



TECHNISCH ONTWERP

Verkeersmonitor

SAMENVATTING

Dit document bevat het technisch ontwerp van het verkeersmonitor project

Sem Overbeek
(s1212918)

ESE Leertaak 1

1 Versiebeheer

Versie	Naam	Wijziging	Datum
0.1	Niels Tielenburg	eerste conceptversie	17-04-25
1.0	Willie Conen	Eerste release versie	15-05-25
1.1	Niels Tielenburg	AIAS toegevoegd	08-09-25
2.0	Sem Overbeek	TO bijgewerkt met informatie over snelheidsmeting	19-09-25

2 Inhoudsopgave

1	Versiebeheer	1
2	Inhoudsopgave	2
3	Inleiding	3
4	Context	4
5	Containers.....	4
6	Componenten	4
7	Controller	5
7.1	Main.....	6
8	Teller	7
8.1	De axle detected functie	7
8.2	Vehicle passed functie.....	7
8.3	De unitPassedButton functie	8
9	Aantal indicatie.....	9
10	Snelheid indicatie.....	9
11	References	9

3 Inleiding

De opdrachtgever heeft een probleem met verkeerssituaties en wil een apparaat ontwikkeld hebben om inzicht te krijgen in verkeersdrukke en snelheid van voertuigen. Om aan te geven hoe het systeem ontwikkeld gaat worden is dit ontwerp opgesteld. Het heeft als doel om voor ontwikkelaars inzicht te geven hoe het systeem technisch ontwikkeld moet worden. Ook de opdrachtgever heeft belang bij dit document, zodat toekomstige ontwikkelaars eenvoudig uitbreidingen aan de code kunnen toevoegen, en makkelijker bugs kunnen vinden en oplossen.

In (ESA team, 2025) staan het functioneel ontwerp van de verkeersmonitor beschreven. De oplossing voor de gevraagde functionaliteiten die daar beschreven staan zijn in dit document technisch uitgewerkt. Verwijzing naar de betreffende use case of requirement is aangegeven met US.1, RV.1, etc. Het technisch ontwerp wordt aan de hand van het C4 model uitgediept. Per laag wordt in een apart hoofdstuk beschreven hoe het systeem eruit moet komen te zien.

4 Context

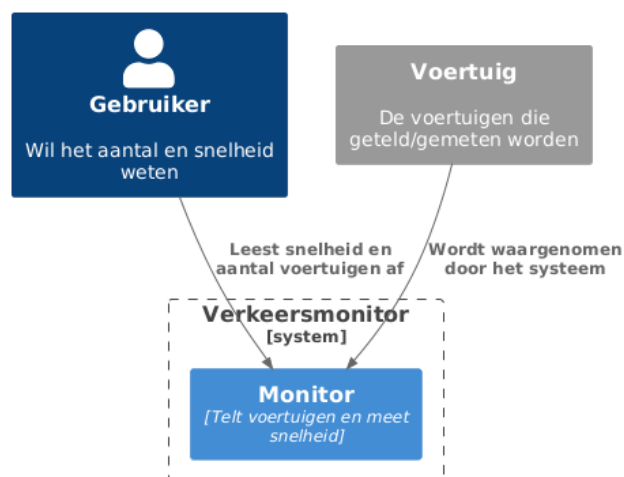
Het systeem dat ontwikkeld moet worden (US1, US2, US3 en US4), heeft interactie met een gebruiker. Deze gebruiker leest de gemeten snelheid af en kan ook aflezen hoeveel voertuigen er geteld zijn. Daarnaast heeft het systeem interactie met de voertuigen die geteld worden en waarvan de snelheid wordt gemeten.

Figuur 1 context verkeersmonitor



5 Containers

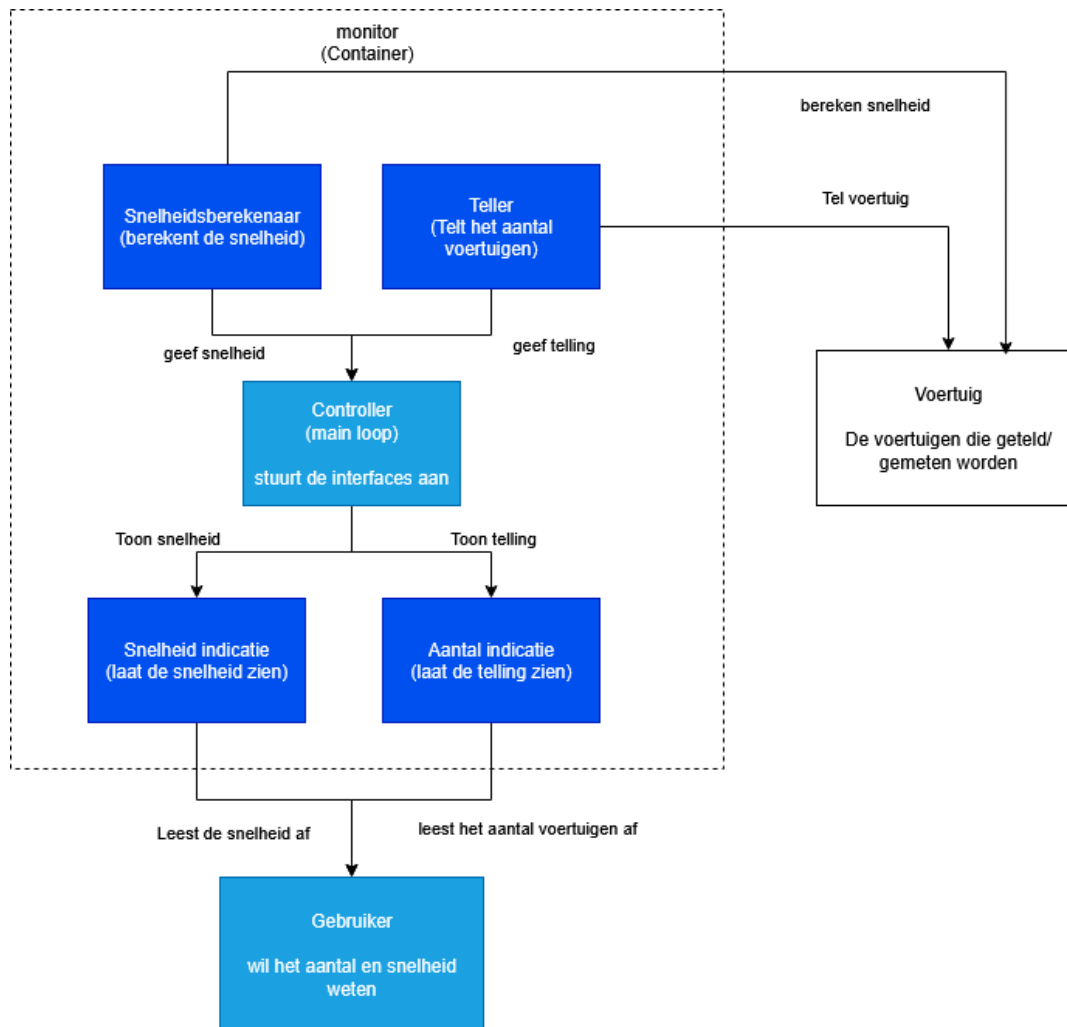
Het systeem heeft slechts 1 container. Er is geen verdeling tussen meerdere containers. Het diagram is daardoor vergelijkbaar met de context.



Figuur 2 Containers verkeersmonitor

6 Componenten

De monitor bestaat uit 5 verschillende componenten. Voor de interactie met het voertuig zijn 2 componenten nodig: de Teller voor het tellen van het aantal voertuigen (US1). Deze hebben onderling geen interactie met elkaar. Voor de gebruiker is er ook een los component: om de telling van de voertuigen weer te geven (US3).



Figuur 3 Components verkeersmonitor

7 Controller

Voor de controller is gekozen om de Arduino Uno te gebruiken met daarop de ATmega328P microcontroller (RV.1). Deze is gekozen vanwege de formfactor waarbij er eventueel makkelijk shields gemaakt kunnen worden die eenvoudig te monteren zijn. Daarnaast is dit een eenvoudige microcontroller, maar is het toch voldoende voor de doeleinden waarvoor het gebruikt gaat worden.



Figuur 4 Arduino UNO

7.1 Main

In de `main()` functie wordt eerst de initialisatie gedaan. De `init()` functie wordt aangeroepen om de timers in te stellen zodat de standaard functie `millis()` werkt. Daarna worden de gebruikte input pins en output pins ingesteld in de functie `InitializeIO()`.

Daarna wordt de functie `unitPassedButton(button)` aangeroepen voor elke knop om te bepalen of een voertuig de detectielus heeft gepasseerd. Bij het passeren van `button0` wordt `unitsInProcess` met één verhoogt. Dit betekent dat er een voertuig bijkomt die langs `button0` is gekomen maar nog niet langs `button1`. Voor nu kan `unitsInProcess` maximaal één zijn, voor in de toekomst kan dit makkelijk aangepast worden.

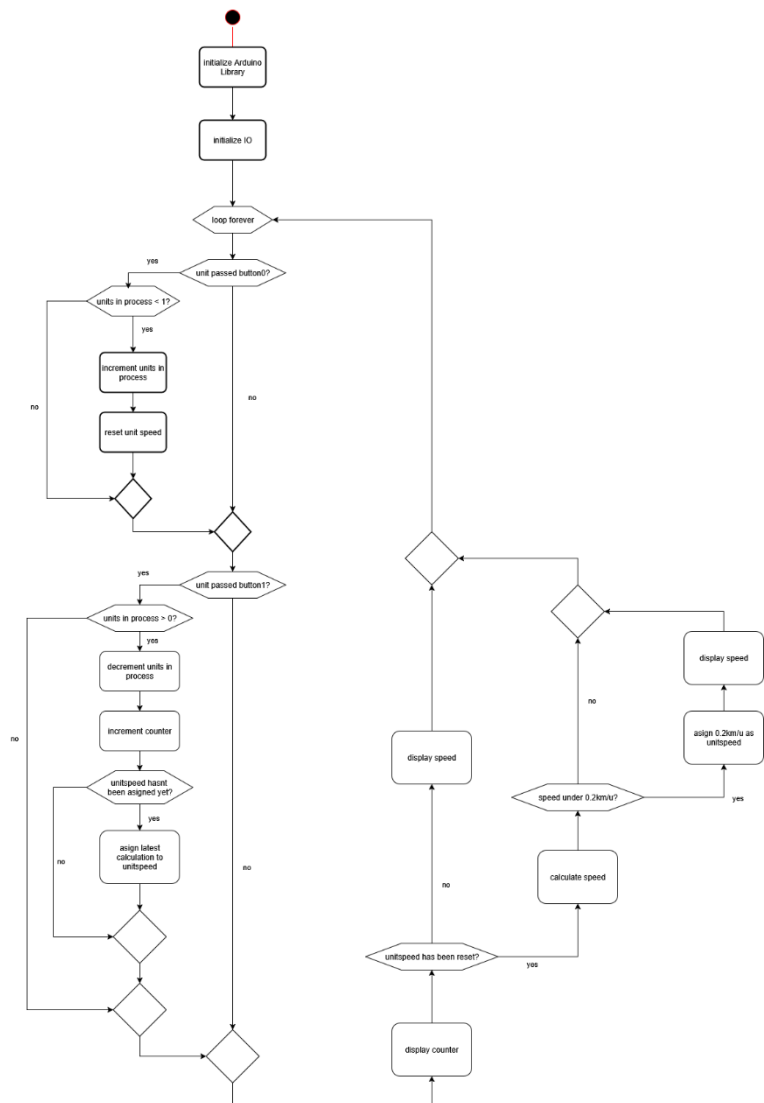
Bij passage van `Button1` wordt `unitsInProcess` (wanneer mogelijk) met één verlaagd en uiteindelijk de counter met één verhoogt

Elke doorgang wordt de counter weergegeven via de functie 'display counter(counter)'.
 Elke doorgang wordt de counter weergegeven via de functie 'display counter(counter)'.

Na het paseren van de eerste knop wordt ook `unitSpeed` gereset. Hierdoor wordt elke doorgang de snelheid berekent met `calculateSpeed(starttime, endtime)`. Wanneer de tweede knop ingedrukt is zal de laaste berekening toegewezen worden aan `unitSpeed` en stopt de berekening. De berekening kan ook eerder stoppen wanneer de snelheid onder 0.2km/u komt. Dit zal dan de snelheid zijn die wordt toegewezen aan `unitSpeed`

Zodra `unitSpeed` een waarde heeft zal elke doorgang de functie `displaySpeed(speed)` aangeroepen worden die de snelheid uitleest en weergeeft op het 4 digits 7 segmenten display.

De activity diagram van de functie main is te zien in Figuur 5.



Figuur 5 Activity diagram main

8 Teller

Voor Use Case US.1 worden er standaard knoppen gebruikt voor het simuleren van de luchtslangentelslangen (RV.2). Een voorbeeld van een standaard knop is te zien in Figuur 6. Wanneer het systeem in gebruik genomen wordt, moeten de echte sensoren worden aangesloten. Dit valt buiten de scope van dit project, maar om toch het systeem te kunnen testen, worden er knoppen gebruikt. Er is gekozen om interne pull-up weerstand te gebruiken, omdat dat uiteindelijk weer een componentje scheelt in de uiteindelijke opstelling.



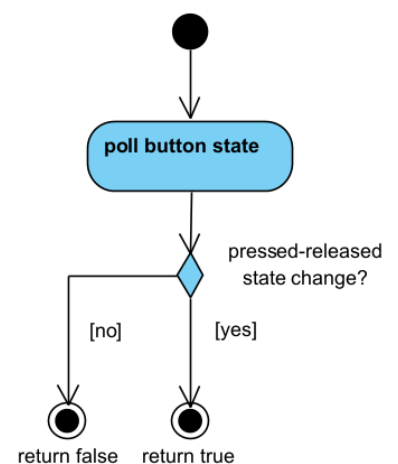
Figuur 6 Druknop

8.1 De axle detected functie

Het figuur hiernaast geeft het activity diagram van de functie **axle detected** .

Via 'poll button state' wordt de status van de drukknop gemeten en alleen als er een verandering is van *pressed* naar *released* wordt true teruggegeven, anders false.

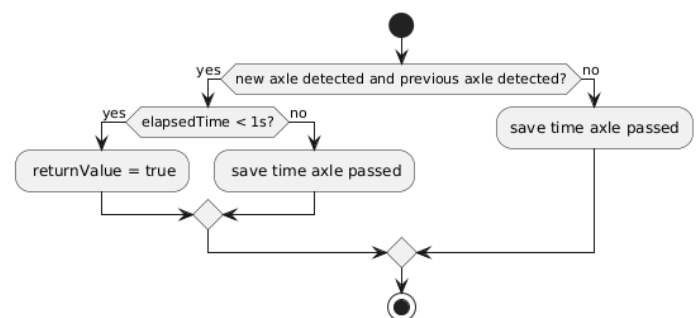
De huidige state moet worden bewaard om de statusverandering te kunnen meten.



Figuur 7 Activity diagram axle detected

8.2 Vehicle passed functie

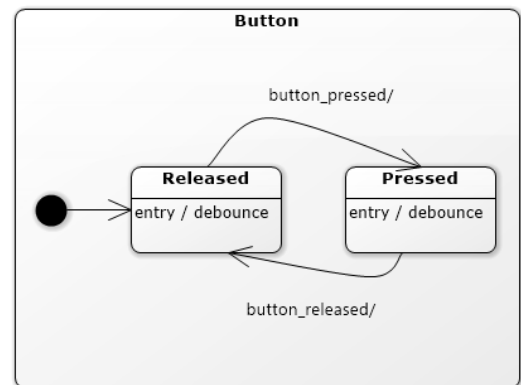
Het diagram hiernaast toont wanneer er een voertuig gedetecteerd wordt. Als er een as wordt gedetecteerd moet de huidige tijd worden opgeslagen. Als er een tweede as wordt gedetecteerd moet er worden gecontroleerd of de tijd tussen de twee assen minder is dan 1 seconde. Als dat het geval is, is er een voertuig gedetecteerd. Als dat niet het geval is, zit er dus meer dan 1 seconde tussen 2 metingen, en gaan we uit van een verkeerde meting.



Figuur 8 Activity diagram vehicle passed

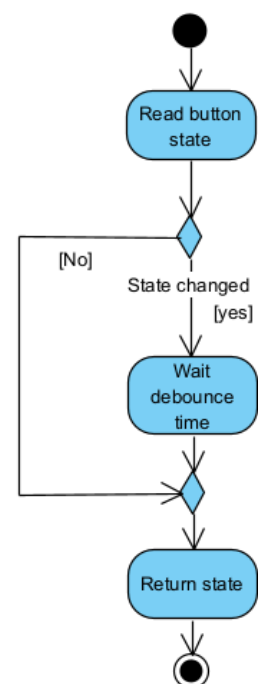
8.3 De unitPassedButton functie

Het diagram hiernaast toont de state verandering bij het indrukken van de schakelaar. Zowel bij indrukken als loslaten moet gedebounced worden. Deze state machine is onderdeel van de functie **unitPassedButton(button)**.



Figuur 9 State diagram button

Het volgende figuur geeft het activity diagram van de **unitPassedButton()** functie met debounce. Om te bepalen of een state change heeft plaatsgevonden is de previous state nodig. Alleen bij een state change moet worden gewacht (busy wait).



Figuur 10 Activity diagram button state

Pin nr	Pin naam	I/O	Verbonden met	Bijzonderheden
D2	PD2	Input	Eerste knop voor tellen en meten snelheid	Interne pull-up
D3	PD3	Input	Tweede knop voor tellen en meten snelheid	Interne pull-up

Tabel 1 Aansluittabel Telsensor

9 Aantal indicatie

Use Case US.3 geeft aan dat er een indicatie moet komen voor het tonen van het aantal getelde voertuigen. Voor het tonen van het aantal voertuigen zullen 4 LEDs gebruikt gaan worden (NFE.7). Deze geven binair het aantal voertuigen weer. Daar is voor gekozen en niet voor een 7-segmenstdisplay, omdat er te weinig pinnen zijn om naast het display voor de snelheid ook nog een 7-segmentsdiplay te hebben in combinatie met ook de sensoren en debugpoort.

De LEDs zijn op de volgende pinnen aangesloten:

Pin nr	Pin naam	I/O	Verbonden met	Bijzonderheden
A0	PC0	Output	LED1 Verkeersteller	Aansluiten met weerstand in serie
A1	PC1	Output	LED2 Verkeersteller	Aansluiten met weerstand in serie
A2	PC2	Output	LED3 Verkeersteller	Aansluiten met weerstand in serie
A3	PC3	Output	LED4 Verkeersteller	Aansluiten met weerstand in serie

Tabel 2 Aansluittabel LEDs

10 Snelheid indicatie

Use Case US.5 geeft aan dat er een indicatie moet komen voor het tonen van de snelheid van voertuigen. Voor het tonen van de snelheid gebruiken wij een 4x7segmenten display. Één digit zal altijd uitstaan, wij gebruiken er dus maar 3. Er zal van multiplexing gebruik gemaakt worden om deze te laten werken met de beperkte pinnen die wij beschikbaar hebben.

Het 4x7 segmentendisplay is op de volgende pinnen aangesloten

Pin nr	Pin naam	I/O	Verbonden met	Bijzonderheden
D9	PB1	Output	4x7segmentendisplay	Switchpin (rechter digit)
D10	PB2	Output	4x7segmentendisplay	Switchpin (midden digit)
D11	PB3	Output	4x7segmentendisplay	Switchpin (linker digit)
D12	BP4	Output	4x7segmententdispay	Datapin (midden segment)
D4	PD4	Output	4x7segmententdispay	Datapin (middenonder segment)
D5	PD5	Output	4x7segmententdispay	Datapin (rechtsboven segment)
D6	PD6	Output	4x7segmententdispay	Datapin (rechtsonder segment)
D7	PD7	Output	4x7segmententdispay	Datapin (linksboven segment)
D8	PB0	Output	4x7segmententdispay	Datapin (linksonder segment)
D13	PB5	Output	4x7segmententdispay	Datapin (middenboven segment)
A4	PC4	Output	4x7segmententdispay	Datapin (decimaal segment)

11 References

ESA team. (2025). *Functioneel Ontwerp Verkeersmonitor*. Zwolle: HBO-ICT / Windesheim.