

*EMBEDDED PROGRAM DEVELOPMENT
(EPD)*

**EINDOPDRACHT:
SPOORWEGOVERGANG**

INHOUDSOPGAVE

1. Introductie.....	3
2. Situatieschets	3
3. Beschrijving van het op te leveren prototype.....	4
1.1 <i>Extra verplichte zaken voor de “eerste mogelijkheid”</i>	6
1.2 <i>Extra verplichte zaken voor “herkansers”</i>	6
1.3 <i>Optionele zaken.....</i>	6
4. Op te leveren producten	7
1.4 <i>Functioneel ontwerp</i>	7
1.5 <i>Technisch ontwerp.....</i>	8
1.6 <i>Programmacode</i>	9
1.6.1 <i>Suggesties om programmeren eenvoudiger/prettiger te maken</i>	9
1.7 <i>Ingevuld self-assessmentformulier.....</i>	9
5. Beoordeling.....	9
1.8 <i>Knock-out-criteria</i>	10

1. INTRODUCTIE

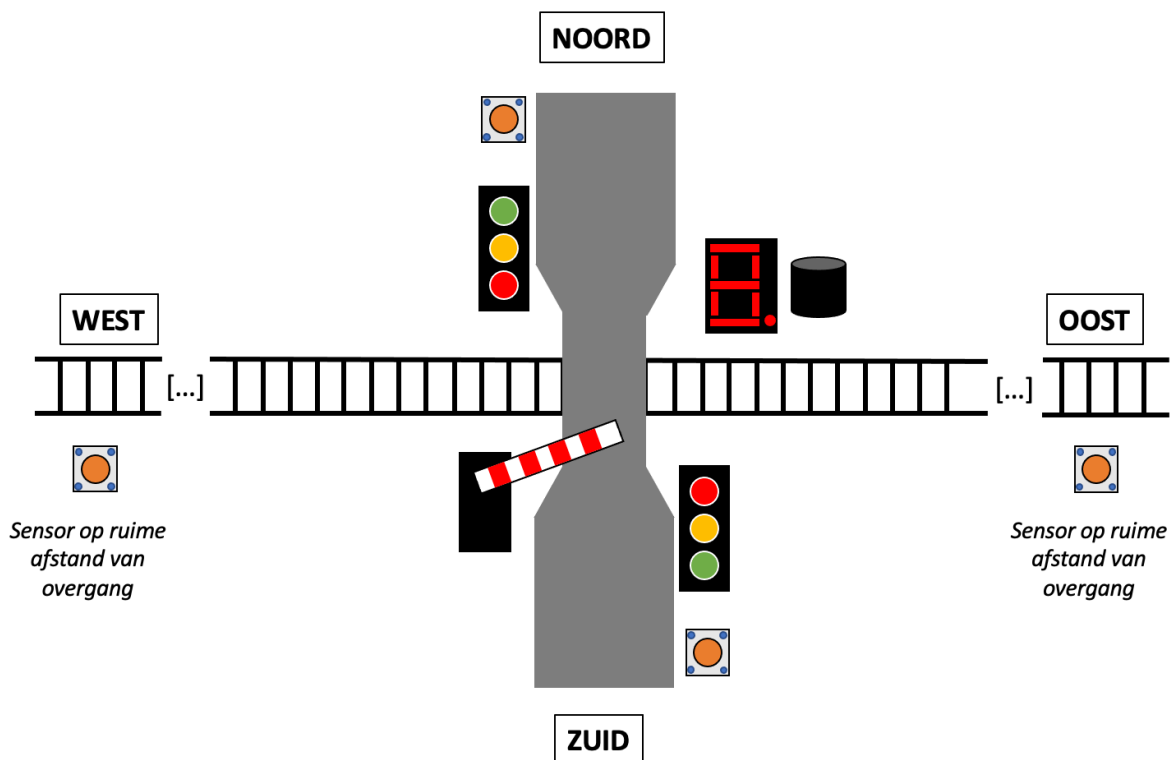
De Betuweroute is een 160 kilometer lange goederenspoorlijn van de Maasvlakte in Rotterdam naar de Duitse grens, enkele kilometers voorbij Zevenaar. Het beheer van de spoorlijn is, net zoals bij de rest van het Nederlandse spoorweganet, in handen van ProRail.

Zoals je zult begrijpen, heeft de aanleg van een spoorlijn nogal wat impact op de leefomgeving van de bewoners van het gebied dat door de lijn doorkruist wordt. Om de impact te verminderen, wordt onderzoek gedaan naar het toevoegen van kleine spoorwegovergangen over de spoorlijn¹. Om de mogelijkheden goed in kaart te brengen, wil ProRail graag een prototype laten ontwikkelen. Aan jou de opdracht dit prototype te maken.

2. SITUATIESCHETS

In het prototype dat je gaat maken, worden enkele zaken uit de werkelijkheid vereenvoudigd. Zo is er slechts 1 slagboom, 1 display en 1 buzzer.

Hieronder staat een schematische weergave van de situatie.



¹ Disclaimer: dit voorbeeld is fictief: er zijn bij ons geen plannen bekend om spoorwegoverwegen toe te voegen aan de Betuweroute.

3. BESCHRIJVING VAN HET OP TE LEVEREN PROTOTYPE

Hieronder vind je een puntsgewijze beschrijving van de functionaliteiten van het prototype. Het is aan jou om hiervan een functioneel ontwerp (FO) op te stellen waarin deze tekst is omgezet naar een lijst requirements. Je maakt ook een technisch ontwerp (TO), een realisatie en je test of alle requirements behaald zijn. Wat er precies in het FO en TO moet worden opgenomen vind je verderop in deze opdrachtschrijving.

- Als er een trein voorbijkomt (van west naar oost of andersom), wachten de voetgangers en/of bestuurders (vanuit noord en/of zuid) tot deze is gepasseerd. Een verkeerslicht geeft aan of ze mogen rijden/lopen.
 - Omdat de weg op de overgang smal is, kan de spoorwegovergang slechts door één richting tegelijk (vanuit noord of vanuit zuid) gebruikt worden, met verkeerslichten (rood, geel, groen) wordt aangegeven wie mag rijden/lopen.
 - Om ongelukken te voorkomen wordt er bij de spoorwegovergang gebruik gemaakt van een poort om het onmogelijk te maken voor bestuurders om alsnog te passeren als er al een trein in aankomst is. Deze spoorboom wordt gesimuleerd door een servo-motor (in het prototype wordt slechts 1 spoorboom gemaakt).
- Als iemand de spoorweg wil oversteken, geeft hij of zij dat aan door een druk op de knop.
 - Als beide lichten (bij noord en zuid) rood zijn, krijgt de eerste persoon die gedrukt heeft ook het eerst groen. Een druk op de knop moet *altijd* onthouden worden tot de bijbehorende actie verwerkt is (als bijv. noord op de knop drukt terwijl zuid groen heeft, moet noord dus op groen gaan zodra dat mogelijk is)
 - Enkele seconden nadat het licht groen is geworden, gaat het op geel en daarna na enige seconden op rood.
 - Voordat het andere verkeerslicht naar groen gaat, blijven **beide** lichten even allebei op rood. Dit om ervoor te zorgen dat de spoorwegovergang echt leeg is deze gebruikt kan worden door een andere richting. Men noemt dit vaak de ontruimingstijd.
 - Als er een trein aankomt (zie hieronder) gaat een eventueel groen licht **onmiddellijk** op geel en vervolgens op rood. Als het licht al op geel staat, gaat het op de normale manier naar rood.
 - Als er een trein aankomt en er wordt vanuit noord of zuid op groen gewacht, krijgt de trein **altijd** voorrang (licht gaat dus pas op groen zodra de trein gepasseerd is). Uiteraard wordt wel onthouden dat de knop ingedrukt is geweest, zodat die richting na het passeren van de trein alsnog groen kan krijgen.
- In het spoor zitten aan twee kanten van de overgang (oost en west) sensoren, gesimuleerd door drukknoppen.
 - Als de ene sensor (west of oost) een trein signaleert, gaat het verkeerslicht dat groen is eerst naar geel (zoals hierboven ook al is aangegeven). Na enige tijd gaat het rode licht van beide verkeerslichten

branden en knippert *tegelijk* het gele licht. Dan sluit de slagboom langzaam en klinkt er een snel tikkend geluid uit de buzzer.

- Zodra de slagboom gesloten is, blijven de rode lichten aan en de gele lichten knipperen, maar stopt de buzzer met tikken.
- Zodra de andere sensor (west in het geval dat de trein via oost aankwam, en oost in het geval dat de trein via west aankwam) een trein signaleert, opent de slagboom weer (de buzzer gaat weer tikken, de rode lichten blijven aan, de gele lichten blijven knipperen).
- Zodra de slagboom volledig geopend is, telt het display van 5 naar 0 en klinkt uit de buzzer een signaal bestaande uit steeds 3 tikken en dan even pauze. Zodra het display op 0 staat stoppen de gele lichten met knipperen en de buzzer gaat op stil. Vanaf dat moment blijven de lichten op rood tot er weer verkeersdeelnemers uit noord of zuid willen oversteken (als er al op de knoppen van noord en/of zuid was gedrukt, worden deze afgehandeld: drukken op de knop wordt dus altijd onthouden door het systeem).
- Als het systeem zich in de ‘ontruimingstijd’ bevindt, krijgt de trein voorrang op een eventuele bestuurder/voetganger uit richtingen noord of zuid.

Enkele aanvullende opmerkingen:

- Overal waar iets wordt gezegd over het indrukken van een knop, wordt bedoeld dat de knop kort wordt ingedrukt (dus niet heel lang ingehouden). Als de druk op de knop niet onmiddellijk kan worden afgehandeld, moet deze onthouden worden. Er is één uitzondering: als de afhandeling van een knop bezig is, en tijdens die afhandeling wordt nogmaals op de knop gedrukt, dan mag dit genegeerd worden.
- Je mag ervan uitgaan dat het passeren van een trein altijd volledig wordt afgehandeld voordat er een volgende trein komt, dus:
 - Je mag ervan uitgaan dat er nooit twee treinen uit verschillende richtingen komen.
 - Je mag ervan uitgaan dat er nooit twee treinen onmiddellijk na elkaar komen (er is altijd voldoende tijd om de slagboom volledig te openen en een van de richtingen noord/zuid tenminste enige tijd groen licht te geven).
- Je mag ervan uitgaan dat een trein de tweede sensor pas activeert als de slagboom volledig gesloten is (je hoeft tijdens het sluiten van de slagboom dus niet te controleren of de andere sensor geactiveerd wordt, maar dit *mag* wel).
- Mocht om de een of andere reden tweemaal (of vaker) achter elkaar dezelfde treinsensor (oost/west) geactiveerd worden zonder dat de andere treinsensor wordt geactiveerd, dan hoef je alleen de eerste activatie te verwerken.
- Bij het opstarten van het systeem zijn beide verkeerslichten op rood en is de slagboom geopend
- Bij het starten van het systeem mag je de slagboom openen met een directe `Servo.write()`-actie, maar verder beweegt de servo overal traag (en zonder `delay()`).

- Wanneer een trein de tweede sensor activeert, is hij de overgang altijd voorbij, het is dan dus veilig om de slagboom te gaan openen.
- De buzzer maakt alleen geluid in de genoemde situaties.

1.1 EXTRA VERPLICHTE ZAKEN VOOR DE “EERSTE MOGELIJKHEID”

Doe je voor het eerst aan dit vak mee (en ben je dus op geen enkele wijze “herkanser”), dan is het volgende ook verplicht (voor herkansers dus niet):

- Om in te kunnen schatten wanneer onderhoud aan de slagboom nodig is, moet worden bijgehouden hoe vaak deze geopend is geweest. Steeds als de teller verhoogd wordt, wordt de waarde geprint via de seriële poort.
- Wanneer de letter “r” binnenkomt via de seriële poort, wordt de waarde van de teller op 0 gezet. Dit gebeurt ook bij het opnieuw opstarten van het systeem (je hoeft de waarde dus niet in het EEPROM te zetten).

1.2 EXTRA VERPLICHTE ZAKEN VOOR “HERKANSERS”

Doe je als “herkanser” mee aan deze toets, dan is het volgende ook verplicht (dit geldt OOK als je bij de eerste kans niets hebt ingeleverd of niet hebt deelgenomen). Twijfel je of dit voor jou geldt? Neem dan contact op met de coördinator van EPD of met je (toegewezen) docent.

- Als er tweemaal op een knop bij noord of zuid wordt gedrukt, gaat het om een overstekende voetganger. Het licht blijft dan langer op groen dan normaal gesproken (ook hier geldt dat als er een trein aankomt, het licht toch meteen op geel gaat).
- Met een potmeter wordt de snelheid van een aankomende trein aangegeven. Hoe lager de op de potmeter ingestelde waarde, hoe meer tijd er zit tussen het signaleren van een trein (door de sensor bij oost of west) en het eventueel naar geel gaan van een groen verkeerslicht. Hiermee wordt voorkomen dat bestuurders/voetgangers lang voor niets staan te wachten.

1.3 OPTIONELE ZAKEN

De opdrachtgever ziet naast de bovenstaande verplichte zaken ook graag de onderstaande zaken terug. Deze zijn optioneel (dus **NIET verplicht**). De hierboven genoemde verplichte zaken die voor jou niet gelden, mag je ook als optioneel uitvoeren.

- Wanneer het slecht weer is (gesimuleerd door een lage waarde van de thermometer en/of een lage gemeten lichtsterkte), moet de “geel”-stand van de lichten wat langer duren, en openen en sluiten de slagbomen net wat sneller.
- Er is een testfunctionaliteit aanwezig: zodra er op een testknop wordt gedrukt of een bepaald teken op de seriële poort binnenkomt, gaan alle lichten een voor een aan en vervolgens weer uit, opent en sluit de slagboom langzaam en geeft de buzzer een pieptoon. Daarna gaat het systeem weer naar normale werking.
- Het aantal keer dat de slagboom geopend is (zie hierboven) wordt bijgehouden in het EEPROM. Elke keer dat de slagboom geopend is, moet deze waarde met

1 verhoogd worden². Je hoeft niet verder te kunnen tellen dan 255 (het hoogste getal dat je in 1 byte kunt opslaan) en je hoeft ook op geen enkele manier rekening mee te houden met de situatie dat de waarde groter wordt dan 255. De waarde wordt op 0 gezet zodra de letter 'r' via de seriële poort wordt gelezen.

- De slagboomteller kan ook grotere getallen dan 255 opslaan.
- Zodra de waarde van de slagboomteller groter is dan een bepaalde instelbare waarde, gaat er een ledje branden zodat een voorbijkomende monteur kan zien dat onderhoud nodig is.
- Er is in de richtingen oost en west een led die knippert op het moment dat de slagboom geopend is. Een machinist kan dan zijn snelheid aanpassen omdat er mogelijk verkeer op de overgang is.
- Wanneer een stoplicht op geel gaat, wordt er afgeteld op het display.
- Wanneer tijdens het passeren van een trein door zowel noord en zuid op de knop is gedrukt, wordt de *volgorde* waarin dat gebeurde ook onthouden en wordt het ook in die volgorde afgehandeld.

4. OP TE LEVEREN PRODUCTEN

Alle onderstaande producten moeten worden ingeleverd en worden beoordeeld. **Als een of meerdere van deze zaken ontbreken, zal je werk niet beoordeeld worden** (let op: ook in dit geval raak je een van de toetsgelegenheden die je elk studiejaar hebt kwijt):

- Functioneel ontwerp
- Technisch ontwerp
- Programmacode
- Ingevuld self-assessmentformulier

De producten staan hieronder nader beschreven.

1.4 FUNCTIONEEL ONTWERP

Maak een document dat een tabel met specificaties bevat. Vraag de opdrachtgever (docent) bij onvolledige of onduidelijke specificatie. Let daarbij op de volgende zaken:

- Maak specificaties met korte zinnen waarbij je steeds slechts één specificatie benoemt: dus geen lange proza waarin verschillende specificaties verstopt zitten.
- Zorg dat de specificaties SMART zijn, gebruik dus geen woorden als “kort” of “snel”, maar benoem (milli)seconden, Hertz en eventueel graden Celsius en Lux.

² Wees voorzichtig met het beschrijven van het EEPROM, want dit kan slechts een beperkt aantal keer. Zorg dus dat deze teller maar 1 keer verhoogd wordt bij het openen van de boom. Test dit eerst door bijvoorbeeld een `Serial.println` uit te voeren in plaats van meteen het EEPROM-geheugen te beschrijven. Ben je er zeker van dat dit precies eenmaal gebeurt bij het openen van de slagboom, dan kun je echt naar het EEPROM gaan schrijven.

- Zorg dat je specificaties zo volledig en specifiek zijn dat je het volledige te bouwen systeem ermee beschrijft.
- Wanneer een bepaalde specificatie samenhangt met een andere (wanneer die bijvoorbeeld in werking treedt nadat een andere specificatie is afgerond), geef dat dan duidelijk aan (bijvoorbeeld door je specificaties te nummeren en te verwijzen naar die nummers).
- Geef per specificatie de prioritering aan (MoSCoW). Alle vereisten geef je aan als M of S (kies zelf een goede prioritering: kernzaken krijgen een M en iets minder belangrijke (maar wel vereiste) zaken een S). Geef optionele zaken de prioriteit C (als je van plan bent ze te realiseren) of W (als je niet van plan bent ze te realiseren).
- Voeg per specificatie kolom toe met de naam “Testresultaat” (die ga je helemaal aan het einde gebruiken).

Een stukje van de tabel zou er als volgt uit kunnen zien (waarbij de kolom “Testresultaat” pas gevuld wordt nadat je helemaal klaar bent met ontwerpen en bouwen). Uiteraard zijn de hier genoemde voorbeeldspecificaties niet passend bij deze opdracht.

Nr	Omschrijving	Prioriteit	Testresultaat
1	De blauwe lamp knippert met 2 Hertz.	M	Geslaagd
2	Zodra de gebruiker via de seriële poort een ‘M’ verstuurt, knippert de blauwe lamp 10 seconden met 10 Hertz.	M	Geslaagd
3	Er wordt een slagboom gesimuleerd met een servo-motor.	M	Geslaagd
4	Zodra de lamp uit req.nr. 2 is gestopt met knipperen, gaat de slagboom in 2 seconden van volledig geopend naar volledig gesloten.	C	Niet geslaagd
5	De gebruiker kan de roze lamp aan- en uitschakelen door op de knop te drukken.	S	Aanzetten werkt, uitzetten niet

1.5 TECHNISCH ONTWERP

Maak een document waarin je de in het FO beschreven specificaties technisch uitwerkt. Het TO bevat minimaal de volgende onderdelen (voeg de bronbestanden eventueel als bijlage, met name als ze in het document niet goed leesbaar zijn):

- Eén of meer toestandsdiagrammen (geef duidelijk initialisatie, toestanden, de overgangs-voorwaarden en de acties per toestand (do, entry, exit) aan). Geef ook de start-toestand aan.
- Een softwareontwerp met functies georganiseerd in een tabbladendiagram (bekijk eventueel het document “EPD Tabbladendiagrammen” op OnderwijsOnline). Uiteraard zorg je dat de verschillende hardwarecomponenten hun eigen tabblad krijgen en dat er geen overbodige afhankelijkheden tussen de tabbladen zijn.

- Een korte toelichting van de tabbladen en functies (beschrijf hun doel en bij de functies ook eventuele de parameters en eventuele returnwaarden). Heel eenvoudige functies en getters/setters hoeven niet beschreven te worden.
- Een netjes verzorgd aansluitschema (export van Tinkercad).

1.6 PROGRAMMACODE

Realiseer de in het FO beschreven specificaties volgens de in het TO opgestelde ontwerpen. Uiteraard zorg je voor goed leesbare code en gebruik je de in de lessen aangeboden templates (zoals het template voor statemachines). Mocht je ontdekken dat bepaalde zaken in het FO of TO niet voldoen of nog niet volledig zijn, pas die dan aan.

Ga met het functioneel ontwerp na of alle specificaties naar behoren werken en voeg daarover informatie toe in de kolom “Testresultaat” in de tabel met specificaties.

1.6.1 SUGGESTIES OM PROGRAMMEREN EENVOUDIGER/PRETTIGER TE MAKEN

- Zet in elke entry-functie een `Serial.println` die de huidige toestand print, zo kun je beter in de gaten houden of de overgangen goed plaatsvinden.
- Maak sowieso veel gebruik van `Serial.println` om de werking van je programma goed in de gaten te houden.
- Zet tijdens het programmeren de groen- en geeltijden niet te lang, anders moet je steeds lang wachten.
- Ben je een lastig stukje aan het programmeren, overweeg dit dan in een aparte sketch te doen, zodat je het geïsoleerd kunt programmeren en testen.
- Programmeer de functionaliteit toestand voor toestand uit, en ga pas naar de volgende toestand als je helemaal klaar bent. Blijft er toch wat liggen? Geef dan duidelijk in je code aan dat je hier later nog wat mee moet doen (bijv. met een markering als `// TODO`).
- Realiseer je dat sommige zaken buiten de statemachine gehouden kunnen worden omdat ze altijd gedaan moeten worden (denk bijvoorbeeld aan het uitlezen van de knoppen).
- Zoals hierboven is aangegeven, moet het systeem onthouden er op een knop gedrukt is (dat kan bijv. met een boolean), maar zodra een druk op de knop verwerkt is, moet dit onthouden uiteraard eindigen (bijv. door de boolean weer op `false` te zetten).

1.7 INGEVULD SELF-ASSESSMENTFORMULIER

Om je te helpen te bepalen of je werk van voldoende kwaliteit is, moet je een self-assessment uitvoeren en de resultaten vastleggen in het formulier dat je vindt op OnderwijsOnline (bij de opdrachtschrijving).

5. BEOORDELING

Beoordeling zal worden gedaan aan de hand van een assessment waarin je de werking van je uitwerking moet demonstreren en je vragen moet beantwoorden over je uitwerking. De beoordelaar zal je beoordelen aan de hand van het

beoordelingsmodel op iSAS. Hierin zijn ook enkele knock-out-criteria opgenomen. De belangrijkste hiervan worden hieronder herhaald (enkele komen in het model terug als knock-outs rond implementeren van eisen).

1.8 KNOCK-OUT-CRITERIA

Je krijgt onmiddellijk een onvoldoende als je werk een of meerdere van deze criteria voldoet (zie ook het beoordelingsmodel):

- Je hebt gebruik gemaakt van de functie `delay()` of zet in je code op een andere manier het systeem voor enige tijd volledig op pauze (wees vooral voorzichtig met `while`-constructies!).
- Je hebt de statemachine (of statemachines) niet uitgeprogrammeerd volgens het in het lesmateriaal aangereikte implementatie-sjabloon. Het gaat hier dus om de `switch/case`-implementatie met functies voor de `entry-` `do-` en `exit-` acties zoals beschreven in het materiaal van onderwijsweek 5 (zie OnderwijsOnline).
- De ingeleverde documenten zijn niet geschreven conform de eisen van de Controlekaart Documenten die gaan over opbouw.
- De verplichte functionaliteit is onvoldoende geïmplementeerd. Er ontbreekt functionaliteit (of er is functionaliteit onjuist geïmplementeerd) waardoor de opdracht significant eenvoudiger is geworden. Dit kan **niet** hersteld worden door extra functionaliteiten te implementeren. Een voorbeeld van een functionaliteit die bij afwezigheid tot een onvoldoende leidt, is het niet laten onthouden van het drukken op een knop.