validatie van tijdwaarnemingen van VRI's volstaat de RTC.

2. Tijdregistratie en Vergelijkingsalgoritme

- De microcontroller moet tijdstempels zonder jitter of vertraging kunnen vastleggen.
- Gebruik interrupts in plaats van polling om vertragingen te minimaliseren.
- Referentie-tijdstempels synchroniseren (bijv. RTC bijstellen met GPS).

Aanbevolen aanpak:

- Bij elke detectie een interrupt genereren.
- Timestamp opslaan met millis() of een RTC.
- Vergelijking uitvoeren met een 64-bit integer om overflows te voorkomen.
- Gebruik een filter (bijv. moving average) als er variatie in meetwaarden is.

3. Opslageisen en Gegevensverwerking

- Opslag van tijdstempels in RAM moet efficiënt gebeuren.
- Tijdsvergelijking moet met lage CPU-belasting worden uitgevoerd.
- Voor lange termijn logging (RTC): registreer tijd als YYYY-MM-DD HH:MM:SS.mmm.

Optimalisatie-tips:

- RAM gebruiken om recente tijdstempels snel te vergelijken (toekomst).
- Batch-schrijfoperaties naar de SD-kaart om de prestaties te verbeteren (toekomst).
- Bijlage van CRC-checksums voor data-integriteit (toekomst).

5. Maximale Verwerkingsvertraging

Stap	Max. Toegestane Vertraging	Optimalisatie
Interrupt detectie	< 10 µs	Gebruik hardware interrupts
Timestamp vastleggen	< 50 µs	Gebruik micros() of hardware timers
Tijdsverschil berekenen	< 500 µs	Gebruik integer wiskunde (geen floating point)
Schrijven naar SD-kaart	< 10 ms	Gebruik een buffer en batch-schrijven

Voor kritische toepassingen:

- Directe SD-kaart schrijfacties vermijden (gebruik een buffer).
- Hardware-timers in de ESP32 of DS3132 gebruiken voor grote nauwkeurigheid.

Conclusie: Eisen voor een nauwkeurige tijdsvergelijking

1. Gebruik de RTC van de Arduino Nano ESP32-S3 voor milliseconde-logging.
2. Gebruik hardware interrupts voor directe tijdstempelregistratie zonder vertraging.
3. Sla tijdstempels op in CSV-formaat (milliseconden).
4. Minimaliseer opslagvertragingen door een buffer en batch-schrijfoperaties te gebruiken (toekomstige verbetering).

7.4 Opslag en verwerking

Vierde subonderzoeks vraag:

Welke opslag- en verwerkingsmogelijkheden zijn nodig voor logging?

Om deze vraag te beantwoorden moet ik eerst bepalen met welke factoren er rekening gehouden dient te worden:

Dataformaat: Hoeveel bytes per logregel?

Er worden logregels geschreven die de volgende parameters bevatten.

- Detectienummer (bijv. 4 cijfers) → 4 bytes
- Tijdstempel (dd/mm/yyyy hh:mm:ss) → 19 bytes
- Tijd tussen detecties (ss/sss, bijv. 12/3456) → 7 bytes
- Scheidingssymbolen (komma's, spaties, newline, etc.) → 5 à 6 bytes
- Maximaal per logregel: ~35 tot 40 bytes

Ik verwacht tussen de 60 en 100 logregels per validatierit.

Totaal opslaggebruik: 4 kB maximaal per validatierit.

Conclusie: Niet extreem veel dataopslag nodig.

Schrijfsnelheid: Hoe vaak per seconde log je?

Hooguit 2 keer per seconde, maar alleen bij detectie van een detectiepunt.

Schrijfsnelheid: Dit is laag genoeg voor een SD-kaart of FRAM zonder prestatieproblemen.

Conclusie: iedere SD-kaart of FRAM kan dit aan.

Gegevensopslagduur: Hoe lang moet de data bewaard blijven?

Tot na de validatie, dus een paar uur of dagen.

Het totaal blijft onder de 4 kB per project, wat makkelijk op een SD-kaart, EEPROM of FRAM past.

Conclusie: Opslagcapaciteit is geen beperking.

Betrouwbaarheid: Moet het bestand zijn tegen herstarts of stroomuitval?

Ik wil voorkomen dat alle data verloren gaat als er tijdens een rit iets misgaat.

Opties:

1. Na elke detectie de logregel direct naar de SD-kaart schrijven.
 - Dit voorkomt dat data in RAM verloren gaat bij een crash.
2. Gebruik een tijdelijke buffer (FRAM of EEPROM)
 - Eerst naar FRAM (bijv. 8 kB I²C of SPI FRAM-chip) loggen en aan het einde van een rit overzetten naar de SD-kaart.
 - Voorkomt dat de SD-kaart telkens wordt beschreven, dit verlengt de levensduur.
3. Elke rit apart opslaan in één bestand
 - Mocht er iets misgaan, dan verlies je hooguit de data van één rit.

Conclusie beste aanpak:

- Optie 1: Direct wegschrijven naar SD-kaart met flush().
- Optie 3: Elke rit in een apart bestand.

Levensduur opslagmedium: Hoe vaak kan het medium overschreven worden?

Dit hangt af van het medium:

Opslagmethode	Schrijflimiet (per sector)	Geschikte optie?
SD-kaart (FAT32)	~100.000 schrijfbewerkingen per sector	Ja, als je slim schrijft (buffers & niet constant overschrijven)
FRAM (I²C/SPI)	Onbeperkte schrijfcycli	Ideaal voor bufferdata
EEPROM (intern of extern)	~1.000.000 schrijfbewerkingen	Kan, maar onnodig
Flash (ESP32-S3 NVS of LittleFS)	10.000 - 100.000 schrijfbewerkingen	Niet aanbevolen, slijtagegevoelig

Ongeveer 60-100 regels per project, dus zelfs een SD-kaart zou pas na miljoenen ritten slijten.

Conclusie:

- Een SD-kaart is prima, zolang je niet constant herschrijft.
- FRAM als tussenbuffer is een mooie extra, maar niet per se nodig als je een SD-kaart gebruikt met flush().

7.5 Validatie en testplan

Vijfde subonderzoeksraag:

Hoe kan de datalogger worden getest en gevalideerd op nauwkeurigheid?

Om dit proces eenvoudig, maar effectief te houden, kan de volgende aanpak worden gehanteerd:

1. Testopstelling en Referentiepunten

Om een betrouwbare vergelijking te maken, heb je een duidelijk referentiepunt nodig. Hier zijn enkele opties:

Optie 1: GPS als referentiepunt (eenvoudig en praktisch)

- Gebruik een GPS-datalogger als externe referentie.
- GPS geeft een nauwkeurig UTC-tijdsignaal dat je kunt vergelijken met je datalogger.
- Vergelijk de tijdstempels van detectiemomenten tussen je logger en de GPS-datalogger.

Optie 2: Vergelijk de datalogger met de andere datalogger op één auto

- Monteer de DAQ en de Mov1Log BM22 op de auto en rijdt referentieritten.
- Dit kan worden gebruikt als referentie om te bepalen wanneer een voertuig een detectiepaal passeert en wat de afwijkingen zijn.

Optie 3: Externe meetapparatuur (hogere precisie, maar complexer)

- Gebruik een hogesnelheidscamera met een tijdstempel-overlay.
- Hiermee kun je de exacte passage van een voertuig bij een detectiepunt handmatig valideren.

Aanbevolen keuze: Optie 2 voor een eenvoudige, robuuste test.

Testprocedure voor Validatie

Om consistentie en betrouwbaarheid te waarborgen, deze stappen:

1. Setup van de testauto

- Monteer twee bestaande dataloggers + eigen datalogger op een stabiele positie.
- Zorg ervoor dat alle apparaten op hetzelfde voedingspunt zitten om timingafwijkingen te minimaliseren.
- Synchroniseer vooraf de klokken indien nodig (RTC of GPS-tijd).
- Leg alle posities vooraf vast.

2. Uitvoeren van referentieritten

- Rij een vaste route met meerdere detectiepunten.
- Log de exacte tijdstempels van elke detectie in CSV-formaat.
- Herhaal de ritten minimaal 5 keer om meetfouten te detecteren.

3. Data-analyse en vergelijking

- Exporteer de logbestanden van alle dataloggers.
- Bereken de gemiddelde afwijking (in ms of μ s) en bepaal de maximale foutmarge.

Validatiecriteria

De tijdwaarneming van de datalogger wordt als betrouwbaar beschouwd als deze binnen een acceptabele foutmarge blijft ten opzichte van de referentiedataloggers. De afwijking wordt vastgesteld door een statistische vergelijking van de tijdstempels, waarbij consistentie tussen meerdere meetmomenten en ritten wordt geanalyseerd. Op basis van de technische specificaties van de tijdbron (RTC of GPS) wordt de maximale toelaatbare afwijking bepaald. Indien de afwijking binnen deze marges valt, wordt de datalogger als nauwkeurig gevalideerd. De afwijkingen worden nader bepaald, die hiervoor de testen.

Bijlage C1 Datasheet Sick WL250-2P2431

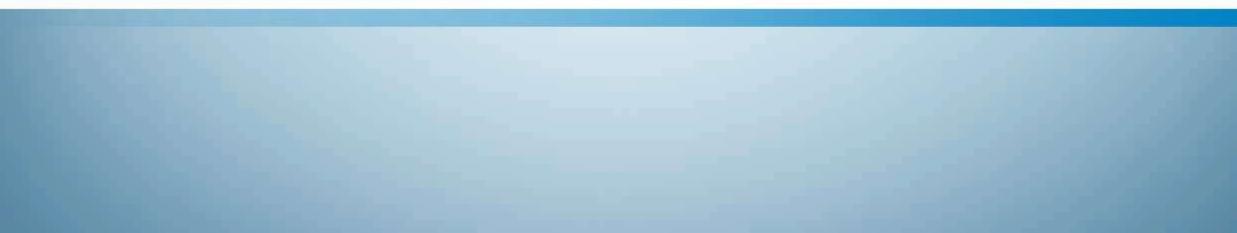


Data sheet

WL250-2P2431
W250-2

PHOTOELECTRIC SENSORS

SICK
Sensor Intelligence.



WL250-2P2431 | W250-2

PHOTOELECTRIC SENSORS



Ordering information

Type	Part no.
WL250-2P2431	6044699

Included in delivery: BEF-W250 (1), P250 (1)

Other models and accessories → www.sick.com/W250-2

Illustration may differ



Detailed technical data

Features

Functional principle	Photoelectric retro-reflective sensor
Functional principle detail	With minimum distance to reflector (dual lens system)
Dimensions (W x H x D)	20 mm x 65 mm x 43.9 mm
Housing design (light emission)	Rectangular
Sensing range max.	0.01 m ... 15 m ¹⁾ 0.01 m ... 12 m ²⁾
Sensing range	0.01 m ... 13 m ¹⁾ 0.01 m ... 11 m ²⁾
Focus	Approx. 2°
Type of light	Visible red light
Light source	LED ³⁾
Light spot size (distance)	Ø 260 mm (8 m)
Angle of dispersion	Approx. 2°
Adjustment	Potentiometer, 2 turns ⁴⁾

¹⁾ Reflector PL80A.

²⁾ Reflector P250.

³⁾ Average service life: 100,000 h at T_U = +25 °C.

⁴⁾ With position indicator.

Mechanics/electronics

Supply voltage U_B	10 V DC ... 30 V DC ¹⁾
Ripple	< 5 V _{pp} ²⁾
Current consumption	20 mA ³⁾
Switching output	PNP
Switching mode	Light/dark switching
Switching mode selector	Selectable via L/D control cable
Output current $I_{max.}$	≤ 100 mA
Response time	≤ 0.5 ms ⁴⁾
Switching frequency	1,000 Hz ⁵⁾
Connection type	Male connector M12, 4-pin ⁶⁾
Circuit protection	A ⁷⁾ B ⁸⁾ C ⁹⁾ D ¹⁰⁾
Protection class	III
Weight	40 g
Polarisation filter	✓
Housing material	Plastic, ABS
Optics material	Plastic, PMMA
Enclosure rating	IP67
Items supplied	Reflector P250, BEF-W250 mounting bracket
Electromagnetic compatibility (EMC)	EN 60947-5-2 ¹¹⁾
Ambient operating temperature	-25 °C ... +55 °C
Ambient temperature, storage	-40 °C ... +70 °C
UL File No.	NRKH2.E300503 & NRKH8.E300503

¹⁾ Limit values when operated in short-circuit protected network: max. 8 A.²⁾ May not fall below or exceed U_V tolerances.³⁾ Without load.⁴⁾ Signal transit time with resistive load.⁵⁾ With light/dark ratio 1:1.⁶⁾ Rotatable through 90 °.⁷⁾ A = V_S connections reverse-polarity protected.⁸⁾ B = inputs and output reverse-polarity protected.⁹⁾ C = interference suppression.¹⁰⁾ D = outputs overcurrent and short-circuit protected.¹¹⁾ The AC/DC devices comply with the Radio Safety Requirements for the industrial sector (Radio Safety Class A). They may cause radio interference if used in a residential area.

Safety-related parameters

MTTF_D	1,832 years
DC_{avg}	0 %

Classifications

ECLASS 5.0	27270902
ECLASS 5.1.4	27270902

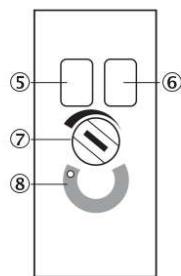
WL250-2P2431 | W250-2

PHOTOELECTRIC SENSORS

ECLASS 6.0	27270902
ECLASS 6.2	27270902
ECLASS 7.0	27270902
ECLASS 8.0	27270902
ECLASS 8.1	27270902
ECLASS 9.0	27270902
ECLASS 10.0	27270902
ECLASS 11.0	27270902
ECLASS 12.0	27270902
ETIM 5.0	EC002717
ETIM 6.0	EC002717
ETIM 7.0	EC002717
ETIM 8.0	EC002717
UNSPSC 16.0901	39121528

Adjustments

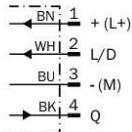
WL250-2



- ⑤ LED indicator green: Stability indicator
- ⑥ LED indicator yellow: Status of received light beam
- ⑦ Sensitivity control: potentiometer
- ⑧ Position indicator for sensitivity setting (270°)

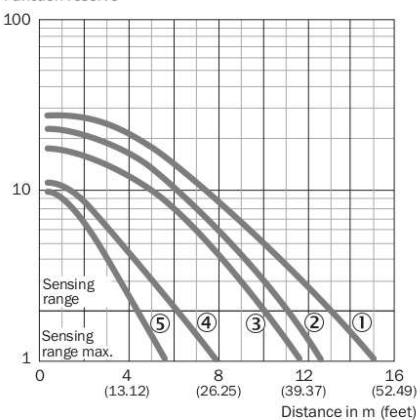
Connection diagram

Cd-087



Characteristic curve

Function reserve



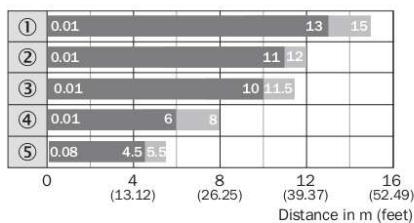
① Reflector PL80A, C110A

② Reflector P250

③ Reflector PL50A, PL40A, PL30A, PL31A

④ Reflector PL20A

⑤ Diamond Grade reflective tape, 100 mm x 100 mm

Sensing range diagram

① Sensing range

① Sensing range max.

① Reflector PL80A, C110A

② Reflector P250

③ Reflector PL50A, PL40A, PL30A, PL31A

④ Reflector PL20A

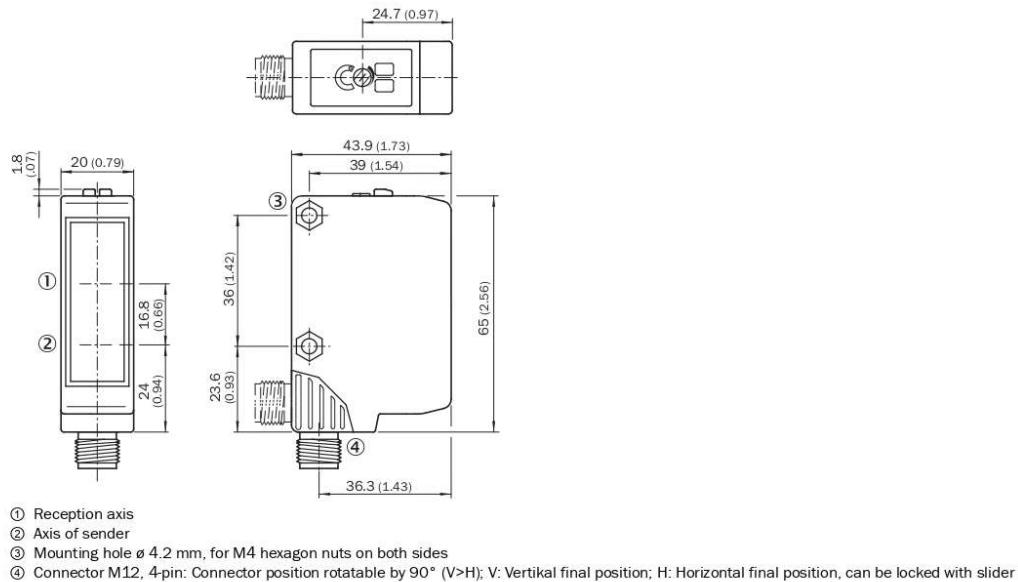
⑤ Diamond Grade reflective tape, 100 mm x 100 mm

WL250-2P2431 | W250-2

PHOTOELECTRIC SENSORS

Dimensional drawing (Dimensions in mm (inch))

WL250-2, DC, connector



Recommended accessories

Other models and accessories → www.sick.com/W250-2

	Brief description	Type	Part no.
Mounting brackets and plates			
	<ul style="list-style-type: none"> Description: Universal mounting bracket for reflectors Dimensions (W x H x L): 85 mm x 90 mm x 35 mm Material: Steel Details: Steel, zinc coated Suitable for: C110A, P250, PL20, PL30A, PL40A, PL80A 	BEF-WN-REFX	2064574
Others			
	<ul style="list-style-type: none"> Description: Rectangular, screw connection Dimensions: 51 mm 61 mm Ambient operating temperature: -30 °C ... +65 °C 	P250	5304812
	<ul style="list-style-type: none"> Connection type head A: Female connector, M12, 4-pin, straight, A-coded Connection type head B: Flying leads Signal type: Sensor/actuator cable Cable: 5 m, 4-wire, PVC Description: Sensor/actuator cable, unshielded Application: Zones with chemicals, Uncontaminated zones 	YF2A14-050VB3XLEAX	2096235
	<ul style="list-style-type: none"> Connection type head A: Male connector, M12, 4-pin, straight, A-coded Description: Unshielded Connection systems: Screw-type terminals Permitted cross-section: ≤ 0.75 mm² 	STE-1204-G	6009932

SICK AT A GLANCE

SICK is one of the leading manufacturers of intelligent sensors and sensor solutions for industrial applications. A unique range of products and services creates the perfect basis for controlling processes securely and efficiently, protecting individuals from accidents and preventing damage to the environment.

We have extensive experience in a wide range of industries and understand their processes and requirements. With intelligent sensors, we can deliver exactly what our customers need. In application centers in Europe, Asia and North America, system solutions are tested and optimized in accordance with customer specifications. All this makes us a reliable supplier and development partner.

Comprehensive services complete our offering: SICK LifeTime Services provide support throughout the machine life cycle and ensure safety and productivity.

For us, that is "Sensor Intelligence."

WORLDWIDE PRESENCE:

Contacts and other locations www.sick.com

SICK
Sensor Intelligence.

SICK AG | Waldkirch | Germany | www.sick.com

Bijlage C2 Datasheet OLED-Scherm

General	
Brand	ZHONGJINGYUAN
Features	Display
General physical appearance	
Main color	Blue
Weight [g]	23
Dimension X [mm]	72
Dimension Y [mm]	43
Mounting options	Mounting hole(s)
Diameter mounting hole(s) [mm]	2.8
Form factor	Module (general)
General electrical properties	
Minimum supply voltage [V DC]	3.3
Maximum supply voltage [V DC]	3.3
Minimum IO-pin input voltage [V]	3.3
Maximum IO-pin input voltage [V]	3.3

General	
IO-pin output voltage [V]	3.3
Communication	
Hardware interface(s)	I2C
Default I2C address	0x3C
Adjustable I2C address	No
Built-in pull-up resistor(s) [kΩ]	4.7
Display	
Type	OLED
Driver	SSD1309
Screen diagonal [inch]	2.42
Number of pixels long side	128
Number of pixels short side	64
Display shape	Rectangular
Pixel colors	White

General**Connectors**

Power supply connector(s)	Pin header male (2.54mm)
---------------------------	--------------------------

IO-connector(s)	Pin header male (2.54mm)
-----------------	--------------------------

Package contents

Package contents	1x Display module
------------------	-------------------

Bijlage C3 Datasheet SD-kaartmodule

SD Card Adapter Module 3.3V-5V



An SD card adapter that works with SPI. The module works with a 3.5V or 5V supply and needs to be controlled with 3.3V signals.

Pin	Description
(2x) GND	Ground, min
+3.3	Direct +3.3V supply*
+5	+5V supply that is converted to +3.3V*
CS	3.3V CS signal
MOSI	3.3V MOSI signal
SCK	3.3V SCK signal
MISO	3.3V MISO signal

* You may never use the +3.3V and +5V connected at the same time, just one of the two.

Bijlage C4 Datasheet RTC DS3231

DS3231

**Extremely Accurate I²C-Integrated
RTC/TCXO/Crystal**

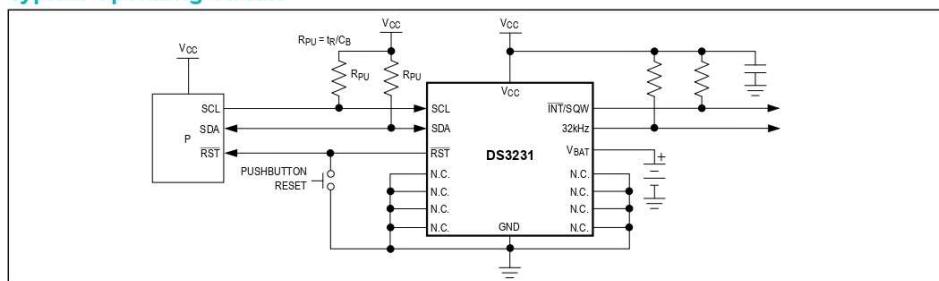
General Description

The DS3231 is a low-cost, extremely accurate I²C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I²C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of V_{CC} to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the RST pin is monitored as a pushbutton input for generating a µP reset.

Typical Operating Circuit



Underwriters Laboratories is a registered certification mark of Underwriters Laboratories Inc.

Benefits and Features

- Highly Accurate RTC Completely Manages All Timekeeping Functions
 - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
 - Accuracy ±2ppm from 0°C to +40°C
 - Accuracy ±3.5ppm from -40°C to +85°C
 - Digital Temp Sensor Output: ±3°C Accuracy
 - Register for Aging Trim
 - RST Output/Pushbutton Reset Debounce Input
 - Two Time-of-Day Alarms
 - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
 - Fast (400kHz) I²C Interface
- Battery-Backup Input for Continuous Timekeeping
 - Low Power Operation Extends Battery-Backup Run Time
 - 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to +70°C) and Industrial (-40°C to +85°C)
- Underwriters Laboratories® (UL) Recognized

Applications

- Servers
- Telematics
- Utility Power Meters
- GPS

Ordering Information and Pin Configuration appear at end of data sheet.



19-5170; Rev 10, 3/15

Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground	-0.3V to +6.0V	Junction Temperature.....	+125°C
Junction-to-Ambient Thermal Resistance (θ_{JA}) (Note 1)	73°C/W	Storage Temperature Range.....	-40°C to +85°C
Junction-to-Case Thermal Resistance (θ_{JC}) (Note 1)	23°C/W	Lead Temperature (soldering, 10s)	+260°C
Operating Temperature Range		Soldering Temperature (reflow, 2 times max)	+260°C
DS3231S	0°C to +70°C	(see the <i>Handling, PCB Layout, and Assembly</i> section)	
DS3231SN.....	-40°C to +85°C		

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to www.maximintegrated.com/thermal-tutorial.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Recommended Operating Conditions

($T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		2.3	3.3	5.5	V
	V_{BAT}		2.3	3.0	5.5	V
Logic 1 Input SDA, SCL	V_{IH}		0.7 x V_{CC}	$V_{CC} +$ 0.3		V
Logic 0 Input SDA, SCL	V_{IL}		-0.3	0.3 x V_{CC}		V

Electrical Characteristics

($V_{CC} = 2.3V$ to $5.5V$, V_{CC} = Active Supply (see Table 1), $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Typical values are at $V_{CC} = 3.3V$, $V_{BAT} = 3.0V$, and $T_A = +25^\circ C$, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Supply Current	I_{CCA}	(Notes 4, 5)	$V_{CC} = 3.63V$	200		μA
			$V_{CC} = 5.5V$	300		
Standby Supply Current	I_{CCS}	I^2C bus inactive, 32kHz output on, SQW output off (Note 5)	$V_{CC} = 3.63V$	110		μA
			$V_{CC} = 5.5V$	170		
Temperature Conversion Current	$I_{CCSConv}$	I^2C bus inactive, 32kHz output on, SQW output off	$V_{CC} = 3.63V$	575		μA
			$V_{CC} = 5.5V$	650		
Power-Fail Voltage	V_{PF}		2.45	2.575	2.70	V
Logic 0 Output, 32kHz, INT/SQW, SDA	V_{OL}	$I_{OL} = 3mA$		0.4		V
Logic 0 Output, RST	V_{OL}	$I_{OL} = 1mA$		0.4		V
Output Leakage Current 32kHz, INT/SQW, SDA	I_{LO}	Output high impedance	-1	0	+1	μA
Input Leakage SCL	I_{LI}		-1	+1		μA
RST Pin I/O Leakage	I_{OL}	RST high impedance (Note 6)	-200	+10		μA
V_{BAT} Leakage Current (V _{CC} Active)	I_{BATLKG}		25	100		nA

Electrical Characteristics (continued)

(V_{CC} = 2.3V to 5.5V, V_{CC} = Active Supply (see Table 1), T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Typical values are at V_{CC} = 3.3V, V_{BAT} = 3.0V, and T_A = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS		
Output Frequency	f _{OUT}	V _{CC} = 3.3V or V _{BAT} = 3.3V		32.768		kHz			
Frequency Stability vs. Temperature (Commercial)	Δf/f _{OUT}	V _{CC} = 3.3V or V _{BAT} = 3.3V, aging offset = 00h	0°C to +40°C	±2		ppm			
			>40°C to +70°C	±3.5					
Frequency Stability vs. Temperature (Industrial)	Δf/f _{OUT}	V _{CC} = 3.3V or V _{BAT} = 3.3V, aging offset = 00h	-40°C to <0°C	±3.5		ppm			
			0°C to +40°C	±2					
			>40°C to +85°C	±3.5					
Frequency Stability vs. Voltage	Δf/V			1		ppm/V			
Trim Register Frequency Sensitivity per LSB	Δf/LSB	Specified at:	-40°C	0.7		ppm			
			+25°C	0.1					
			+70°C	0.4					
			+85°C	0.8					
Temperature Accuracy	Temp	V _{CC} = 3.3V or V _{BAT} = 3.3V		-3	+3		°C		
Crystal Aging	Δf/f _O	After reflow, not production tested	First year	±1.0		ppm			
			0–10 years	±5.0					

Electrical Characteristics

(V_{CC} = 0V, V_{BAT} = 2.3V to 5.5V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 2)

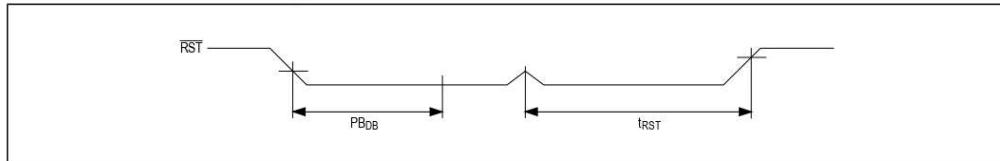
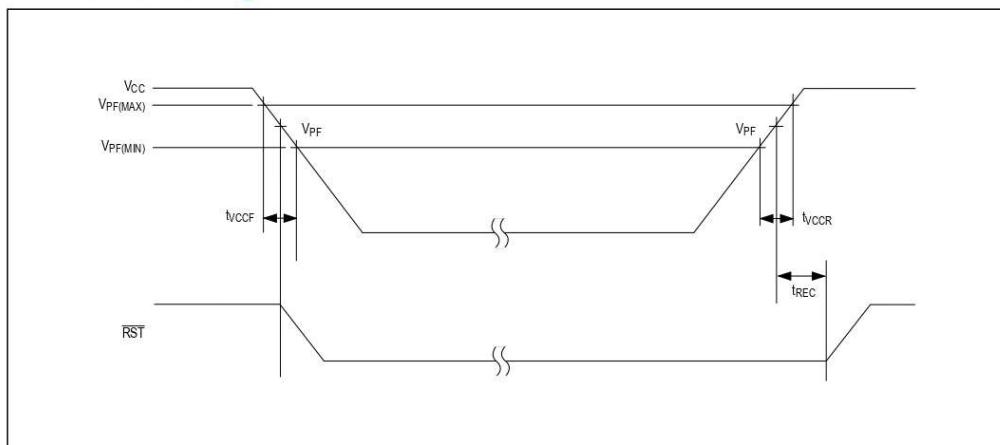
PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Active Battery Current	I _{BATA}	EOSC = 0, BBSQW = 0, SCL = 400kHz (Note 5)	V _{BAT} = 3.63V V _{BAT} = 5.5V	70		150	
Timekeeping Battery Current	I _{BATT}	EOSC = 0, BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = V _{BAT} (Note 5)	V _{BAT} = 3.63V	0.84		3.0	μA
			V _{BAT} = 5.5V	1.0			
Temperature Conversion Current	I _{BATTC}	EOSC = 0, BBSQW = 0, SCL = SDA = 0V or SCL = SDA = V _{BAT}	V _{BAT} = 3.63V	575		650	μA
			V _{BAT} = 5.5V				
Data-Retention Current	I _{BATTDR}	EOSC = 1, SCL = SDA = 0V, +25°C		100		nA	

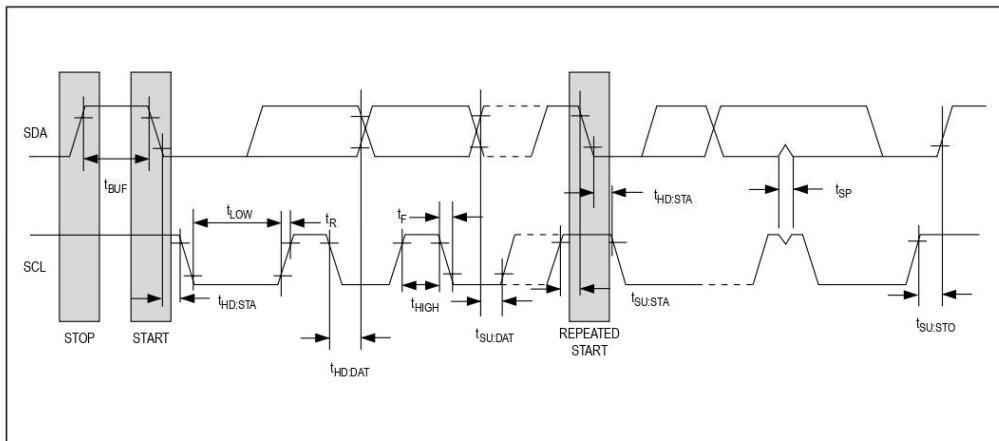
AC Electrical Characteristics(V_{CC} = V_{CC(MIN)} to V_{CC(MAX)} or V_{BAT} = V_{BAT(MIN)} to V_{BAT(MAX)}, V_{BAT} > V_{CC}, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f _{SCL}	Fast mode	100	400		kHz
		Standard mode	0	100		
Bus Free Time Between STOP and START Conditions	t _{BUF}	Fast mode	1.3			μs
		Standard mode	4.7			
Hold Time (Repeated) START Condition (Note 7)	t _{HD:STA}	Fast mode	0.6			μs
		Standard mode	4.0			
Low Period of SCL Clock	t _{LOW}	Fast mode	1.3			μs
		Standard mode	4.7			
High Period of SCL Clock	t _{HIGH}	Fast mode	0.6			μs
		Standard mode	4.0			
Data Hold Time (Notes 8, 9)	t _{HD:DAT}	Fast mode	0	0.9		μs
		Standard mode	0	0.9		
Data Setup Time (Note 10)	t _{SU:DAT}	Fast mode	100			ns
		Standard mode	250			
START Setup Time	t _{SU:STA}	Fast mode	0.6			μs
		Standard mode	4.7			
Rise Time of Both SDA and SCL Signals (Note 11)	t _R	Fast mode	20 + 0.1C _B	300		ns
		Standard mode		1000		
Fall Time of Both SDA and SCL Signals (Note 11)	t _F	Fast mode	20 + 0.1C _B	300		ns
		Standard mode		300		
Setup Time for STOP Condition	t _{SU:STO}	Fast mode	0.6			μs
		Standard mode	4.7			
Capacitive Load for Each Bus Line	C _B	(Note 11)		400		pF
Capacitance for SDA, SCL	C _{I/O}			10		pF
Pulse Width of Spikes That Must Be Suppressed by the Input Filter	t _{SP}			30		ns
Pushbutton Debounce	PB _{DB}			250		ms
Reset Active Time	t _{RST}			250		ms
Oscillator Stop Flag (OSF) Delay	t _{OSF}	(Note 12)		100		ms
Temperature Conversion Time	t _{CONV}			125	200	ms

Power-Switch Characteristics(T_A = T_{MIN} to T_{MAX})

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V _{CC} Fall Time; V _{PF(MAX)} to V _{PF(MIN)}	t _{VCCF}		300			μs
V _{CC} Rise Time; V _{PF(MIN)} to V _{PF(MAX)}	t _{VCCR}		0			μs
Recovery at Power-Up	t _{REC}	(Note 13)		250	300	ms

Pushbutton Reset Timing**Power-Switch Timing**

Data Transfer on I²C Serial Bus

WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

Note 2: Limits at -40°C are guaranteed by design and not production tested.

Note 3: All voltages are referenced to ground.

Note 4: I_{CCA}=SCL clocking at max frequency = 400kHz.

Note 5: Current is the averaged input current, which includes the temperature conversion current.

Note 6: The RST pin has an internal 50kΩ (nominal) pullup resistor to V_{CC}.

Note 7: After this period, the first clock pulse is generated.

Note 8: A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V_{IH(MIN)} of the SCL signal) to bridge the undefined region of the falling edge of SCL.

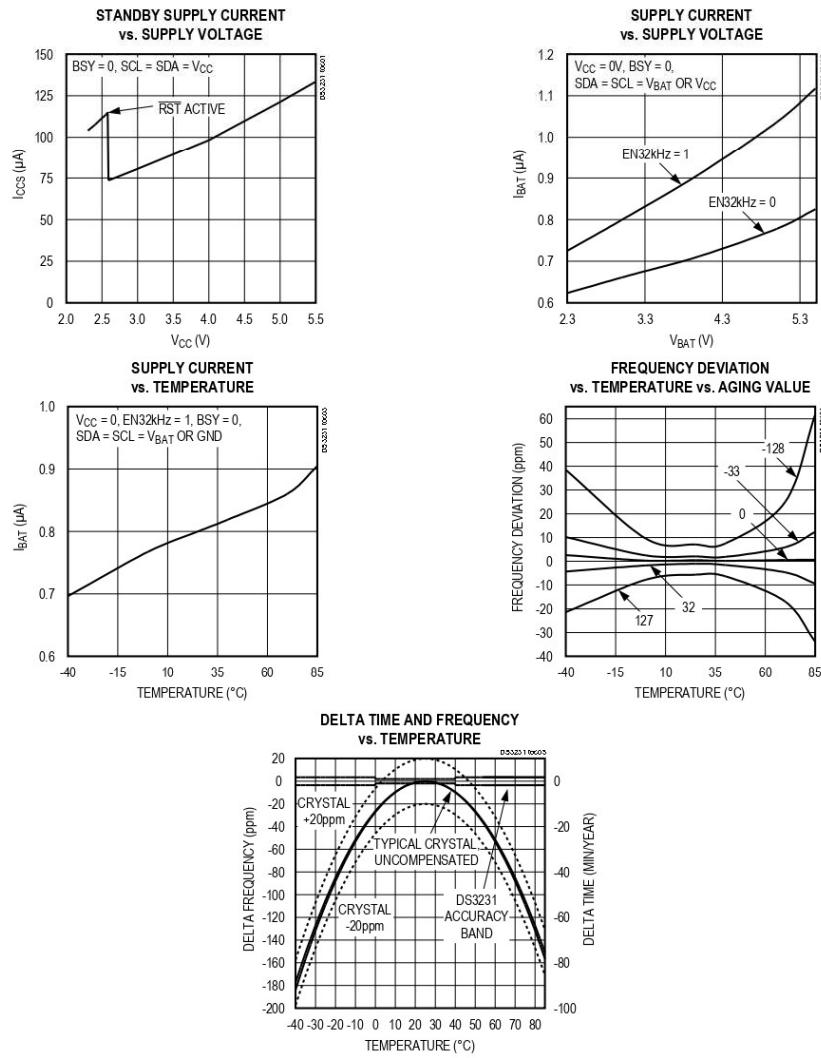
Note 9: The maximum t_{HD:DAT} needs only to be met if the device does not stretch the low period (t_{LOW}) of the SCL signal.

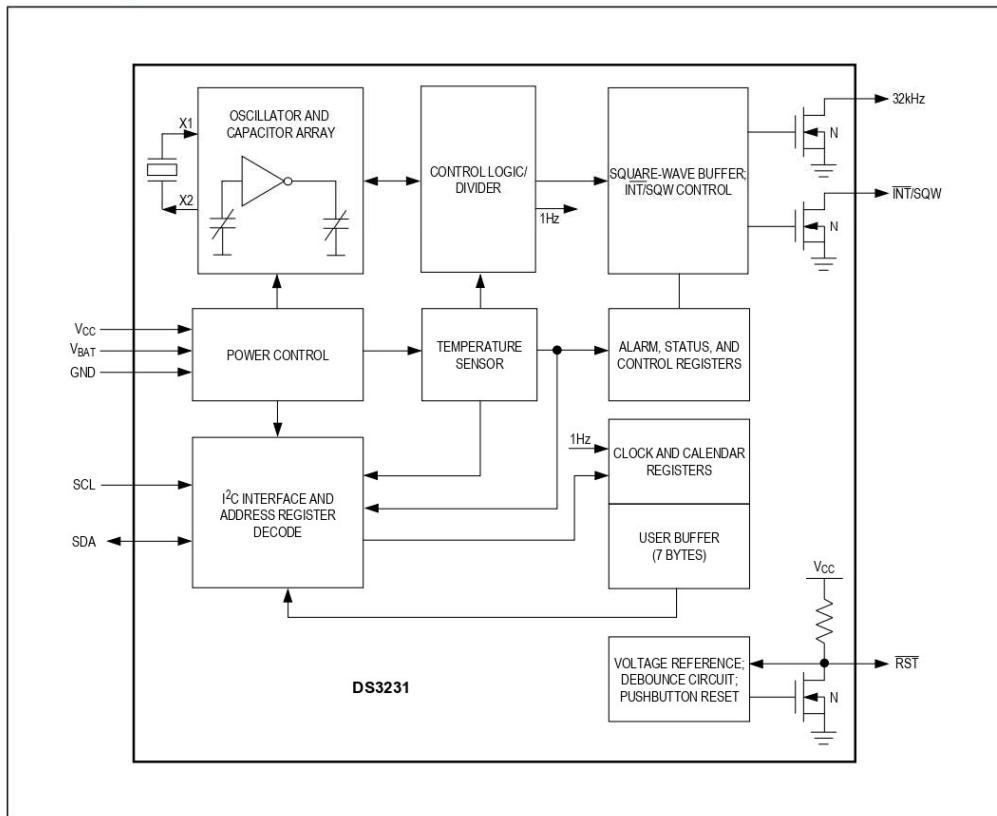
Note 10: A fast-mode device can be used in a standard-mode system, but the requirement t_{SU:DAT} ≥ 250ns must then be met. This is automatically the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line t_{R(MAX)} + t_{SU:DAT} = 1000 + 250 = 1250ns before the SCL line is released.

Note 11: C_B=total capacitance of one bus line in pF.

Note 12: The parameter t_{OSF} is the period of time the oscillator must be stopped for the OSF flag to be set over the voltage range of 0.0V ≤ V_{CC} ≤ V_{CC(MAX)} and 2.3V ≤ V_{BAT} ≤ 3.4V.

Note 13: This delay applies only if the oscillator is enabled and running. If the EOSC bit is a 1, t_{REC} is bypassed and RST immediately goes high. The state of RST does not affect the I²C interface, RTC, or TCXO.

Typical Operating Characteristics(V_{CC} = +3.3V, T_A = +25°C, unless otherwise noted.)

Block Diagram

Pin Description

PIN	NAME	FUNCTION
1	32kHz	32kHz Output. This open-drain pin requires an external pullup resistor. When enabled, the output operates on either power supply. It may be left open if not used.
2	V _{CC}	DC Power Pin for Primary Power Supply. This pin should be decoupled using a 0.1μF to 1.0μF capacitor. If not used, connect to ground.
3	INT/SQW	Active-Low Interrupt or Square-Wave Output. This open-drain pin requires an external pullup resistor connected to a supply at 5.5V or less. This multifunction pin is determined by the state of the INTCN bit in the Control Register (0Eh). When INTCN is set to logic 0, this pin outputs a square wave and its frequency is determined by RS2 and RS1 bits. When INTCN is set to logic 1, then a match between the timekeeping registers and either of the alarm registers activates the INT/SQW pin (if the alarm is enabled). Because the INTCN bit is set to logic 1 when power is first applied, the pin defaults to an interrupt output with alarms disabled. The pullup voltage can be up to 5.5V, regardless of the voltage on V _{CC} . If not used, this pin can be left unconnected.
4	RST	Active-Low Reset. This pin is an open-drain input/output. It indicates the status of V _{CC} relative to the V _{PF} specification. As V _{CC} falls below V _{PF} , the RST pin is driven low. When V _{CC} exceeds V _{PF} , for t _{RST} , the RST pin is pulled high by the internal pullup resistor. The active-low, open-drain output is combined with a debounced pushbutton input function. This pin can be activated by a pushbutton reset request. It has an internal 50kΩ nominal value pullup resistor to V _{CC} . No external pullup resistors should be connected. If the oscillator is disabled, t _{REC} is bypassed and RST immediately goes high.
5-12	N.C.	No Connection. Must be connected to ground.
13	GND	Ground
14	V _{BAT}	Backup Power-Supply Input. When using the device with the V _{BAT} input as the primary power source, this pin should be decoupled using a 0.1μF to 1.0μF low-leakage capacitor. When using the device with the V _{BAT} input as the backup power source, the capacitor is not required. If V _{BAT} is not used, connect to ground. The device is UL recognized to ensure against reverse charging when used with a primary lithium battery. Go to www.maximintegrated.com/qa/info/ui .
15	SDA	Serial Data Input/Output. This pin is the data input/output for the I ² C serial interface. This open-drain pin requires an external pullup resistor. The pullup voltage can be up to 5.5V, regardless of the voltage on V _{CC} .
16	SCL	Serial Clock Input. This pin is the clock input for the I ² C serial interface and is used to synchronize data movement on the serial interface. Up to 5.5V can be used for this pin, regardless of the voltage on V _{CC} .

Detailed Description

The DS3231 is a serial RTC driven by a temperature-compensated 32kHz crystal oscillator. The TCXO provides a stable and accurate reference clock, and maintains the RTC to within ±2 minutes per year accuracy from -40°C to +85°C. The TCXO frequency output is available at the 32kHz pin. The RTC is a low-power clock/calendar with two programmable time-of-day alarms and a programmable square-wave output. The INT/SQW provides either an interrupt signal due to alarm conditions or a square-wave output. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap

year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. The internal registers are accessible through an I²C bus interface.

A temperature-compensated voltage reference and comparator circuit monitors the level of V_{CC} to detect power failures and to automatically switch to the backup supply when necessary. The RST pin provides an external pushbutton function and acts as an indicator of a power-fail event.

Operation

The block diagram shows the main elements of the DS3231. The eight blocks can be grouped into four functional groups: TCXO, power control, pushbutton function, and RTC. Their operations are described separately in the following sections.

32kHz TCXO

The temperature sensor, oscillator, and control logic form the TCXO. The controller reads the output of the on-chip temperature sensor and uses a lookup table to determine the capacitance required, adds the aging correction in AGE register, and then sets the capacitance selection registers. New values, including changes to the AGE register, are loaded only when a change in the temperature value occurs, or when a user-initiated temperature conversion is completed. Temperature conversion occurs on initial application of V_{CC} and once every 64 seconds afterwards.

Power Control

This function is provided by a temperature-compensated voltage reference and a comparator circuit that monitors the V_{CC} level. When V_{CC} is greater than V_{PF}, the part is powered by V_{CC}. When V_{CC} is less than V_{PF} but greater than V_{BAT}, the DS3231 is powered by V_{CC}. If V_{CC} is less than V_{PF} and is less than V_{BAT}, the device is powered by V_{BAT}. See Table 1.

Table 1. Power Control

SUPPLY CONDITION	ACTIVE SUPPLY
V _{CC} < V _{PF} , V _{CC} < V _{BAT}	V _{BAT}
V _{CC} < V _{PF} , V _{CC} > V _{BAT}	V _{CC}
V _{CC} > V _{PF} , V _{CC} < V _{BAT}	V _{CC}
V _{CC} > V _{PF} , V _{CC} > V _{BAT}	V _{CC}

To preserve the battery, the first time V_{BAT} is applied to the device, the oscillator will not start up until V_{CC} exceeds V_{PF}, or until a valid I²C address is written to the part. Typical oscillator startup time is less than one second. Approximately 2 seconds after V_{CC} is applied, or a valid I²C address is written, the device makes a temperature measurement and applies the calculated correction to the oscillator. Once the oscillator is running, it continues to run as long as a valid power source is available (V_{CC} or V_{BAT}), and the device continues to measure the temperature and correct the oscillator frequency every 64 seconds.

On the first application of power (V_{CC}) or when a valid I²C address is written to the part (V_{BAT}), the time and date registers are reset to 01/01/00 01 00:00:00 (DD/MM/YY DOW HH:MM:SS).

V_{BAT} Operation

There are several modes of operation that affect the amount of V_{BAT} current that is drawn. While the device

is powered by V_{BAT} and the serial interface is active, active battery current, I_{BATA}, is drawn. When the serial interface is inactive, timekeeping current (I_{BATT}), which includes the averaged temperature conversion current, I_{BATTC}, is used (refer to Application Note 3644: *Power Considerations for Accurate Real-Time Clocks* for details). Temperature conversion current, I_{BATTC}, is specified since the system must be able to support the periodic higher current pulse and still maintain a valid voltage level. Data retention current, I_{BATTDR}, is the current drawn by the part when the oscillator is stopped (EOSC = 1). This mode can be used to minimize battery requirements for times when maintaining time and date information is not necessary, e.g., while the end system is waiting to be shipped to a customer.

Pushbutton Reset Function

The DS3231 provides for a pushbutton switch to be connected to the RST output pin. When the DS3231 is not in a reset cycle, it continuously monitors the RST signal for a low going edge. If an edge transition is detected, the DS3231 debounces the switch by pulling the RST low. After the internal timer has expired (PB_{DB}), the DS3231 continues to monitor the RST line. If the line is still low, the DS3231 continuously monitors the line looking for a rising edge. Upon detecting release, the DS3231 forces the RST pin low and holds it low for t_{RST}.

RST is also used to indicate a power-fail condition. When V_{CC} is lower than V_{PF}, an internal power-fail signal is generated, which forces the RST pin low. When V_{CC} returns to a level above V_{PF}, the RST pin is held low for approximately 250ms (t_{REC}) to allow the power supply to stabilize. If the oscillator is not running (see the *Power Control* section) when V_{CC} is applied, t_{REC} is bypassed and RST immediately goes high. Assertion of the RST output, whether by pushbutton or power-fail detection, does not affect the internal operation of the DS3231.

Real-Time Clock

With the clock source from the TCXO, the RTC provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator.

The clock provides two programmable time-of-day alarms and a programmable square-wave output. The INT/SQW pin either generates an interrupt due to alarm condition or outputs a square-wave signal and the selection is controlled by the bit INTCN.

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE		
00h	0		10 Seconds			Seconds			Seconds	00–59		
01h	0		10 Minutes			Minutes			Minutes	00–59		
02h	0	12/24	AM/PM 20 Hour	10 Hour	Hour			Hours	1–12 + AM/PM 00–23			
03h	0	0			0	0	Day		Day	1–7		
04h	0	0	10 Date		Date			Month	Month/ Century	01–12 + Century		
05h	Century	0	0	10 Month	Month							
06h		10 Year			Year			Year	00–99			
07h	A1M1		10 Seconds			Seconds			Alarm 1 Seconds	00–59		
08h	A1M2		10 Minutes			Minutes			Alarm 1 Minutes	00–59		
09h	A1M3	12/24	AM/PM 20 Hour	10 Hour	Hour			Alarm 1 Hours	1–12 + AM/PM 00–23			
0Ah	A1M4	DY/DT	10 Date		Day				Alarm 1 Day	1–7		
0Bh	A2M2		10 Minutes			Date			Alarm 1 Date	1–31		
0Ch	A2M3	12/24	AM/PM 20 Hour	10 Hour	Minutes			Alarm 2 Minutes	00–59			
0Dh	A2M4	DY/DT	10 Date		Hour				Alarm 2 Hours			
0Eh	EOSC	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE	Control	—		
0Fh	OSF	0	0	0	EN32kHz	BSY	A2F	A1F	Control/Status	—		
10h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	Aging Offset	—		
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—		
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—		

Figure 1. Timekeeping Registers

Note: Unless otherwise specified, the registers' state is not defined when power is first applied.

Address Map

Figure 1 shows the address map for the DS3231 time-keeping registers. During a multibyte access, when the address pointer reaches the end of the register space (12h), it wraps around to location 00h. On an I²C START or address pointer incrementing to location 00h, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to reread the registers in case the main registers update during a read.

I²C Interface

The I²C interface is accessible whenever either V_{CC} or V_{BAT} is at a valid level. If a microcontroller connected

to the DS3231 resets because of a loss of V_{CC} or other event, it is possible that the microcontroller and DS3231 I²C communications could become unsynchronized, e.g., the microcontroller resets while reading data from the DS3231. When the microcontroller resets, the DS3231 I²C interface may be placed into a known state by toggling SCL until SDA is observed to be at a high level. At that point the microcontroller should pull SDA low while SCL is high, generating a START condition.

Clock and Calendar

The time and calendar information is obtained by reading the appropriate register bytes. Figure 1 illustrates the RTC registers. The time and calendar data are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the binary-coded

decimal (BCD) format. The DS3231 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic-high being PM. In the 24-hour mode, bit 5 is the 20-hour bit (20–23 hours). The century bit (bit 7 of the month register) is toggled when the years register overflows from 99 to 00.

The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on). Illogical time and date entries result in undefined operation.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any START and when the register pointer rolls over to zero. The time information is read from these secondary registers, while the clock continues to run. This eliminates the need to reread the registers in case the main registers update during a read.

The countdown chain is reset whenever the seconds register is written. Write transfers occur on the acknowledge from the DS3231. Once the countdown chain is reset, to avoid rollover issues the remaining time and date registers must be written within 1 second. The 1Hz square-wave output, if enabled, transitions high 500ms after the seconds data transfer, provided the oscillator is already running.

Alarms

The DS3231 contains two time-of-day/date alarms. Alarm 1 can be set by writing to registers 07h to 0Ah. Alarm 2 can be set by writing to registers 0Bh to 0Dh. The alarms can be programmed (by the alarm enable and INTcn bits of the control register) to activate the INT/SQW output on an alarm match condition. Bit 7 of each of the time-of-day/date alarm registers are mask bits (Table 2). When all the mask bits for each alarm are logic 0, an alarm only occurs when the values in the timekeeping registers match the corresponding values stored in the time-of-day/date alarm registers. The alarms can also be programmed to repeat every second, minute, hour, day, or date. Table 2 shows the possible settings. Configurations not listed in the table will result in illogical operation.

The DY/D^T bits (bit 6 of the alarm day/date registers) control whether the alarm value stored in bits 0 to 5 of that register reflects the day of the week or the date of the month. If DY/D^T is written to logic 0, the alarm will be the result of a match with date of the month. If DY/D^T is written to logic 1, the alarm will be the result of a match with day of the week.

When the RTC register values match alarm register settings, the corresponding Alarm Flag 'A1F' or 'A2F' bit is set to logic 1. If the corresponding Alarm Interrupt Enable 'A1IE' or 'A2IE' is also set to logic 1 and the INTcn bit is set to logic 1, the alarm condition will activate the INT/SQW signal. The match is tested on the once-per-second update of the time and date registers.

Table 2. Alarm Mask Bits

DY/D ^T	ALARM 1 REGISTER MASK BITS (BIT 7)				ALARM RATE
	A1M4	A1M3	A1M2	A1M1	
X	1	1	1	1	Alarm once per second
X	1	1	1	0	Alarm when seconds match
X	1	1	0	0	Alarm when minutes and seconds match
X	1	0	0	0	Alarm when hours, minutes, and seconds match
0	0	0	0	0	Alarm when date, hours, minutes, and seconds match
1	0	0	0	0	Alarm when day, hours, minutes, and seconds match

DY/D ^T	ALARM 2 REGISTER MASK BITS (BIT 7)			ALARM RATE
	A2M4	A2M3	A2M2	
X	1	1	1	Alarm once per minute (00 seconds of every minute)
X	1	1	0	Alarm when minutes match
X	1	0	0	Alarm when hours and minutes match
0	0	0	0	Alarm when date, hours, and minutes match
1	0	0	0	Alarm when day, hours, and minutes match

Control Register (0Eh)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	EOSC	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE
POR:	0	0	0	1	1	1	0	0

Special-Purpose Registers

The DS3231 has two additional registers (control and status) that control the real-time clock, alarms, and square-wave output.

Control Register (0Eh)

Bit 7: Enable Oscillator (EOSC). When set to logic 0, the oscillator is started. When set to logic 1, the oscillator is stopped when the DS3231 switches to V_{BAT}. This bit is clear (logic 0) when power is first applied. When the DS3231 is powered by V_{CC}, the oscillator is always on regardless of the status of the EOSC bit. When EOSC is disabled, all register data is static.

Bit 6: Battery-Backed Square-Wave Enable (BBSQW). When set to logic 1 with INTCN = 0 and V_{CC} < V_{PF}, this bit enables the square wave. When BBSQW is logic 0, the INT/SQW pin goes high impedance when V_{CC} < V_{PF}. This bit is disabled (logic 0) when power is first applied.

Bit 5: Convert Temperature (CONV). Setting this bit to 1 forces the temperature sensor to convert the temperature into digital code and execute the TCXO algorithm to update the capacitance array to the oscillator. This can only happen when a conversion is not already in progress. The user should check the status bit BSY before forcing the controller to start a new TCXO execution. A user-initiated temperature conversion does not affect the internal 64-second update cycle.

A user-initiated temperature conversion does not affect the BSY bit for approximately 2ms. The CONV bit remains at a 1 from the time it is written until the conversion is finished, at which time both CONV and BSY go to 0. The CONV bit should be used when monitoring the status of a user-initiated conversion.

Bits 4 and 3: Rate Select (RS2 and RS1). These bits control the frequency of the square-wave output when

the square wave has been enabled. The following table shows the square-wave frequencies that can be selected with the RS bits. These bits are both set to logic 1 (8.192kHz) when power is first applied.

SQUARE-WAVE OUTPUT FREQUENCY

RS2	RS1	SQUARE-WAVE OUTPUT FREQUENCY
0	0	1Hz
0	1	1.024kHz
1	0	4.096kHz
1	1	8.192kHz

Bit 2: Interrupt Control (INTCN). This bit controls the INT/SQW signal. When the INTCN bit is set to logic 0, a square wave is output on the INT/SQW pin. When the INTCN bit is set to logic 1, then a match between the time-keeping registers and either of the alarm registers activates the INT/SQW output (if the alarm is also enabled). The corresponding alarm flag is always set regardless of the state of the INTCN bit. The INTCN bit is set to logic 1 when power is first applied.

Bit 1: Alarm 2 Interrupt Enable (A2IE). When set to logic 1, this bit permits the alarm 2 flag (A2F) bit in the status register to assert INT/SQW (when INTCN = 1). When the A2IE bit is set to logic 0 or INTCN is set to logic 0, the A2F bit does not initiate an interrupt signal. The A2IE bit is disabled (logic 0) when power is first applied.

Bit 0: Alarm 1 Interrupt Enable (A1IE). When set to logic 1, this bit permits the alarm 1 flag (A1F) bit in the status register to assert INT/SQW (when INTCN = 1). When the A1IE bit is set to logic 0 or INTCN is set to logic 0, the A1F bit does not initiate the INT/SQW signal. The A1IE bit is disabled (logic 0) when power is first applied.

Status Register (0Fh)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	OSF	0	0	0	EN32kHz	BSY	A2F	A1F
POR:	1	0	0	0	1	X	X	X

Status Register (0Fh)

Bit 7: Oscillator Stop Flag (OSF). A logic 1 in this bit indicates that the oscillator either is stopped or was stopped for some period and may be used to judge the validity of the timekeeping data. This bit is set to logic 1 any time that the oscillator stops. The following are examples of conditions that can cause the OSF bit to be set:

- 1) The first time power is applied.
- 2) The voltages present on both V_{CC} and V_{BAT} are insufficient to support oscillation.
- 3) The EOSC bit is turned off in battery-backed mode.
- 4) External influences on the crystal (i.e., noise, leakage, etc.).

This bit remains at logic 1 until written to logic 0.

Bit 3: Enable 32kHz Output (EN32kHz). This bit controls the status of the 32kHz pin. When set to logic 1, the 32kHz pin is enabled and outputs a 32.768kHz square-wave signal. When set to logic 0, the 32kHz pin goes to a high-impedance state. The initial power-up state of this bit is logic 1, and a 32.768kHz square-wave signal appears at the 32kHz pin after a power source is applied to the DS3231 (if the oscillator is running).

Bit 2: Busy (BSY). This bit indicates the device is busy executing TCXO functions. It goes to logic 1 when the conversion signal to the temperature sensor is asserted and then is cleared when the device is in the 1-minute idle state.

Bit 1: Alarm 2 Flag (A2F). A logic 1 in the alarm 2 flag bit indicates that the time matched the alarm 2 registers. If the A2IE bit is logic 1 and the INTCN pin is set to logic 1, the INT/SQW pin is also asserted. A2F is cleared when written to logic 0. This bit can only be written to logic 0. Attempting to write to logic 1 leaves the value unchanged.

Bit 0: Alarm 1 Flag (A1F). A logic 1 in the alarm 1 flag bit indicates that the time matched the alarm 1 registers. If the

A1IE bit is logic 1 and the INTCN bit is set to logic 1, the INT/SQW pin is also asserted. A1F is cleared when written to logic 0. This bit can only be written to logic 0. Attempting to write to logic 1 leaves the value unchanged.

Aging Offset

The aging offset register takes a user-provided value to add to or subtract from the codes in the capacitance array registers. The code is encoded in two's complement, with bit 7 representing the sign bit. One LSB represents one small capacitor to be switched in or out of the capacitance array at the crystal pins. The aging offset register capacitance value is added or subtracted from the capacitance value that the device calculates for each temperature compensation. The offset register is added to the capacitance array during a normal temperature conversion, if the temperature changes from the previous conversion, or during a manual user conversion (setting the CONV bit). To see the effects of the aging register on the 32kHz output frequency immediately, a manual conversion should be started after each aging register change.

Positive aging values add capacitance to the array, slowing the oscillator frequency. Negative values remove capacitance from the array, increasing the oscillator frequency.

The change in ppm per LSB is different at different temperatures. The frequency vs. temperature curve is shifted by the values used in this register. At +25°C, one LSB typically provides about 0.1ppm change in frequency.

Use of the aging register is not needed to achieve the accuracy as defined in the EC tables, but could be used to help compensate for aging at a given temperature. See the *Typical Operating Characteristics* section for a graph showing the effect of the register on accuracy over temperature.

Aging Offset (10h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Sign	Data						
POR:	0	0	0	0	0	0	0	0

Temperature Register (Upper Byte) (11h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Sign	Data						
POR:	0	0	0	0	0	0	0	0

Temperature Register (Lower Byte) (12h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Data	Data	0	0	0	0	0	0
POR:	0	0	0	0	0	0	0	0

Temperature Registers (11h–12h)

Temperature is represented as a 10-bit code with a resolution of 0.25°C and is accessible at location 11h and 12h. The temperature is encoded in two's complement format. The upper 8 bits, the integer portion, are at location 11h and the lower 2 bits, the fractional portion, are in the upper nibble at location 12h. For example, 00011001 01b = +25.25°C. Upon power reset, the registers are set to a default temperature of 0°C and the controller starts a temperature conversion. The temperature is read on initial application of V_{CC} or I²C access on V_{BAT} and once every 64 seconds afterwards. The temperature registers are updated after each user-initiated conversion and on every 64-second conversion. The temperature registers are read-only.

I²C Serial Data Bus

The DS3231 supports a bidirectional I²C bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data is defined as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS3231 operates as a slave on the I²C bus. Connections to the bus are made through the SCL input and open-drain SDA I/O lines. Within the bus specifications, a standard mode (100kHz maximum clock rate) and a fast mode (400kHz maximum clock rate) are defined. The DS3231 works in both modes.

The following bus protocol has been defined (Figure 2):

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high. Changes in the data

line while the clock line is high are interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain high.

START data transfer: A change in the state of the data line from high to low, while the clock line is high, defines a START condition.

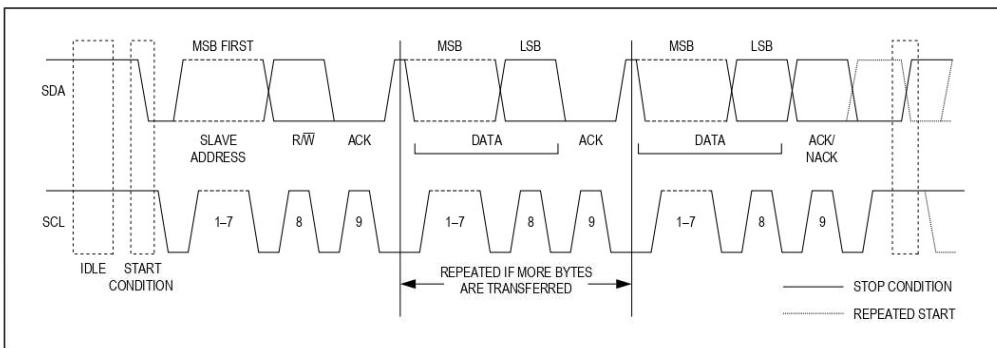
STOP data transfer: A change in the state of the data line from low to high, while the clock line is high, defines a STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the high period of the clock signal. The data on the line must be changed during the low period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and the STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse, which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge-related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generat-

Figure 2. I²C Data Transfer Overview

ing an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line high to enable the master to generate the STOP condition.

Figures 3 and 4 detail how data transfer is accomplished on the I²C bus. Depending upon the state of the R/W bit, two types of data transfer are possible:

Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master

is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.

Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows a number of data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the

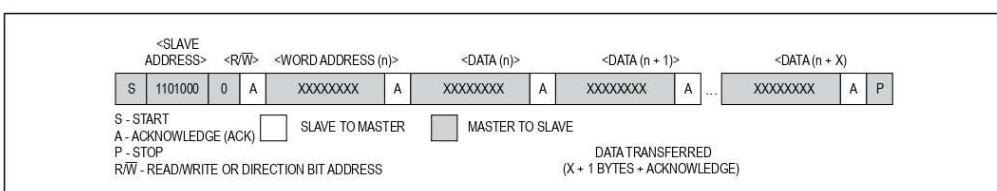


Figure 3. Data Write—Slave Receiver Mode

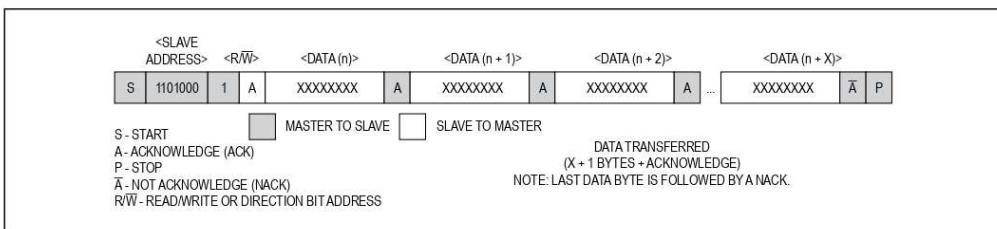


Figure 4. Data Read—Slave Transmitter Mode

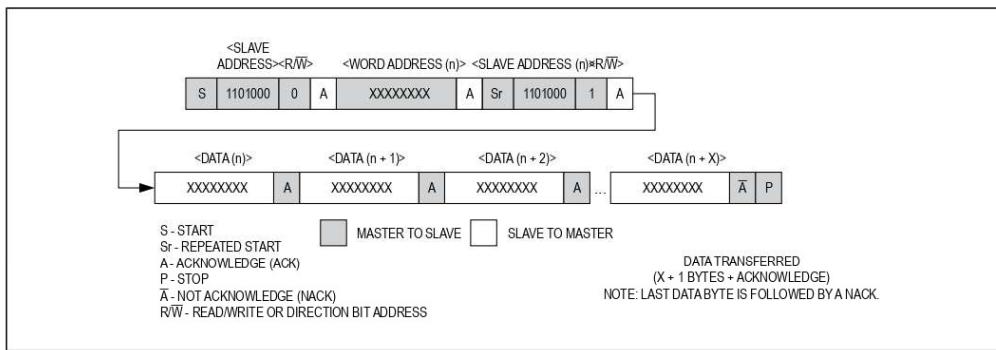


Figure 5. Data Write/Read (Write Pointer, Then Read)—Slave Receive and Transmit

last byte. At the end of the last received byte, a not acknowledge is returned.

The master device generates all the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS3231 can operate in the following two modes:

Slave receiver mode (DS3231 write mode): Serial data and clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit. The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS3231 address, which is 1101000, followed by the direction bit (R/W), which is 0 for a write. After receiving and decoding the slave address byte, the DS3231 outputs an acknowledge on SDA. After the DS3231 acknowledges the slave address + write bit, the master transmits a word address to the DS3231. This sets the register pointer on the DS3231, with the DS3231 acknowledging the

transfer. The master may then transmit zero or more bytes of data, with the DS3231 acknowledging each byte received. The register pointer increments after each data byte is transferred. The master generates a STOP condition to terminate the data write.

Slave transmitter mode (DS3231 read mode): The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit indicates that the transfer direction is reversed. Serial data is transmitted on SDA by the DS3231 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit. The slave address byte is the first byte received after the master generates a START condition. The slave address byte contains the 7-bit DS3231 address, which is 1101000, followed by the direction bit (R/W), which is 1 for a read. After receiving and decoding the slave address byte, the DS3231 outputs an acknowledge on SDA. The DS3231 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode, the first address that is read is the last one stored in the register pointer. The DS3231 must receive a not acknowledge to end a read.

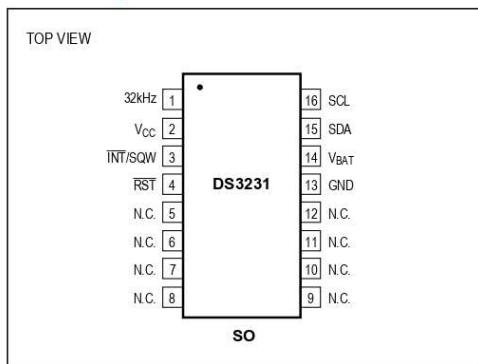
Handling, PCB Layout, and Assembly

The DS3231 package contains a quartz tuning-fork crystal. Pick-and-place equipment can be used, but precautions should be taken to ensure that excessive shocks are avoided. Ultrasonic cleaning should be avoided to prevent damage to the crystal.

Avoid running signal traces under the package, unless a ground plane is placed between the package and the

signal line. All N.C. (no connect) pins must be connected to ground.

Moisture-sensitive packages are shipped from the factory dry packed. Handling instructions listed on the package label must be followed to prevent damage during reflow. Refer to the IPC/JEDEC J-STD-020 standard for moisture-sensitive device (MSD) classifications and reflow profiles. Exposure to reflow is limited to 2 times maximum.

Pin Configuration**Chip Information**

SUBSTRATE CONNECTED TO GROUND
PROCESS: CMOS

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
DS3231S#	0°C to +70°C	16 SO
DS3231SN#	-40°C to +85°C	16 SO

#Denotes an RoHS-compliant device that may include lead (Pb) that is exempt under RoHS requirements. The lead finish is JEDEC97 category e3, and is compatible with both lead-based and lead-free soldering processes. A "#" anywhere on the top mark denotes an RoHS-compliant device.

Package Information

For the latest package outline information and land patterns (footprints), go to www.maximintegrated.com/packages. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

PACKAGE TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN NO.
16 SO	W16#H2	21-0042	90-0107

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	1/05	Initial release.	—
1	2/05	Changed Digital Temp Sensor Output from $\pm 2^{\circ}\text{C}$ to $\pm 3^{\circ}\text{C}$.	1, 3
		Updated <i>Typical Operating Circuit</i> .	1
		Changed $T_A = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ to $T_A = T_{\text{MIN}}$ to T_{MAX} .	2, 3, 4
		Updated <i>Block Diagram</i> .	8
2	6/05	Added "UL Recognized" to <i>Features</i> ; added lead-free packages and removed S from top mark info in <i>Ordering Information</i> table; added ground connections to the N.C. pin in the <i>Typical Operating Circuit</i> .	1
		Added "noncondensing" to operating temperature range; changed V_{PF} MIN from 2.35V to 2.45V.	2
		Added aging offset specification.	3
		Relabeled TOC4.	7
		Added arrow showing input on X1 in the <i>Block Diagram</i> .	8
		Updated pin descriptions for V_{CC} and V_{BAT} .	9
		Added the I ² C Interface section.	10
		<i>Figure 1:</i> Added sign bit to aging and temperature registers; added MSB and LSB.	11
		Corrected title for rate select bits frequency table.	13
		Added note that frequency stability over temperature spec is with aging offset register = 00h; changed bit 7 from Data to Sign (Crystal Aging Offset Register).	14
		Changed bit 7 from Data to Sign (Temperature Register); correct pin definitions in I ² C Serial Data Bus section.	15
		Modified the <i>Handling, PC Board Layout, and Assembly</i> section to refer to J-STD-020 for reflow profiles for lead-free and leaded packages.	17
3	11/05	Changed lead-free packages to RoHS-compliant packages.	1
4	10/06	Changed $\overline{\text{RST}}$ and UL bullets in <i>Features</i> .	1
		Changed EC condition " $V_{\text{CC}} > V_{\text{BAT}}$ " to " $V_{\text{CC}} = \text{Active Supply}$ (see Table 1)."	2, 3
		Modified Note 12 to correct t_{REC} operation.	6
		Added various conditions text to TOCs 1, 2, and 3.	7
		Added text to pin descriptions for 32kHz, V_{CC} , and $\overline{\text{RST}}$.	9
		Table 1: Changed column heading "Powered By" to "Active Supply"; changed "applied" to "exceeds V_{PF} " in the <i>Power Control</i> section.	10
		Indicated BBSQW applies to both SQW and interrupts; simplified temp convert description (bit 5); added "output" to INT/SQW (bit 2).	13
5	4/08	Changed the <i>Crystal Aging</i> section to the <i>Aging Offset</i> section; changed "this bit indicates" to "this bit controls" for the enable 32kHz output bit.	14
		Added Warning note to EC table notes; updated Note 12.	6
		Updated the <i>Typical Operating Characteristics</i> graphs.	7
		In the <i>Power Control</i> section, added information about the POR state of the time and date registers; in the <i>Real-Time Clock</i> section, added to the description of the RST function.	10
		In Figure 1, corrected the months date range for 04h from 00–31 to 01–31.	11

Revision History (continued)

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
6	10/08	Updated the <i>Typical Operating Circuit</i> .	1
		Removed the V_{PU} parameter from the <i>Recommended DC Operating Conditions</i> table and added verbiage about the pullup to the <i>Pin Description</i> table for INT/SQW, SDA, and SCL.	2, 9
		Added the Delta Time and Frequency vs. Temperature graph in the <i>Typical Operating Characteristics</i> section.	7
		Updated the <i>Block Diagram</i> .	8
		Added the V_{BAT} <i>Operation</i> section, improved some sections of text for the 32kHz TCXO and Pushbutton Reset Function sections.	10
		Added the register bit POR values to the register tables.	13, 14, 15
		Updated the Aging Offset and Temperature Registers (11h–12h) sections.	14, 15
7	3/10	Updated the I ² C timing diagrams (Figures 3, 4, and 5).	16, 17
		Removed the "S" from the top mark in the <i>Ordering Information</i> table and the <i>Pin Configuration</i> to match the packaging engineering marking specification.	1, 18
8	7/10	Updated the <i>Typical Operating Circuit</i> ; removed the "Top Mark" column from the <i>Ordering Information</i> ; in the <i>Absolute Maximum Ratings</i> section, added the theta-JA and theta-JC thermal resistances and Note 1, and changed the soldering temperature to +260°C (lead(Pb)-free) and +240°C (leaded); updated the functional description of the V_{BAT} pin in the <i>Pin Description</i> ; changed the timekeeping registers 02h, 09h, and 0Ch to "20 Hour" in Bit 5 of Figure 1; updated the BBSQW bit description in the <i>Control Register (0Eh)</i> section; added the land pattern no. to the <i>Package Information</i> table.	1, 2, 3, 4, 6, 9, 11, 12, 13, 18
9	1/13	Updated <i>Absolute Maximum Ratings</i> , and last paragraph in <i>Power Control</i> section	2, 10
10	3/15	Revised <i>Benefits and Features</i> section.	1

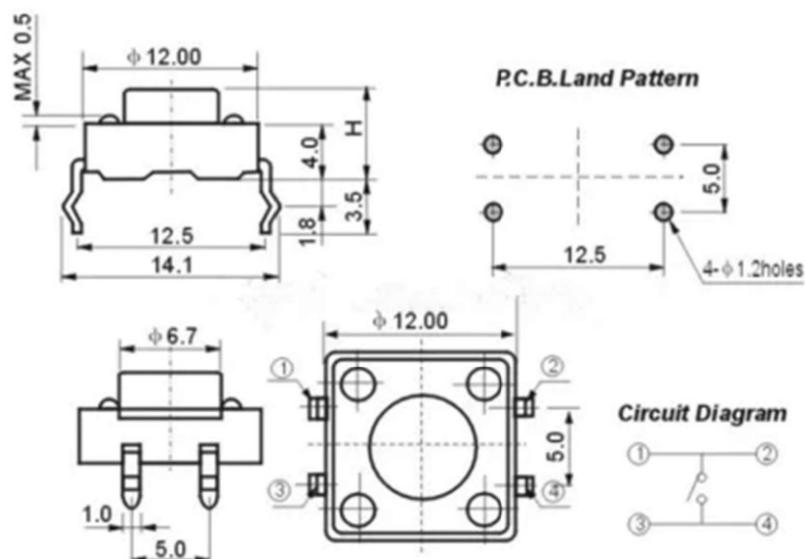
For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim Integrated's website at www.maximintegrated.com.

Maxim Integrated cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim Integrated product. No circuit patent licenses are implied. Maxim Integrated reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.

Maxim Integrated and the Maxim Integrated logo are trademarks of Maxim Integrated Products, Inc. © 2015 Maxim Integrated Products, Inc. | 20

Bijlage C6 Datasheet Tactile Momentary Push Button White

Tactile Momentary Push Button White 12mm



Description

Momentary contact, round push button, 4 pins, through hole tact switch.

- Push-button Switch
- catching clamps and buttons.
- 12 x 12mm Push Button Switch with white round cap.
- Overall height: 15mm

Bijlage C7 Datasheet Arduino Nano ESP32-S3



Arduino® Nano ESP32

Product Reference Manual
SKU: ABX00083



Description

The Arduino® Nano ESP32 (with and without headers) is a Nano form factor board based on the ESP32-S3 (embedded in the NORA-W106-10B from u-blox®). This is the first Arduino board to be based fully on an ESP32, and features Wi-Fi® as well as Bluetooth® LE.

The Nano ESP32 is compatible with the Arduino Cloud, and has support for MicroPython. It is an ideal board for getting started with IoT development.

Target areas:

Maker, IoT, MicroPython



Features

- **Xtensa® Dual-core 32-bit LX7 Microprocessor**

- Up to 240 MHz
- 384 kB ROM
- 512 kB SRAM
- 16 kB SRAM in RTC (low power mode)
- DMA Controller

- **Power**

- Operating voltage 3.3 V
- VBUS supplies 5 V via USB-C® connector
- VIN range is 6-21 V

- **Connectivity**

- Wi-Fi®
- Bluetooth® LE
- Built-in antenna
- 2.4 GHz transmitter/receiver
- Up to 150 Mbps

- **Pins**

- 14x digital (21x including analog)
- 8x analog (available in RTC mode)
- SPI(D11,D12,D13), I2C (A4/A5), UART(D0/D1)

- **Communication Ports**

- SPI
- I2C
- I2S
- UART
- CAN (TWAI®)

- **Low Power**

- 7 µA consumption in deep sleep mode*
- 240 µA consumption in light sleep mode*
- RTC Memory
- Ultra Low Power (ULP) Coprocessor
- Power Management Unit (PMU)
- ADC in RTC mode

*The power consumption ratings listed in low power modes are only for the ESP32-S3 SoC. Other components on the board (such as LEDs), consumes power as well, which increases the overall power consumption of the board.



Contents

1 The Board	6
1.1 Application Examples	6
2 ESP32 Core	6
3 Recommended Operating Conditions	6
4 Block Diagram	7
5 Board Topology	8
5.1 Front View	8
6 NORA-W106-10B (Radio Module / MCU)	9
6.1 Xtensa® Dual-Core 32bit LX7 Microprocessor	9
6.2 Wi-Fi®	9
6.3 Bluetooth®	9
6.4 PSRAM	10
6.5 Antenna Gain	10
7 System	10
7.1 Resets	10
7.2 Timers	11
7.3 Interrupts	11
8 Serial Communication Protocols	11
8.1 Inter-Integrated Circuit (I2C)	11
8.2 Inter-IC Sound (I2S)	12
8.3 Serial Peripheral Interface (SPI)	12
8.4 Universal Asynchronous Receiver/Transmitter (UART)	12
8.5 Two Wire Automotive Interface (TWAI®)	13
9 External Flash Memory	13
10 USB Connector	13
11 Power Options	13
11.1 Power Tree	14
11.2 Pin Voltage	14
11.3 VIN Rating	14
11.4 VBUS	15
11.5 Using the 3.3 V Pin	15
11.6 Pin Current	15
12 Pinout	15



12.1 Analog (JP1)	17
12.2 Digital (JP2)	17
13 Mounting Holes And Board Outline	18
14 Board Operation	18
14.1 Getting Started - IDE	18
14.2 Getting Started - Arduino Cloud Editor	19
14.3 Getting Started - Arduino Cloud	19
14.4 Online Resources	19
14.5 Board Recovery	19
15 Declaration of Conformity CE DoC (EU)	19
16 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021	
17 Conflict Minerals Declaration	21
18 FCC Caution	21
19 SRRCC	22
20 Company Information	22
21 Reference Documentation	23
22 Change Log	23
23 电路板简介	
23.1 应用示例	
24 ESP32 核心板	
25 建议运行条件	
26 方框图	
27 电路板拓扑结构	
27.1 前视图	
28 NORA-W106-10B (无线电模块 / MCU)	
28.1 Xtensa® 双核 32 位 LX7 微处理器	29
28.2 Wi-Fi®	29
28.3 Bluetooth®	29
28.4 PSRAM	30
28.5 天线增益	
29 系统	
29.1 复位	
29.2 定时器	
29.3 中断引脚	
30 串行通信协议	
30.1 集成电路总线 (I2C)	



30.2 I ₂ S总线 (I ₂ S)	
30.3 串行外设接口 (SPI)	
30.4 通用异步接收器/发射器 (UART)	
30.5 双线汽车接口 (TWAI®)	
31 外部闪存	
32 USB 连接器	
33 电源选项	
33.1 电源树	
33.2 引脚电压	
33.3 VIN 额定值	
33.4 VBUS	34
33.5 使用 3.3 V 引脚	
33.6 引脚电流	
34 引脚布局	
34.1 模拟引脚 (JP1)	
34.2 数字引脚 (JP2)	
35 安装孔和电路板外形图	
36 电路板操作	
36.1 入门指南 - IDE	
36.2 入门指南 - Arduino Cloud Editor	
36.3 入门指南 - Arduino Cloud	
36.4 在线资源	
36.5 电路板恢复	
37 符合性声明 CE DoC (欧盟)	
38 声明符合欧盟 RoHS 和 REACH 2011/01/19/2021	
39 冲突矿产声明	
40 FCC 警告	
41 SRRC	41
42 公司信息	
43 参考资料	
44 变更日志	



1 The Board

Nano ESP32 is a 3.3 V development board based on the NORA-W106-10B from u-blox®, a module that includes a ESP32-S3 system on a chip (SoC). This module has support for Wi-Fi® and Bluetooth® Low Energy (LE), with amplified communication through a built-in antenna. The CPU (32-bit Xtensa® LX7) supports clock frequencies at up to 240 MHz.

1.1 Application Examples

Home automation: an ideal board for automating your home, and can be used for smart switches, automatic lighting and motor control for e.g. motor controlled blinds.

IoT sensors: with several dedicated ADC channels, accessible I2C/SPI buses and a robust ESP32-S3 based radio module, this board can easily be deployed to monitor sensor values.

Low power designs: create battery powered applications with low power consumption, utilising the built in low power modes of the ESP32-S3 SoC.

2 ESP32 Core

The Nano ESP32 uses the Arduino Board Package for ESP32 boards, a derivation of Espressif's arduino-esp32 core.

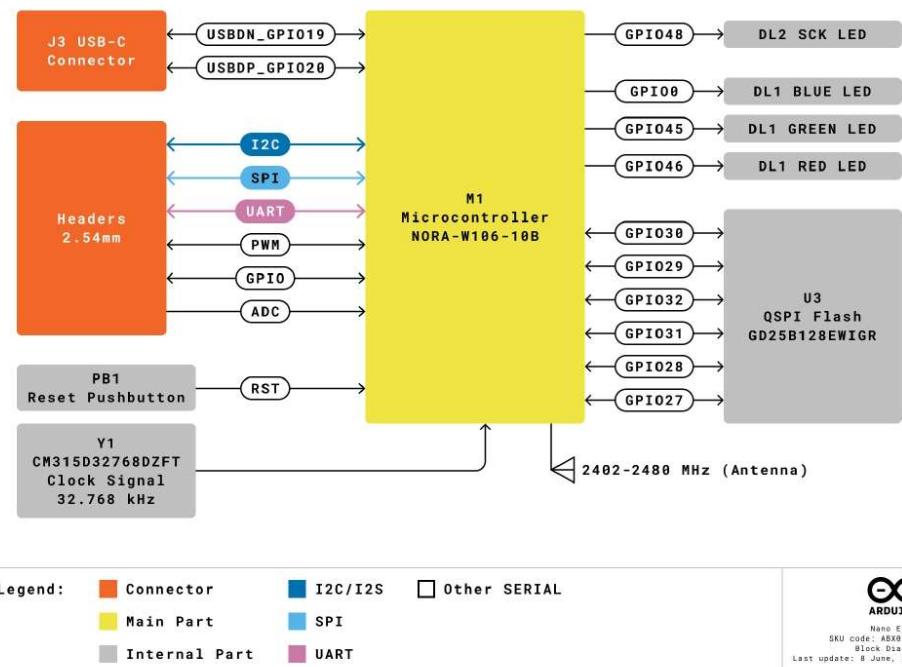
Rating

3 Recommended Operating Conditions

Symbol	Description	Min	Typ	Max	Unit
V _{IN}	Input voltage from VIN pad	6	7.0	21	V
V _{USB}	Input voltage from USB connector	4.8	5.0	5.5	V
T _{ambient}	Ambient Temperature	-40	25	105	°C

Functional Overview

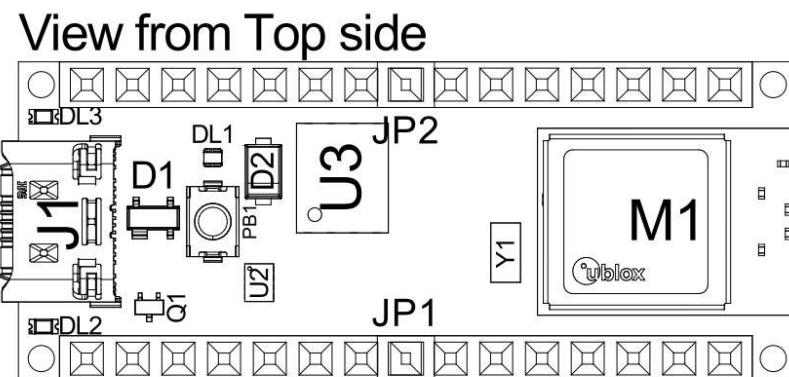
4 Block Diagram



Arduino Nano ESP32 Block Diagram

5 Board Topology

5.1 Front View



Top View of Arduino Nano ESP32

Ref.	Description
M1	NORA-W106-10B (ESP32-S3 SoC)
J1	CX90B-16P USB-C® connector
JP1	1x15 analog header
JP2	1x15 digital header
U2	MP2322GQH step down converter
U3	GD25B128EWIGR 128 Mbit (16 MB) ext. flash memory
DL1	RGB LED
DL2	LED SCK (serial clock)
DL3	LED Power (green)
D2	PMEG6020AELRX Schottky Diode
D3	PRTR5V0U2X,215 ESD Protection



6 NORA-W106-10B (Radio Module / MCU)

The Nano ESP32 features the **NORA-W106-10B** stand alone radio module, embedding an ESP32-S3 series SoC as well as an embedded antenna. The ESP32-S3 is based on an Xtensa® LX7 series microprocessor.

6.1 Xtensa® Dual-Core 32bit LX7 Microprocessor

The microprocessor for the ESP32-S3 SoC inside the NORA-W106 module is a dual-core 32-bit Xtensa® LX7. Each core can run at up to 240 MHz and has 512 kB SRAM memory. The LX7 features:

- 32-bit customized instruction set
- 128-bit data bus
- 32-bit multiplier / divider

The LX7 has a 384 kB ROM (Read Only Memory), and 512 kB of SRAM (Static Random Access Memory). It also features an 8 kB **RTC FAST** and **RTC SLOW** memory. These memories are designed for low-power operations, where the **SLOW** memory can be accessed by the ULP (Ultra Low Power) coprocessor, retaining the data in deep sleep mode.

6.2 Wi-Fi®

The NORA-W106-10B module supports the Wi-Fi® 4 IEEE 802.11 standards b/g/n, with an output power EIRP at up to 10 dBm. The max range for this module is 500 meters.

- 802.11b: 11 Mbit/s
- 802.11g: 54 Mbit/s
- 802.11n: 72 Mbit/s max at HT-20 (20 MHz), 150 Mbit/s max at HT-40 (40 MHz)

6.3 Bluetooth®

The NORA-W106-10B module supports Bluetooth® LE v5.0 with an output power EIRP at up to 10 dBm and data rates up to 2 Mbps. It has the option to scan and advertise simultaneously, as well as supporting multiple connections in peripheral/central mode.



6.4 PSRAM

The NORA-W106-10B module includes 8 MB of embedded PSRAM. (Octal SPI)

6.5 Antenna Gain

The built-in antenna on the NORA-W106-10B module uses GFSK modulation technique, with the performance ratings listed below:

Wi-Fi®:

- Typical conducted output power: **17 dBm**.
- Typical radiated output power: **20 dBm EIRP**.
- Conducted sensitivity: **-97 dBm**.

Bluetooth® Low Energy:

- Typical conducted output power: **7 dBm**.
- Typical radiated output power: **10 dBm EIRP**.
- Conducted sensitivity: **-98 dBm**.

This data is retrieved from the uBlox NORA-W10 data sheet (page 7, section 1.5) available here.

7 System

7.1 Resets

The ESP32-S3 has support for four levels of reset:

- **CPU**: resets CPU0/CPU1 core
- **Core**: resets the digital system, except for the RTC peripherals (ULP coprocessor, RTC memory).
- **System**: resets the entire digital system, including the RTC peripherals.
- **Chip**: resets the entire chip.

It is possible to conduct a software reset of this board, as well as obtaining the reset reason.

To do a hardware reset of the board, use the onboard reset button (PB1).



7.2 Timers

The Nano ESP32 has the following timers:

- 52-bit system timer with 2x 52-bit counters (16 MHz) and 3x comparators.
- 4x general-purpose 54-bit timers
- 3x watchdog timers, two in main system (MWDT0/1), one in the RTC module (RWDT).

7.3 Interrupts

All GPIOs on the Nano ESP32 can be configured to be used as interrupts, and is provided by an interrupt matrix. Interrupt pins are configured on an application level, using the following configurations:

- LOW
- HIGH
- CHANGE
- FALLING
- RISING

8 Serial Communication Protocols

The ESP32-S3 chip provides flexibility for the various serial protocols it supports. For example, the I2C bus can be assigned to almost any available GPIO.

8.1 Inter-Integrated Circuit (I2C)

Default pins:

- A4 - SDA
- A5 - SCL

The I2C bus is by default assigned to the A4/A5 (SDA/SCL) pins for retro compatibility. This pin assignment can however be changed, due to the flexibility of the ESP32-S3 chip.

The SDA and SCL pins can be assigned to most GPIOs, however some of these pins may have other essential functions that prevents I2C operations to run successfully.

Please note: many software libraries uses the standard pin assignment (A4/A5).



8.2 Inter-IC Sound (I2S)

There two I2S controllers that are typically used for communication with audio devices. There are no specific pins assigned for I2S, this can be used by any free GPIO.

Using standard or TDM mode, the following lines are used:

- **MCLK** - master clock
- **BCLK** - bit clock
- **WS** - word select
- **DIN/DOUT** - serial data

Using PDM mode:

- **CLK** - PDM clock
- **DIN/DOUT** serial data

Read more about the I2S protocol in Espressif's Peripheral API - InterIC Sounds (I2S)

8.3 Serial Peripheral Interface (SPI)

- **SCK** - D13
- **CPO** - D12
- **CPI** - D11
- **CS** - D10

The SPI controller is by default assigned to the pins above.

8.4 Universal Asynchronous Receiver/Transmitter (UART)

- **D1 / TX**
- **D0 / RX**

The UART controller is by default assigned to the the pins above.



8.5 Two Wire Automotive Interface (TWAI®)

The CAN/TWAI® controller is used to communicate with systems using the CAN/TWAI® protocol, particularly common in the automotive industry. There are no specific pins assigned for the CAN/TWAI® controller, any free GPIO can be used.

Please note: TWAI® is also known as the CAN2.0B, or "CAN classic". The CAN controller is **NOT** compatible with CAN FD frames.

9 External Flash Memory

Nano ESP32 features a 128 Mbit (16 MB) external flash, the GD25B128EWIGR (U3). This memory is connected to the ESP32 via Quad Serial Peripheral Interface (QSPI).

The operating frequency for this IC is 133 MHz, and has a data transfer rate at up to 664 Mbit/s.

10 USB Connector

The Nano ESP32 has one USB-C® port, used to power and program your board as well as sending & receiving serial communication.

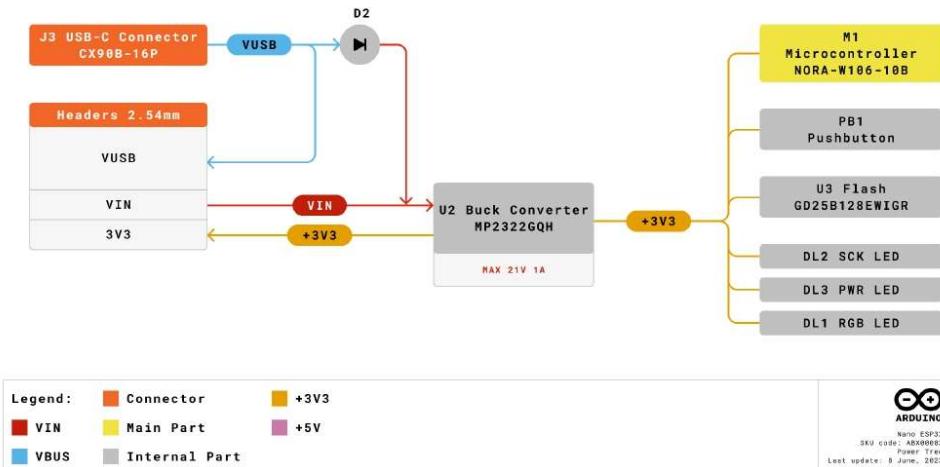
Note that you should not power the board with more than 5 V via the USB-C® port.

11 Power Options

Power can either be supplied via the VIN pin, or via USB-C® connector. Any voltage input either via USB or VIN is stepped down to 3.3 V using the MP2322GQH (U2) converter.

The operating voltage for this board is 3.3 V. Please note that there's no 5V pin available on this board, only the VBUS can provide 5 V when the board is powered via USB.

11.1 Power Tree



Arduino Nano ESP32 power tree.

11.2 Pin Voltage

All digital & analog pins on the Nano ESP32 are 3.3 V. Do not connect any higher voltage devices to any of the pins as it will risk damaging the board.

11.3 VIN Rating

The recommended input voltage range is **6-21 V**.

You should not attempt to power the board with a voltage outside the recommended range, particularly not higher than 21 V.

The efficiency of the converter depends on the input voltage via the VIN pin. See the average below for a board operation with normal current consumption:

- **4.5 V** - >90%.
- **12 V** - 85-90%
- **18 V** - <85%

This information is extracted from the MP2322GQH's datasheet.



11.4 VBUS

There is no 5V pin available on the Nano ESP32. 5 V can only be provided via the **VBUS**, which is supplied directly from the USB-C® power source.

While powering the board via the VIN pin, the VBUS pin is not activated. This means you have no option of providing 5 V from the board unless powered via USB or externally.

11.5 Using the 3.3 V Pin

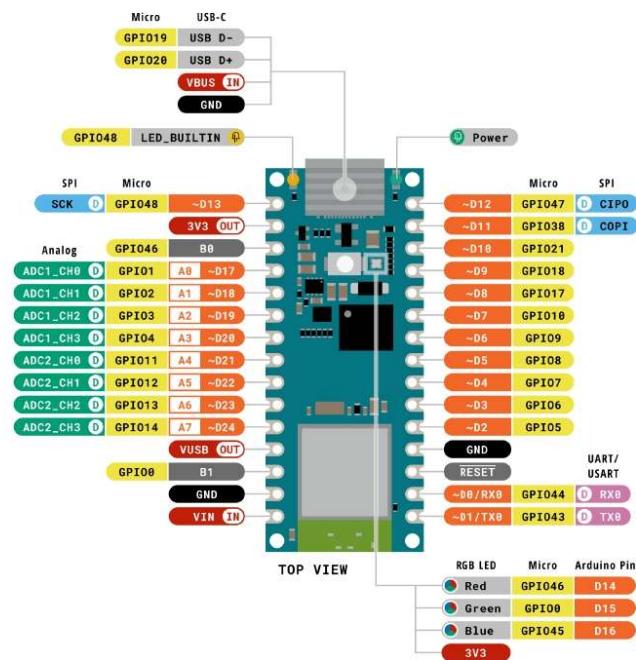
The 3.3 V pin is connected to the 3.3 V rail which is connected to the output of the MP2322GQH step down converter. This pin is primarily used to power external components.

11.6 Pin Current

The GPIOs on the Nano ESP32 can handle **source currents** up to **40 mA**, and **sink currents** up to **28 mA**. Never connect devices that draw higher current directly to a GPIO.

Mechanical Information

12 Pinout



Legend:	Digital	I2C	Other SERIAL
Power	(Orange)	(Blue)	(Yellow)
Ground	(Black)	(Grey)	(Purple)
Main Part	(Grey)	(Grey)	(Purple)
UART/USART			(Green)
PWM/Timer			(Green)

ARDUINO
Nano ESP32
SKU code: ABX00083
Pinout
Last update: 14 July, 2023

Pinout for Nano ESP32.



12.1 Analog (JP1)

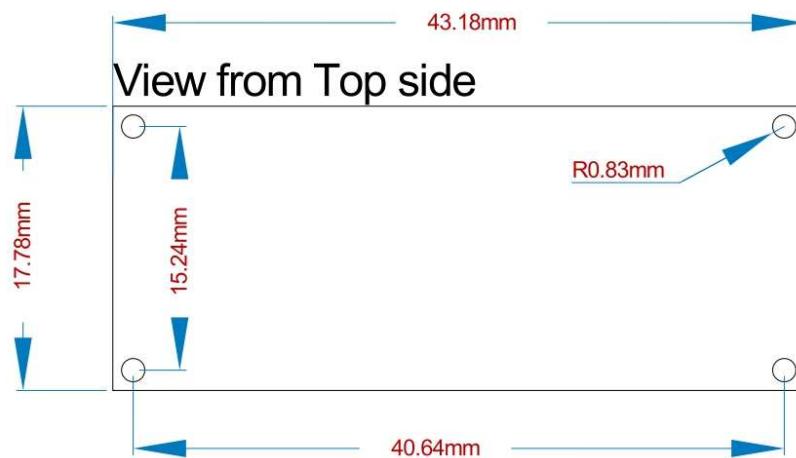
Pin	Function	Type	Description
1	D13 / SCK	NC	Serial Clock
2	+3V3	Power	+3V3 Power Rail
3	BOOT0	Mode	Board Reset 0
4	A0	Analog	Analog input 0
5	A1	Analog	Analog input 1
6	A2	Analog	Analog input 2
7	A3	Analog	Analog input 3
8	A4	Analog	Analog input 4 / I ² C Serial Datal (SDA)
9	A5	Analog	Analog input 5 / I ² C Serial Clock (SCL)
10	A6	Analog	Analog input 6
11	A7	Analog	Analog input 7
12	VBUS	Power	USB power (5V)
13	BOOT1	Mode	Board Reset 1
14	GND	Power	Ground
15	VIN	Power	Voltage Input

12.2 Digital (JP2)

Pin	Function	Type	Description
1	D12 / CIPO*	Digital	Controller In Peripheral Out
2	D11 / COPI*	Digital	Controller Out Peripheral In
3	D10 / CS*	Digital	Chip Select
4	D9	Digital	Digital pin 9
5	D8	Digital	Digital pin 8
6	D7	Digital	Digital pin 7
7	D6	Digital	Digital pin 6
8	D5	Digital	Digital pin 5
9	D4	Digital	Digital pin 4
10	D3	Digital	Digital pin 3
11	D2	Digital	Digital pin 2
12	GND	Power	Ground
13	RST	Internal	Reset
14	D0/RX	Digital	Digital pin 1 / Serial Receiver (RX)
15	D1/TX	Digital	Digital pin 0 / Serial Transmitter (TX)

*CIPO/COPI/CS replaces the MISO/MOSI/SS terminology.

13 Mounting Holes And Board Outline



Mechanical View of Nano ESP32

14 Board Operation

14.1 Getting Started – IDE

If you want to program your Nano ESP32 while offline you need to install the Arduino IDE [1]. To connect the Nano ESP32 to your computer, you will need a Type-C® USB cable, which can also provide power to the board, as indicated by the LED (DL1).



14.2 Getting Started – Arduino Cloud Editor

All Arduino boards, including this one, work out-of-the-box on the Arduino Cloud Editor [2], by just installing a simple plugin.

The Arduino Cloud Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow [3] to start coding on the browser and upload your sketches onto your board.

14.3 Getting Started – Arduino Cloud

All Arduino IoT enabled products are supported on Arduino Cloud which allows you to log, graph and analyze sensor data, trigger events, and automate your home or business.

14.4 Online Resources

Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on Arduino Project Hub [4], the Arduino Library Reference [5], and the online store [6]; where you will be able to complement your board with sensors, actuators and more.

14.5 Board Recovery

All Arduino boards have a built-in bootloader which allows flashing the board via USB. In case a sketch locks up the processor and the board is not reachable anymore via USB, it is possible to enter bootloader mode by double-tapping the reset button right after the power-up.

Certifications

15 Declaration of Conformity CE DoC (EU)

We declare under our sole responsibility that the products above are in conformity with the essential requirements of the following EU Directives and therefore qualify for free movement within markets comprising the European Union (EU) and European Economic Area (EEA).



16 Declaration of Conformity to EU RoHS & REACH 211

01/19/2021

Arduino boards are in compliance with RoHS 2 Directive 2011/65/EU of the European Parliament and RoHS 3 Directive 2015/863/EU of the Council of 4 June 2015 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

Substance	Maximum Limit (ppm)
Lead (Pb)	1000
Cadmium (Cd)	100
Mercury (Hg)	1000
Hexavalent Chromium (Cr6+)	1000
Poly Brominated Biphenyls (PBB)	1000
Poly Brominated Diphenyl ethers (PBDE)	1000
Bis(2-Ethylhexyl) phthalate (DEHP)	1000
Benzyl butyl phthalate (BBP)	1000
Dibutyl phthalate (DBP)	1000
Diisobutyl phthalate (DIBP)	1000

Exemptions : No exemptions are claimed.

Arduino Boards are fully compliant with the related requirements of European Union Regulation (EC) 1907 /2006 concerning the Registration, Evaluation, Authorization and Restriction of Chemicals (REACH). We declare none of the SVHCs (<https://echa.europa.eu/web/guest/candidate-list-table>), the Candidate List of Substances of Very High Concern for authorization currently released by ECHA, is present in all products (and also package) in quantities totaling in a concentration equal or above 0.1%. To the best of our knowledge, we also declare that our products do not contain any of the substances listed on the "Authorization List" (Annex XIV of the REACH regulations) and Substances of Very High Concern (SVHC) in any significant amounts as specified by the Annex XVII of Candidate list published by ECHA (European Chemical Agency) 1907 /2006/EC.



17 Conflict Minerals Declaration

As a global supplier of electronic and electrical components, Arduino is aware of our obligations with regards to laws and regulations regarding Conflict Minerals, specifically the Dodd-Frank Wall Street Reform and Consumer Protection Act, Section 1502. Arduino does not directly source or process conflict minerals such as Tin, Tantalum, Tungsten, or Gold. Conflict minerals are contained in our products in the form of solder, or as a component in metal alloys. As part of our reasonable due diligence Arduino has contacted component suppliers within our supply chain to verify their continued compliance with the regulations. Based on the information received thus far we declare that our products contain Conflict Minerals sourced from conflict-free areas.

18 FCC Caution

Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

- (1) This device may not cause harmful interference
- (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC RF Radiation Exposure Statement:

1. This Transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
2. This equipment complies with RF radiation exposure limits set forth for an uncontrolled environment.
3. This equipment should be installed and operated with a minimum distance of 20 cm between the radiator & your body.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

English: User manuals for licence-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both. This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions:

- (1) this device may not cause interference



(2) this device must accept any interference, including interference that may cause undesired operation of the device.

French: Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

(1) l' appareil nedoit pas produire de brouillage

(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

IC SAR Warning:

English This equipment should be installed and operated with a minimum distance of 20 cm between the radiator and your body.

French: Lors de l' installation et de l' exploitation de ce dispositif, la distance entre le radiateur et le corps est d'au moins 20 cm.

Important: The operating temperature of the EUT can't exceed 85°C and shouldn't be lower than -40 °C.

Hereby, Arduino S.r.l. declares that this product is in compliance with essential requirements and other relevant provisions of Directive 2014/53/EU. This product is allowed to be used in all EU member states.

19 SRRC

This equipment contains a radio transmitter module with model approval code: CMIIT ID: 24J993CLD252.

20 Company Information

Company name	Arduino S.r.l.
Company Address	Via Andrea Appiani, 25 Monza, MB, 20900 Italy



21 Reference Documentation

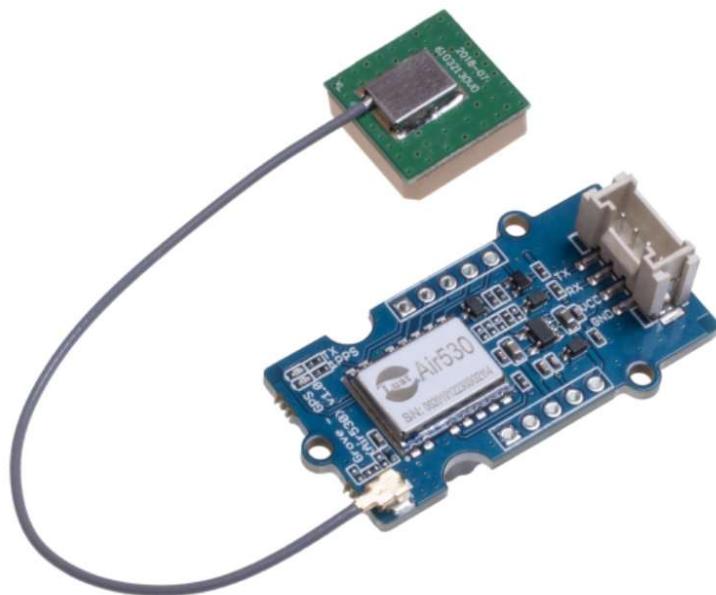
Reference	Link
Arduino IDE (Desktop)	https://www.arduino.cc/en/Main/Software
Arduino Cloud Editor	https://create.arduino.cc/editor
Arduino Cloud Editor - Getting Started	https://docs.arduino.cc/arduino-cloud/guides/editor/
Arduino Project Hub	https://create.arduino.cc/projecthub?by=part&part_id=11332&sort=trending
Library Reference	https://github.com/arduino-libraries/
Online Store	https://store.arduino.cc/

22 Change Log

Date	Changes
08/06/2023	Release
09/01/2023	Update power tree flowchart.
09/11/2023	Update SPI section, update analog/digital pin section.
11/06/2023	Correct company name, correct VBUS/VUSB
11/09/2023	Block Diagram Update, Antenna Specifications
11/15/2023	Ambient temperature update
11/23/2023	Added label to LP modes
23/02/2024	Added antenna frequency to block diagram
25/04/2024	Updated link to new Cloud Editor
23/08/2024	Added SRRC certification
23/08/2024	Cloud Editor updated from Web Editor

Bijlage C8 Datasheet GROVE – GPS (AIR530) Seeed Studio

Grove - GPS (Air530) - Seeed Studio



Grove - GPS (Air530)

SKU

109020022

Grove-GPS (Air530) is a high-performance, highly integrated multi-mode satellite positioning and navigation module, supporting plenty of location systems, such as GPS/Beidou/Galileo and etc.

PRODUCT DETAILS

Feature

- Positioning error within 10 meters
- Multi-mode satellite positioning & navigation and support more than 6 satellites at the same time
- The position still quickly and accurately under the condition of bad signal
- Low power consumption at only 31uA
- Compatible hard/soft interface for other modules

1/5

- Adopt the integrated design of RF baseband, which integrates DC/DC, LDO, LNA, RF front-end, baseband processing, 32-bit RISC based chip, RAM, FLASH storage, RTC and power management functions

Description

Confused about your GPS not working well in urban or outdoors under only one or a few satellite module? Then you should not miss our new Grove-GPS (Air530). It's a high-performance, highly integrated multi-mode satellite positioning and navigation module to use with Arduino. It supports GPS / Beidou / Glonass / Galileo / QZSS / SBAS, which makes it suitable for GNSS positioning applications such as car navigation, smart wear, and drone. Meanwhile, this module is capable of receiving more than 6 satellites at the same time and is able to work excellently even if there's a very bad signal.

Grove – GPS Module vs Grove – GPS (Air530)

Specs	<u>Grove – GPS Module</u>	<u>Grove – GPS (Air530)</u>
Size	40mm x 20mm x 13mm	40mm x 20mm x 13mm
Update Rate	1Hz, max 10Hz	–
Baud Rate	9,600 – 115,200	9600 – 921600
Navigation Sensitivity	-160dBm	-166dBm
Power Requirements	3.3/5V	3.3/5V
Number of Channels	22 tracking, 66 channels	–
Time to first start	Cold start: 13s Warm start: 1-2s Hot start: <1s	Cold start: 30 seconds Warm Start: 4 seconds
Antennas	Antenna included	Antenna included
Accuracy	2.5m GPS Horizontal Position Accuracy	2.5m Horizontal positioning accuracy

Specification

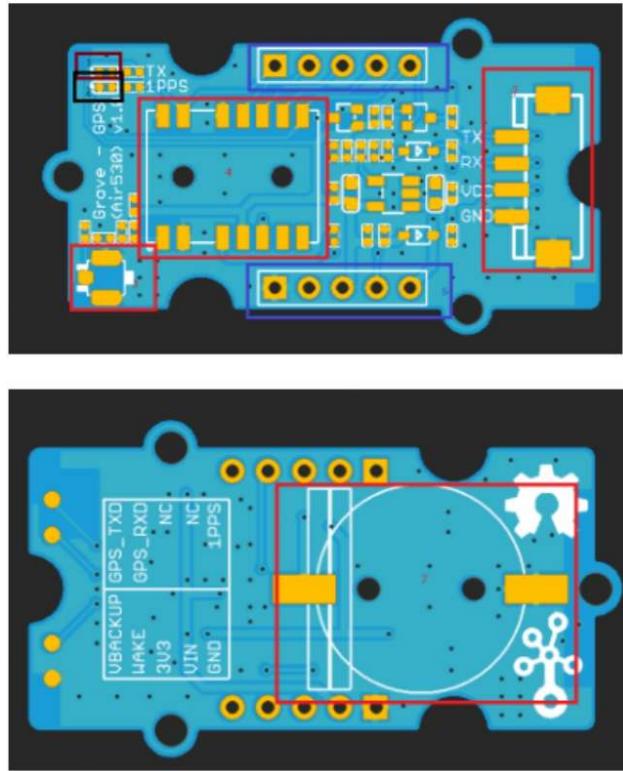
parameters	values
Supply voltage	3.3V/5V
Working current	up to 60mA

parameters	values
Interface	UART
Time of warm start	4s
Time of cold boot	30s
Working Temperature	-35C-85°C
Storage Temperature	-55°C-100°C
Humidity	5%-95%
Time of Positioning	Pure hardware cold start
	<1S
	Pure hardware hot start
	<5S
Sensitivity	Pure hardware recapture
	Cold start
	Hot Start
	Recapture
Accuracy	Track
	Horizontal positioning accuracy
	High positioning accuracy
	Speed accuracy
Consumption	Time transfer accuracy
	2.5m
	3.5m
	0.1m/s
	Capture current value@3.3V
	Tracking current value@3.3V
	Low power mode@3.3V
	Ultra-low-power mode@3.3V
	42.6mA
	36.7mA
	0.85mA
	31uA

Note

The test above is taken under bi-modules of GPS&Beidou.

Pinout



1. AIR530 TX Indicator (BLUE LED);
2. 1PPS Indicator (GREEN LED);
3. U.FL Antenna Interface;
4. AIR530 Module;
5. Grove Interface;
6. AIR530 Headers;
7. 3V Button Battery (CR1220).

Applications

- GPS tracker
- GPS navigation
- Distance measurement
- Car navigation
- Drone

Part list

Items	Values
GPS (Air 530) board	1
Grove cable	1

Cautions

- The input power supply voltage of Grove power supply interface, should not exceed 5.6V, otherwise it may damage the device.
- The voltage of the selected button battery should not exceed 3V, or it will probably damage the device.

ECCN/HTS

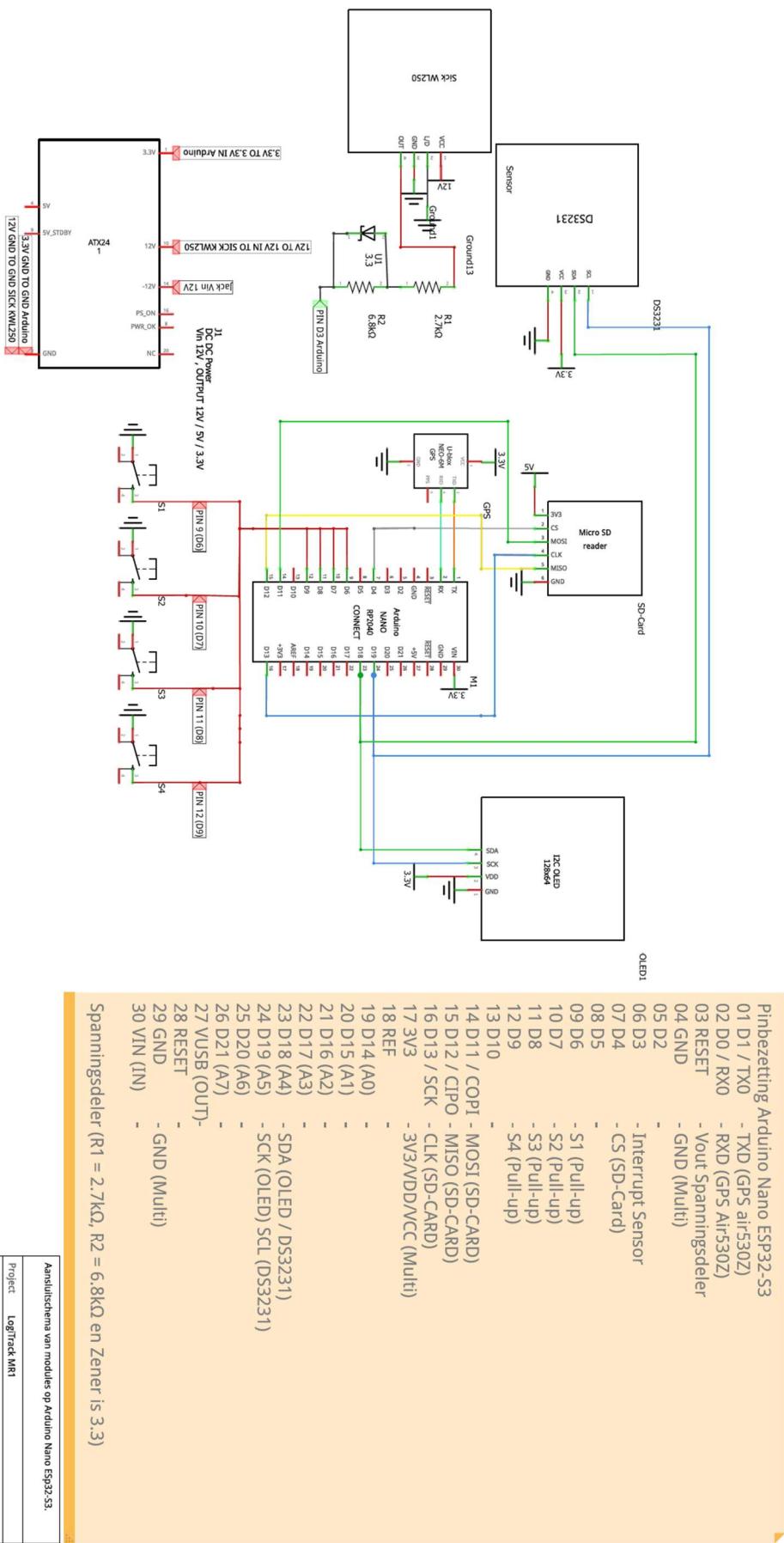
HSCODE 8526919090

UPC

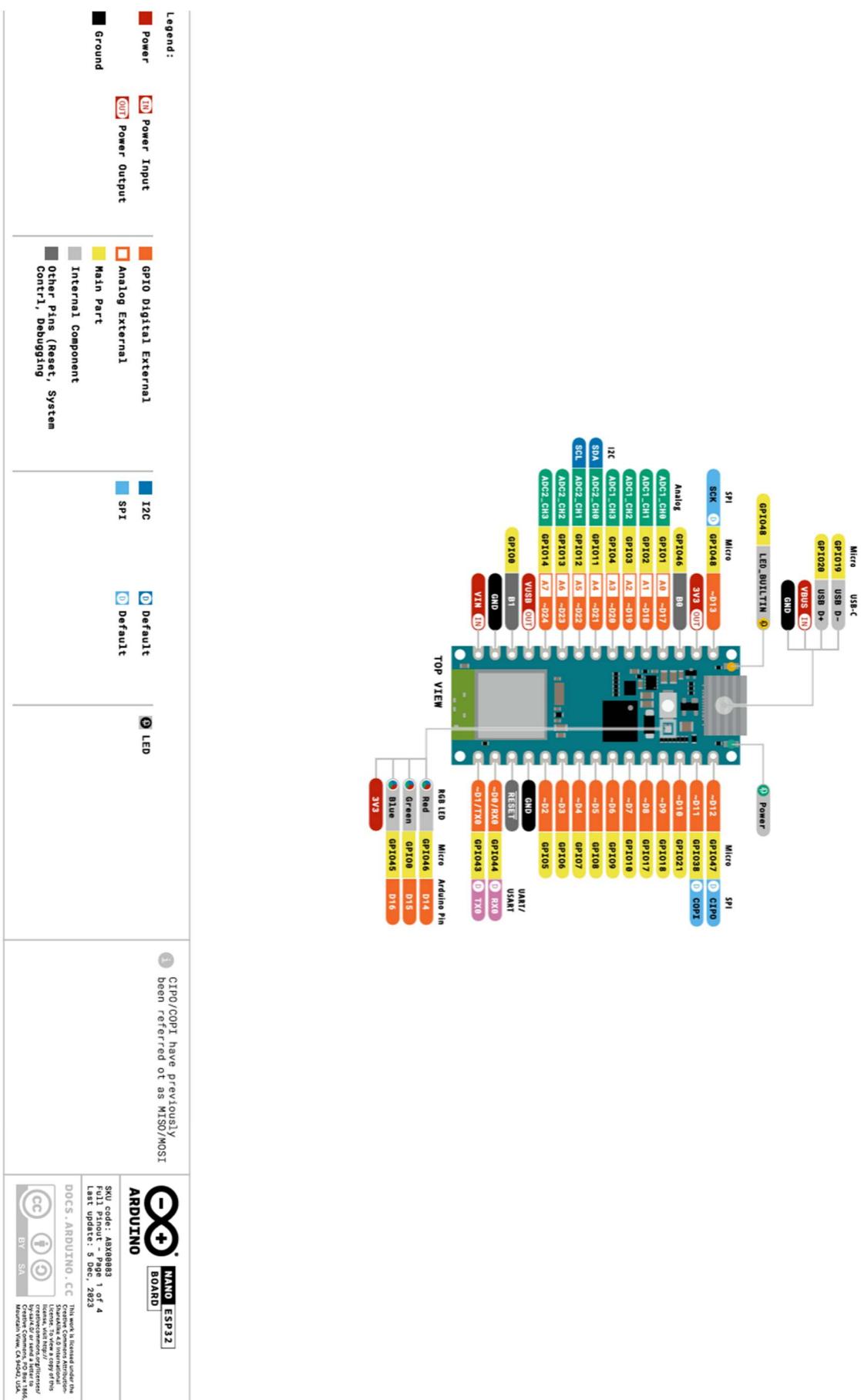
Bijlage C9 Verwijzingen Onderzoek & analyse

- Arduino. (2025). Arduino IDE. *Software Arduino IDE downloads*.
- 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>
- HAN. (2025). <https://han.onderwijsonline.nl/elearning/lesson/Vq32z9kD>. Opgehaald van <https://han.onderwijsonline.nl>.
- <https://www.logmore.com/post/what-is-data-logging>. (sd). Opgehaald van
<https://www.logmore.com/>: <https://www.logmore.com/post/what-is-data-logging#:~:text=Data%20logging%20is%20the%20process,regardless%20of%20the%20method%20used>.
- 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 D Elektrisch schema



Bijlage D1 Pinout Arduino Nano ESP32-S3



Bijlage E Plan van aanpak testen prototype LogiTrack MR1

Doel van de testen:

Het doel is om vast te stellen of de LogiTrack MR1 werkt zoals ontworpen en wat de mate van betrouwbaarheid is, vergeleken met andere dataloggers in dienst bij de politie.

Om dit vast te stellen zal de LogiTrack worden vergeleken met twee dataloggers die in bezit zijn bij het team Forensische Opsporing Zeeland – West – Brabant en Rotterdam.

Het team Forensische Opsporing Rotterdam heeft de beschikking over de MoviLog DL1 PRO.

Het team Forensische Opsporing Zeeland West Brabant heeft de beschikking over de DataTransition.

De testen worden uitgevoerd door alle drie de systemen tegelijkertijd op één voertuig te monteren en met een constante snelheid voorbij reflectiepaaltjes te rijden. In totaal worden er vijf testen uitgevoerd die ieder uit vijf ritten bestaan. Vervolgens wordt geanalyseerd of er verschil is in tijdstempels en waar dit verschil door veroorzaakt is/wordt.

Waar en wanneer:

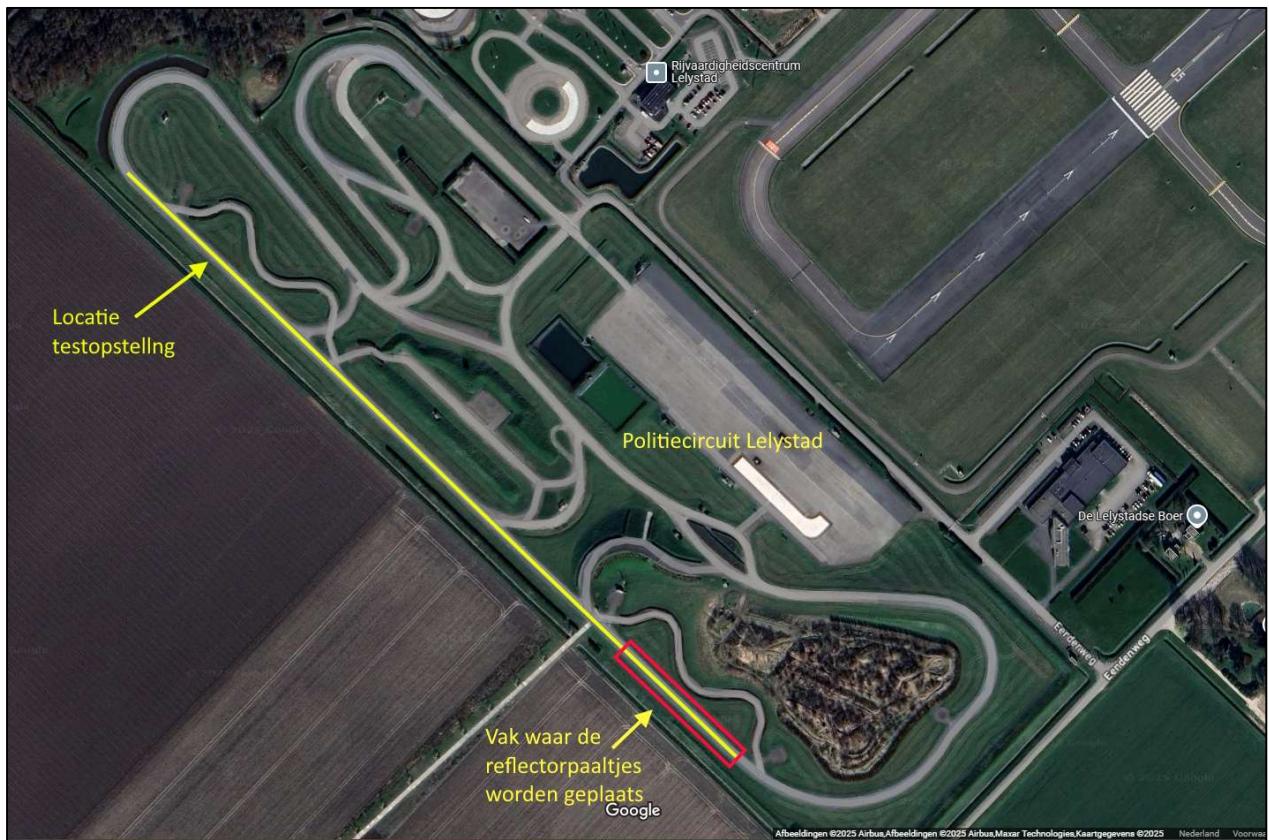
- Op vrijdag 9 mei 2025 is de grote omloop op het circuit van de politieacademie gelegen aan de Eendenweg 1 te Lelystad gereserveerd van 10 tot 12 uur;

Materialen:

- Prototype datalogger (LogiTrack MR1 inclusief afstandssensor SICK WLP);
- Datalogger FO ZWB (DAQ)
- Datalogger FO RTD (Mov1Log BM22);
- Referentievoertuig (Seat Leon);
- Meetlint of distometer (om de afstanden te bepalen tussen de reflectorpaaltjes);
- Poetsdoek en reinigingsmiddel (buitenzijde van de auto stofvrij te maken);
- Gereedsschapset en ducttape (om reparaties uit te voeren);
- Instructie hoe de dataloggers werken;
- Reflectorpaaltjes (Bij voorkeur vier stuks);
- Stroom of verlengkabel (Omdat de DAQ en de LogiTrack MR1 12 Volt voeding nodig hebben).

Hoe te werk gaan?

- Voor 10 uur, zijn alle drie de loggers aan de rechterzijde van de personenauto bevestigd (bij voorkeur naast elkaar) en de datalogger zijn gebruiksklaar;
- Om 10 uur, de baan op;
- Met kegels wordt de aanvang van de testopstelling gemarkeerd alsmede het einde van de testopstelling (zie schermafdruk 1);
- Binnen het vak worden de reflectorpaaltjes geplaatst aan tegen de as van de weg aan;
- Iedere test bestaat uit 5 ritten met verschillende rijsnelheden 40, 50, 60 ,70 en 80 km/h;
- Bij het rijden van de rit wordt er uitsluitend met een constante snelheid gereden op cc;
- Bij de testen worden reflectiepaaltjes gevarieerd van elkaar geplaatst:
 1. de eerste test 20 meter uit elkaar geplaatst;
 2. de tweede test 15 meter uit elkaar geplaatst;
 3. de derde test 10 meter uit elkaar geplaatst;
 4. de vierde test 5 meter uit elkaar geplaatst;
 5. de vijfde test 1 meter uit elkaar geplaatst;



Schermafdruk 1: Overzichtskaart van het circuit lelystad.

Bijlage F Testuitslagen

Rit 1, 2025-05-09, 11:16:17	LogiTrack	DAQ	Verschil	Afstand	Snelheid	Nummer
11:16:47.703						
11:16:49.224	1,521	1,530	0,009	15	40	1
11:16:50.757	1,533	1,525	0,008	15	40	2
11:16:52.212						
Rit 2, 2025-05-09, 11:18:06						
11:18:32.348						
11:18:33.578	1,23	1,214	0,016	15	50	3*
11:18:34.775	1,197	1,208	0,011	15	50	4*
11:18:35.926	1,151	1,148	0,003	15	50	5
Rit 3, 2025-05-09, 11:19:50						
11:20:09.133						
11:20:10.139	1,006	1,016	0,010	15	60	6
11:20:11.157	1,018	1,009	0,009	15	60	7
11:20:12.108	0,951	0,960	0,009	15	60	8
Rit 4, 2025-05-09, 11:21:41						
11:22:01.011						
11:22:01.896	0,885	0,871	0,014	15	70	9
11:22:02.748	0,852	0,868	0,016	15	70	10
11:22:03.580	0,8	0,826	0,026	15	70	11
Rit 5, 2025-05-09, 11:23:27						
11:23:43.784						
11:23:44.551	0,767	0,772	0,005	15	80	12
11:23:45.330	0,779	0,769	0,010	15	80	13
11:23:46.055	0,725	0,732	0,007	15	80	14
11:24:29.412						
Rit 6, 2025-05-09, 11:26:58 bij 10M						
11:27:22.738						
11:27:23.743	1,005	1,006	0,001	10	40	15
11:27:24.761	1,018	1,000	0,018	10	40	16*
11:27:25.748	0,987	1,000	0,013	10	40	17*
Rit 7, 2025-05-09, 11:28:48						
11:29:10.822						
11:29:11.628	0,806	0,807	0,001	10	50	18
11:29:12.432	0,804	0,802	0,002	10	50	19*
11:29:13.231	0,799	0,802	0,003	10	50	20*
Rit 8, 2025-05-09, 11:30:35						
11:30:51.517						

11:30:52.190	0,673	0,680	0,007	10	60	21
11:30:52.862	0,672	0,674	0,002	10	60	22
11:30:53.526	0,664	0,673	0,009	10	60	23*

Rit 9, 2025-05-09, 11:32:21

11:32:37.654	0,58	0,582	0,002	10	70	24
11:32:38.234	0,567	0,579	0,012	10	70	25*
11:32:38.801	0,592	0,579	-0,013	10	70	26*

Rit 10, 2025-05-09, 11:34:37

11:34:52.275	0,514	0,520	0,006	10	80	27
11:34:52.789	0,513	0,516	0,003	10	80	28
11:34:53.302	0,512	0,514	0,002	10	80	29

Rit 11, 2025-05-09, 11:37:36

11:37:55.053	0,514	0,507	0,007	5	40	30
11:37:55.567	0,487	0,505	0,018	5	40	31*
11:37:56.080	0,565	0,574	0,009	5	40	32*

Rit 12, 2025-05-09, 11:39:57

11:40:18.684	0,382	0,407	0,025	5	50	33
11:40:18.474	0,408	0,405	0,003	5	50	34
11:40:18.934	0,46	0,460	0,000	5	50	35

Rit 13, 2025-05-09, 11:42:21

11:42:35.485	0,328	0,340	0,012	5	60	36
11:42:35.813	0,327	0,339	0,012	5	60	37
11:42:36.140	0,408	0,385	0,024	5	60	38

Rit 14, 2025-05-09, 11:44:33

11:44:45.399	0,316	0,294	0,022	5	70	39
11:44:49.346	0,275	0,293	0,018	5	70	40
11:44:49.662	0,327	0,333	0,006	5	70	41

Rit 15, 2025-05-09, 11:47:24

11:48:01.807	0,275	0,259	0,016	5	80	42
11:48:02.082	0,235	0,258	0,023	5	80	43
11:48:02.317	0,302	0,293	0,009	5	80	44

Rit 16, 2025-05-09, 11:51:15

11:51:32.494

11:51:32.584	0,09	0,103	0,013	1	40	45
11:51:32.699	0,115	0,103	0,012	1	40	46
11:51:32.788	0,089	0,101	0,012	1	40	47

Rit 17, 2025-05-09, 11:52:28

11:52:41.522

11:52:41.612	0,09	0,083	0,007	1	50	48
11:52:41.685	0,073	0,083	-0,010	1	50	49
11:52:41.773	0,088	0,082	0,006	1	50	50

Rit 18, 2025-05-09, 11:53:37

11:53:50.679

11:53:50.769	0,09	0,069	0,021	1	60	51
11:53:50.831	0,062	0,069	0,007	1	60	52
11:53:50.893	0,062	0,068	0,006	1	60	53

Rit 19, 2025-05-09, 11:54:44

11:54:53.739

11:54:53.802	0,063	0,060	0,003	1	70	54
11:54:53.864	0,062	0,060	0,002	1	70	55*
11:54:53.926	0,062	0,059	0,003	1	70	56

Rit 20, 2025-05-09, 11:55:56

11:56:08.826

11:56:08.889	0,063	0,053	0,010	1	80	57
11:56:08.959	0,07	0,053	0,017	1	80	58
11:56:08.994	0,035	0,052	0,017	1	80	59

Rit 21, 2025-05-09, 11:57:44

11:57:53.841

11:57:53.904 0,039 1 100
0,039 1 100

Rit 22, 2025-05-09, 11:59:19

11:59:36.735

11:59:36.825 0,041 1 100
0,041 1 100

Rit 23, 2025-05-09, 12:01:48

12:02:03.464

12:02:03.607	0,143	0,127	0,016	3	120	60
12:02:03.722	0,115	0,129	0,014	3	120	61
12:02:03.863	0,141	0,127	0,014	3	120	62

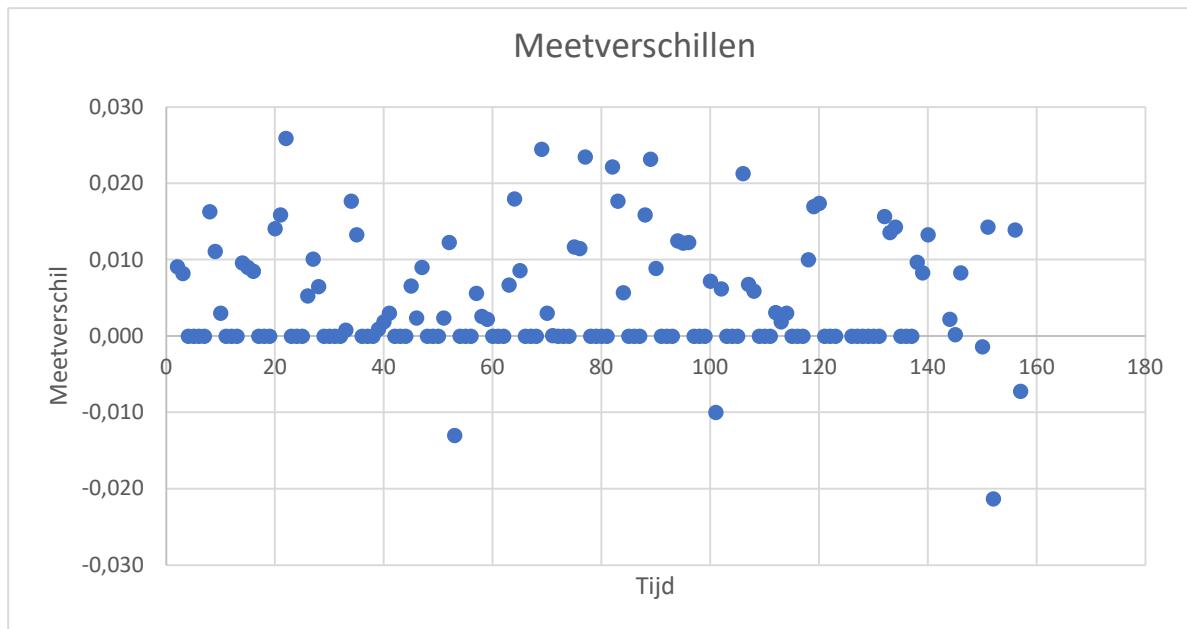
Rit 24, 2025-05-09, 12:03:39

	LogiTrack	DAQ	Gemiddelde verschil in ms			
12:03:54.265						
12:03:54.381	0,116	0,106	0,010	3	120	63
12:03:54.496	0,115	0,107	0,008	3	120	64
12:03:54.587	0,091	0,104	0,013	3	120	65
Rit 25, 2025-05-09, 11:01:25						
11:01:43.190						
11:01:44.605	1,415	1,413	0,002	20	60	66
11:01:45.956	1,351	1,351	0,000	20	60	67
11:01:47.251	1,295	1,287	0,008	20	60	68
Rit 26, 2025-05-09, 11:03:09						
11:03:22.136						
11:03:24.351	1,215	1,216	-0,001	20	70	69
11:03:25.528	1,177	1,163	0,014	20	70	70
11:03:26.614	1,086	1,107	-0,021	20	70	71
Rit 27, 2025-05-09, 11:05:23						
11:05:42.553				20	80	
11:05:43.635	1,082	1,068	0,014	20	80	72
11:05:44.653	1,018	1,025	-0,007	20	80	73
11:05:45.654						
Totaal	42,204	42,257	0,009			73 Metingen
				Grootste verschil is 0,026 ms (Meting 11)		

Tabel 9: Meetgegevens van de LogiTrack en de DAQ.

*Betreffen de metingen waarbij de seconde handmatig werd aangepast.

Rit 21 en 22, werden uitgesloten van analyse. Omdat de LogiTrack maar 2 van de 4 reflectoren had waargenomen. Het is achteraf niet vast te stellen welke detecties er precies hebben plaatsgevonden.



Grafiek 1: Overzicht meetafwijkingen.

De grafiek van de meetverschillen tussen de DAQ en de LogiTrack MR1. Op de X-as staan de tijden en op de Y-as zijn de meetverschillen in ms weergegeven.

Ref fast 1038 1030 dipro.
Ref fast 1048.

Test 1 (Reflectie op 20 meter van elkaar)

Test	Tijd start	Tijd stop	Bijzonderheden
Rit 1 -> 40 km/h	1033		
Rit 2 -> 50 km/h	1038	1030	40 → 50
Rit 3 -> 60 km/h	1101		
Rit 4 -> 70 km/h	1103		
Rit 5 -> 80 km/h	1105		

1030
1031
1032
1033
1034
1035
1036

Test 2 (Reflectie op 15 meter van elkaar)

Test	Tijd start	Tijd stop	Bijzonderheden
Rit 1 -> 40 km/h	1116-	1117-	
Rit 2 -> 50 km/h	1118	1119	
Rit 3 -> 60 km/h	1119	1120	
Rit 4 -> 70 km/h	1121	1122	
Rit 5 -> 80 km/h	1122	1123	

1032 rit 1.
1033 rit 2
1035 rit 4.
1036 rit 5

Test 3 (Reflectie op 10 meter van elkaar)

Test	Tijd start	Tijd stop	Bijzonderheden
Rit 1 -> 40 km/h	1127-		
Rit 2 -> 50 km/h	1128	1129	Dipro niet!
Rit 3 -> 60 km/h	1129	1130	
Rit 4 -> 70 km/h	1132		
Rit 5 -> 80 km/h	1134-		

1032 rit 6
1033 8

Test 4 (Reflectie op 5 meter van elkaar)

Test	Tijd start	Tijd stop	Bijzonderheden
Rit 1 -> 40 km/h	1137-	1138-	
Rit 2 -> 50 km/h	1140	1140	
Rit 3 -> 60 km/h	1142		
Rit 4 -> 70 km/h	1143-	1144-	lud2 detectie op achter!
Rit 5 -> 80 km/h	1147-	1148	

1032 rit 11
1033 12

Test 5 (Reflectie op 1 meter van elkaar)

Test	Tijd start	Tijd stop	Bijzonderheden
Rit 1 -> 40 km/h	1150		
Rit 2 -> 50 km/h	1152		
Rit 3 -> 60 km/h	1153		
Rit 4 -> 70 km/h	1154		
Rit 5 -> 80 km/h	1155		

1032 rit 16

Test 6 (Reflectie op meter van elkaar)

Test	Tijd start	Tijd stop	Bijzonderheden
Rit 1 -> km/h	1157		100km
Rit 2 -> km/h	1159		100km
Rit 3 -> km/h	1201		3 meter
Rit 4 -> km/h	1202	120 km/h.	3 meter
Rit 5 -> km/h			

n>2
lossd r1/22
loss34 n22

100km
100km
100km

Test 7 (Reflectie op meter van elkaar)

Test	Tijd start	Tijd stop	Bijzonderheden
Rit 1 -> km/h			
Rit 2 -> km/h			
Rit 3 -> km/h			
Rit 4 -> km/h			
Rit 5 -> km/h			

Test 8 (Reflectie op meter van elkaar)

Test	Tijd start	Tijd stop	Bijzonderheden
Rit 1 -> km/h			
Rit 2 -> km/h			
Rit 3 -> km/h			
Rit 4 -> km/h			
Rit 5 -> km/h			

Test 9 (Reflectie op meter van elkaar)

Test	Tijd start	Tijd stop	Bijzonderheden
Rit 1 -> km/h			
Rit 2 -> km/h			
Rit 3 -> km/h			
Rit 4 -> km/h			
Rit 5 -> km/h			

Test 6 (Reflectie op meter van elkaar)

Test	Tijd start	Tijd stop	Bijzonderheden
Rit 1 -> km/h			
Rit 2 -> km/h			
Rit 3 -> km/h			
Rit 4 -> km/h			
Rit 5 -> km/h			

Bijlage G Acceptatietesten

Deze bijlage beschrijft de testscenario's tot in detail hoe de acceptatietesten werden uitgevoerd.

Testscenario 1: Starten en stoppen van meting

Benodigdheden:

- Datalogger
 - Hardware versie 1.0
 - Software versie 1.0

Geteste functionele eisen:

- F1

Test uitgevoerd:

- 18 mei 2025 door M. van Riel

Stappen:

01	Schakel de datalogger in. Het OLED-scherm gaat aan en het politie-logo verschijnt met daaronder de naam en het type van de datalogger "LogiTrach MR1". Hierna wordt het hoofdmenu weergegeven op het OLED-scherm. In de userinfo wordt weergegeven of de SD-kaart gereed is met daaronder de datum en de tijd van de RTC DS3132. Aan de rechterzijde staat informatie over de SD-kaart, de afstandssensor en het aantal ritten. Onderin de functies van de drukknoppen. Waargenomen zoals beschreven.	✓
02	Plaats een SD-kaart in de SD-kaartmodule. Op het OLED-scherm wordt in de userinfo de tekst afgedrukt dat de SD gereed is. Rechts op het OLED-scherm wijzigt de "X" achter SD naar een "O". Waargenomen zoals beschreven.	✓
03	Sluit de afstandssensor aan op de datalogger en houdt een reflecterend voorwerp op 20 cm voor het oog van de afstandssensor. Op het OLED-scherm wijzigt de "X" achter IR, 0,5 seconden naar een "O" indien een detectie heeft plaatsgevonden. Hierna verschijnt weer een "X". Waargenomen zoals beschreven.	✓
04	Druk eenmaal op de "drukknop Start" en laat los. De meting wordt gestart. Op het OLED-scherm verschijnt de tekst in de userinfo-kader: "Meting gestart". Het aantal ritten wordt met één opgehoogd. Waargenomen zoals beschreven.	✓
05	Druk eenmaal op de "drukknop Stop" en laat los. De meting wordt gestopt, op het OLED-scherm verschijnt de tekst in de userinfo-kader: "Meting gestopt". Daarboven staat de tekst hoeveel	✓

	<p>detecties er hebben plaatsgevonden. Helemaal bovenaan staat de tekst of de opslag op de SD-kaart is gelukt.</p> <p>Waargenomen zoals beschreven.</p>	
--	---	--

Conclusie: ✓ Test geslaagd.

Testscenario 2: Weergeven van de tijd en wijzigen van de waardes

Benodigdheden:

- Datalogger
 - Hardware versie 1.0
 - Software versie 1.0

Geteste functionele eisen:

- F2

Test uitgevoerd:

- 18 mei 2025 door M. van Riel

Stappen:

01	<p>Schakel de datalogger in.</p> <p>In het scherm worden tijd en datum onder elkaar weergegeven: 00:00:00 2000-01-01</p> <p>Waargenomen zoals beschreven</p>	✓
02	<p>Druk eenmaal op de “drukknop Klok” en laat los.</p> <p>Het hoofdmenu maakt plaats voor het klokmenu. Bovenin het OLED-scherm staat de tekst: “Stel tijd in:”. Twee regels daaronder staan de datum en de tijd.</p> <p>Waargenomen zoals beschreven</p>	✓
03	<p>Druk eenmaal op de “drukknop +” en laat los.</p> <p>De huidige waarde met contrasterende achtergrond wordt verhoogd met één.</p> <p>Waargenomen zoals beschreven.</p>	✓
04	<p>Druk eenmaal op de “drukknop –“ en laat los.</p> <p>De huidige waarde met contrasterende achtergrond wordt verlaagd met één.</p> <p>Waargenomen zoals beschreven.</p>	✓
05	<p>Druk eenmaal op de “drukknop >” en laat los.</p> <p>De cursor verspringt naar de volgende waarde rechts.</p> <p>Waargenomen zoals beschreven.</p>	✓

06	<p>Druk eenmaal op de drukknop OK en laat los.</p> <p>De ingestelde tijd wordt opgeslagen. Het OLED-scherm keert terug naar het hoofdmenu. In de userinfo verschijnen onder elkaar de meldingen: "Tijd aangepast", "Tijd: 00:00:00", "Dat: 2000-01-01".</p> <p>Waargenomen zoals beschreven.</p>	✓
----	--	---

Conclusie: ✓

Test geslaagd.

Testscenario 3: Errors rond metingen

Benodigdheden:

- Datalogger
 - Hardware versie 1.0
 - Software versie 1.0

Geteste functionele eisen:

- F3

Test uitgevoerd:

- 18 mei 2025 door M. van Riel

Stappen:

01	<p>Schakel de datalogger in zonder SD-kaart.</p> <p>Druk eenmaal op de "drukknop Start". In de userinfo verschijnt de tekst: "Plaats SD-kaart", tevens wordt er geen meting gestart.</p> <p>Waargenomen zoals beschreven</p>	✓
02	<p>Schakel de datalogger in. Breng SD-kaart in de SD-kaartmodule.</p> <p>Druk eenmaal op "drukknop start" zodat de meting wordt gestart. Verwijder tijdens de meting de SD-kaart. De meting wordt gestopt en in de userinfo verschijnen onder elkaar de tekst: "SD verwijderd!" en "Metting gestopt!"</p> <p>Waargenomen zoals beschreven.</p>	✓
03	<p>Schakel de datalogger in. Breng SD-kaart in de SD-kaartmodule.</p> <p>Druk eenmaal op de "drukknop stop". Op het OLED-scherm verschijnt in de userinfo-kader: "Metting was al gestopt".</p> <p>Waargenomen zoals beschreven.</p>	✓
04	<p>Schakel de datalogger in. Breng SD-kaart in de SD-kaartmodule.</p> <p>Druk eenmaal op "drukknop start" zodat de meting gaat lopen. Druk nu eenmaal op de "drukknop Klok". Op het OLED-scherm verschijnt in de userinfo-kader: "Niet beschikbaar".</p> <p>Waargenomen zoals beschreven.</p>	✓
05	<p>Schakel de datalogger in. Breng SD-kaart in de SD-kaartmodule.</p>	✓

	Druk eenmaal op “drukknop start” zodat de meting gaat lopen. Druk nu eenmaal op de “drukknop GPS”. Op het OLED-scherm verschijnt in de userinfo-kader: “Niet beschikbaar”. Waargenomen zoals beschreven.	
--	---	--

Conclusie: ✓

Test geslaagd.

Testscenario 4: Errors instellen klok

Benodigdheden:

- Datalogger
 - Hardware versie 1.0
 - Software versie 1.0

Geteste functionele eisen:

- F4

Test uitgevoerd:

- 18 mei 2025 door M. van Riel

Stappen:

01	Schakel de datalogger in en ga naar het klokmenu. Stel de dag in op “32”, druk eenmaal op de “drukknop OK”. Op het scherm verschijnt “Dag ongeldig”. De tekst knippert twee keer, waarna het scherm terug naar het klokmenu. Waargenomen zoals beschreven	✓
02	Schakel de datalogger in en ga naar het klokmenu. Stel de maand in op “13”, druk eenmaal op de “drukknop OK”. Op het scherm verschijnt “Maand ongeldig”. De tekst knippert twee keer, waarna het scherm terug naar het klokmenu. Waargenomen zoals beschreven	✓
03	Schakel de datalogger in en ga naar het klokmenu. Stel het uur in op “25”, druk eenmaal op de “drukknop OK”. Op het scherm verschijnt “Uur ongeldig”. De tekst knippert twee keer, waarna het scherm terug naar het klokmenu. Waargenomen zoals beschreven	✓
04	Schakel de datalogger in en ga naar het klokmenu. Stel de minuten in op “61”, druk eenmaal op de “drukknop OK”. Op het scherm verschijnt “Minuten ongeldig”. De tekst knippert twee keer, waarna het scherm terug naar het klokmenu. Waargenomen zoals beschreven	✓

Conclusie: ✓

Test geslaagd.

Bijlage H Handleiding LogiTrack MR1

GEBRUIKSAANWIJZING Datalogger voor validatie van verkeersregelinstallaties LogiTrack MR1



1.	Inleiding	3
2.	Leveromvang	3
3.	Apparaat beschrijving	3
4.	Veiligheidsinstructies.....	4
5.	In 7 stappen starten met meten.....	4
7.	Na de meting	7
8.	Storingen en foutoplossingen	8

1. Inleiding

De datalogger met SD-kaart en afstandssensor slaat tijdstempels op na detectie van reflecterende objecten en slaat de waarden op, op een SD-geheugenkaart. De datalogger heeft een variabel geheugen, afhankelijk van de gebruikte SD-kaart (1 tot maximaal 32 GB). Het aantal detecties wordt direct na het beëindigen van de meting op het OLED-display getoond en tegelijkertijd in de SD-kaart opgeslagen, als een CSV-bestand. De datalogger kan maximaal 10 detecties per rit opslaan, iedere extra detectie wordt genegeerd.

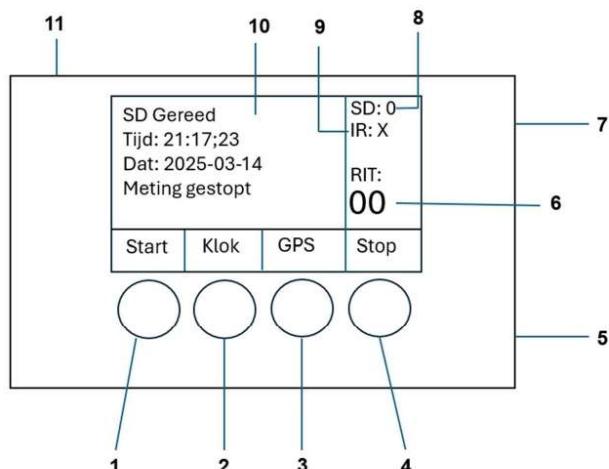
Een interne klok registreert het tijdstip waarop de meting start. Deze interne klok kan in het menu worden gewijzigd naar de huidige lokale tijd. Iedere meting wordt in 1 kHz vastgelegd.

De behuizing van deze datalogger is stevig en robuust, maar niet spatwaterdicht. De meegeleverde afstandssensor is geschikt voor gebruik in buitenomgeving en is spatwaterdicht.

2. Leveromvang

1x datalogger LogiTrack MR1, 1x kabel afstandssensor, 1x afstandssensor Sick WL250, 1x zuignap Manfrotto, 1x statief manfrotto, 1x handleiding.

3. Apparaat beschrijving



Menu	Drukknop 1	Drukknop 2	Drukknop 3	Drukknop 4
Hoofdmenu	Start meting	Klokmenu	GPS-menu	Stop meting
Klokmenu	Waarde + 1	Volgend veld	Instellen en keer terug	Waarde - 1
GPS-menu	-	-	-	-

5. Afstandssensor aansluiting

6. Ritten teller

7. SD-kaartsleuf

8. SD-kaart indicator (0 = aanwezig X= afwezig)

10. OLED-scherm met informatie

11. 12V voeding

9. IR-indicator (0 = detectie X= geen detectie)

4. Veiligheidsinstructies

- Gebruik uitsluitend een gestabiliseerde 12V voeding, indien geen gebruik wordt gemaakt van 12V aansluiting van een voertuig;
- Koppel de afstandssensor correct aan voordat het systeem wordt ingeschakeld;
- Bevestig de zuignap van de afstandssensor op een vlak, droog en schoon oppervlak;
- Sluit het raam helemaal wanneer de kabel van de afstandssensor zich tussen de sluitnaad bevindt, de kabel is gevoelig voor breuken;
- Haal de SD-kaart nooit uit de datalogger tijdens een actieve meting;
- Bescherm de datalogger tegen water en stof. De behuizing is onvoldoende waterdicht.

5. In 7 stappen starten met meten

1. Sluit de afstandssensor en de 12V voedingskabel aan op de datalogger



2. Plaats een geformatteerde (FAT-32) SD-kaart;



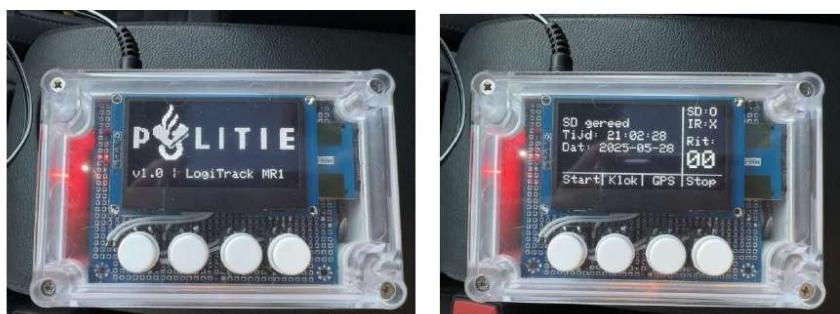
3. Verwijder de afdekhoes van de zuignap en bepaal de locatie van de zuignap van de afstandssensor aan de zijkant van het voertuig;
Zorg dat het gekozen oppervlak nagenoeg vlak is en maak het droog en stofvrij;
Houd de zuignap stevig tegen het oppervlak en pomp net zolang totdat de rode lijn niet meer zichtbaar is.



4. Sluit de 12V stekker aan op de voeding van een voertuig / of 12V batterij;
Controleer of de afstandssensor aan is, de groene LED aan de bovenzijde van de sensor moet branden.



5. Nadat het politie logo op het OLED-scherm is verdwenen, verschijnt het hoofdmenu;



6. Indien nodig kan de lokale tijd worden ingesteld in het klok-menu;



6. Door het bedienen van de startknop (drukknop 1) wordt de meting gestart. De RIT-indicatie op het OLED-scherm wordt met één opgehoogd. De tijdstempels van de detecties worden opgeslagen in een CSV-bestand op de SD-kaart. Na het passeren van de laatste reflectorpaal kan de meting worden beëindigd met de stopknop. Iedere RIT hierna wordt opgeslagen in hetzelfde CSV-bestand, onder elkaar.



7. Na de meting

1. Verwijderen van SD-kaart
Schakel de voeding uit door de 12V-voedingsstekker los te koppelen;
Verwijder hierna pas de SD-kaart uit de sleuf.
2. Uitlezen van de data
Elke rit wordt opgeslagen in hetzelfde CSV-bestand met bestandsnaam:
Metingen.csv;
De CSV-bestanden zijn te openen met Microsoft Excel, Libre office of een andere teksteditor;
Elke regel bevat een tijdstempel van een detectie in milliseconden (1 kHz).

	A	B	C	D
1	Rit 1, 2025-04-21, 15:46:28			
2	15:46:30.741			
3	15:46:31.044			
4	15:46:31.367			
5	15:46:31.419			
6	15:46:31.797			
7	15:46:32.082			
8	15:46:32.130			
9	15:46:32.175			
10	15:46:32.220			
11	15:46:32.264			
12	15:46:32.309			
13	15:46:32.355			
14	15:46:32.503			
15	15:46:32.653			
16	15:46:32.775			
17				
18	Rit 1, 2025-04-28, 15:49:59			
19	15:50:00.538			
20	15:50:01.917			

3. Verwijderen van de afstandssensor (en zuignap)
Trek voorzichtig de zuignap los van het voertuig, door het zwarte lipje omhoog te trekken;
Reinig eventueel het oppervlak;
Koppel de sensor los van de datalogger.



8. Storingen en foutoplossingen

Onderwerp	Aard storing/probleem	Mogelijke oplossing
SD-kaart	SD-kaart bevind zich in de sleuf. Melding "Plaats SD-kaart" verdwijnt niet.	Controleer of de SD-kaart maximaal 32 GB opslag heeft; Controleer of de SD-kaart op de juiste wijze is geplaatst; Formateer SD-kaart.
Afstandssensor	Afstandssensor detecteert niet.	Controleer de LED aan de bovenzijde van de afstandssensor; Controleer of de kabel is aangesloten op de datalogger en de afstandssensor; Controleer of de datalogger 12 V voeding krijgt.
Datalogger	Scherm blijft zwart.	Controleer of de rode LED achter het OLED-scherm brand; Controleer of er 12 V voeding is; Controleer of de voedingskabel met de datalogger is verbonden.