

Technisch Ontwerp

Periode 1.3

26-03-2025

wouter.wesseling.wesseling@student.nhlstenden.com

stefan.spitze@student.nhlstenden.com

Pascal.Westerhof@student.nhlstenden.com

thijs.arkes@student.nhlstenden.com

Silvio@student.nhlstenden.com

Inhoud

Voorwoord	3
1. Overzicht en achtergrondinformatie	4
2. Systeem Samenvatting	6
3. Functionele en Technische Vereisten en doel	7
4. Vereisten voor prestatie	11
5. Extra Systeem Vereisten	12
6. Equipment en Software	14
7. Hardware Ontwerp	15
8. Schematische weergave Battlebot	Fout! Bladwijzer niet gedefinieerd.
9. Interpretatie van Hardware Componenten en Schema's	Fout! Bladwijzer niet gedefinieerd.
10. Metingen en Testen	17
11. Bijlagen & overige documenten	Fout! Bladwijzer niet gedefinieerd.

Voorwoord

In dit document presenteren wij het BattleBots-project, ontwikkeld door projectgroep Makro uit klas IC-INF-1A.

Het doel van dit project is het ontwerpen en bouwen van een robot die een testbaan snel en nauwkeurig kan voltooien. In dit verslag worden de opzet van het project, de technische specificaties, en de toegepaste hardware en software uitgebreid toegelicht.

Daarnaast bespreken we de verschillende functionaliteiten die in de robot zijn geïntegreerd, inclusief hun werking en het nut binnen het geheel.

1. Overzicht en achtergrondinformatie

1.1 Doel

Dit technisch ontwerpdocument heeft als doel de technische aspecten van de BattleBot te specificeren die vereist zijn voor de ontwikkeling en programmering, zodat de robot een parcours kan afleggen. Het document fungeert als blauwdruk voor de realisatie van de BattleBot en beschrijft de technische specificaties en gemaakte keuzes met betrekking tot de functionaliteit van de robot.

1.2 Scope

De reikwijdte van dit technisch ontwerpdocument omvat de specificatie van zowel het hardware- als softwareontwerp van de BattleBot. Dit document richt zich op de technische aspecten die essentieel zijn voor de ontwikkeling en programmering, zodat de BattleBot autonoom een parcours met obstakels kan afleggen in estafettevorm.

De robot bevat de volgende onderdelen:

▪Hardware:

- Arduino NANO microcontroller.
- PCB met een Arduino UNO footprint.
- Metalen chassis.
- Powerbank(10.000 mAh).
- 2 Aan/Uit schakelaars voor het logic circuit en het motor circuit.
- Een gripper met een servo motor.
- 4 Neopixels (tricolor leds) voor het signaleren (Hoofd, knipper of remlicht).
- 3 Knoppen (optioneel).
- 3 Leds (rood, geel en groen optineel).
- Motor voor twee elektromotoren.
- 2 Elektromotoren.
- 2 Rotatiesensoren.
- Ultrasonic afstand sensor.
- 8 Analoge lijnvolger sensoren.
- Mini breadboard voor experimentele doelen.

- ESP01 module, AT Wifi Modem (optioneel).
- Bluetooth HC05 module (optioneel).
- Gyroscope I2C (optioneel).
- Oled Display I2C (optioneel).

▪ softwarearchitectuur en -implementatie:

- Arduino IDE.
- C++.

1.3 Projectreferenties

- [Battlebot Moduleboek 2024-2025 P3 NL.docx](#)
- <https://docs.arduino.cc/>
- The ultimate BattleBot Guide

1.4 Contactpersonen

- Module coördinator:
 - Marcel Baron: marcel.baron@nhlstenden.com
- Docenten
 - Aminah Balfaqih: aminah.balfaqih@nhlstenden.com
 - Bart Oerlemans: bart.oerlemans@nhlstenden.com
 - Jaqueline Berghout: jaqueline.berghout@nhlstenden.com

2. Systeem Samenvatting

De basis van het huidige systeem is de BattleBot. Deze is voorzien van een Arduino, afstandssensor, een lijn-sensor en een rotatiesensor) en actuatoren (motoren voor de aandrijving en servo-motoren voor de gripper). Het doel is om deze hardware correct aan te sluiten en te programmeren met de Arduino IDE (C/C++).

3. Functionele en Technische Vereisten en doel

3.1 Samenvatting van Functies

De BattleBot moet in staat zijn tot:

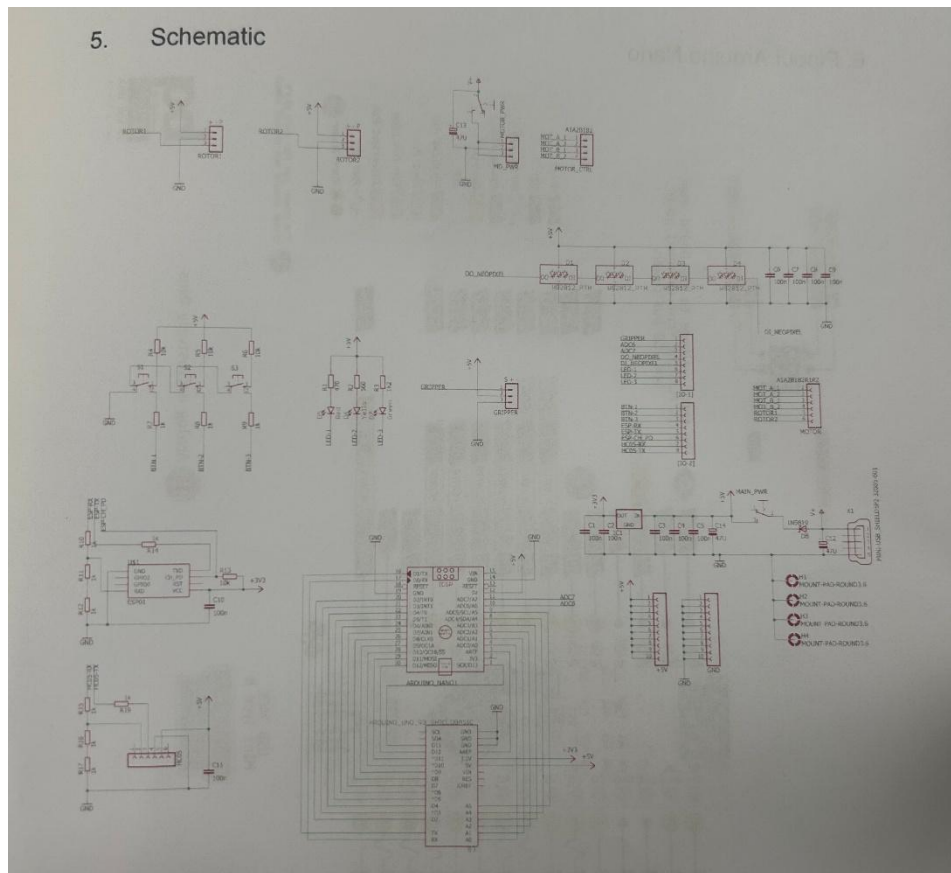
- Zelf kunnen navigeren over een parcours.
- Het volgen van een zwarte lijn.
- Objecten kunnen ontwijken die geplaatst zijn op de baan.
- Het afleggen van een parcours in estafettevorm.
- Deelnemen aan spellen in wedstrijdverband, zoals een fysiek en/of een lijndoolhof.
- Meenemen van een pion door alle onderdelen en doorgeven aan de volgende robot.

3.2 Functionele Vereisten

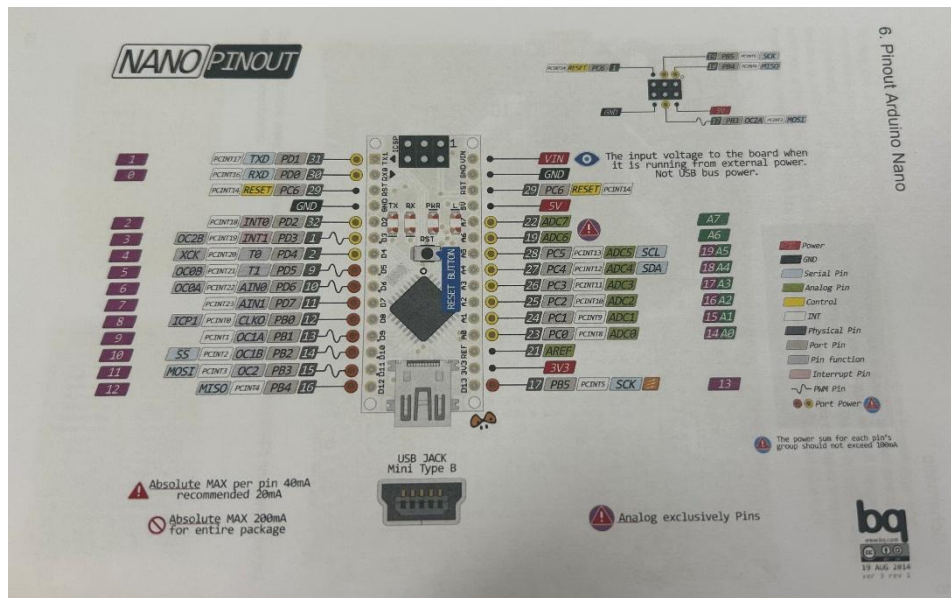
- De BattleBot **moet automatisch zijn**.
- De BattleBot moet een zwarte **lijn volgen**.
- De BattleBot moet zijn eigen **onderdeel van de estafette** kunnen afleggen.
- De **requirements analyse** (zie leeruitkomst Analyseren - hoofdstuk 1.2.2) bevat de (niet-)functionele eisen en de acceptatiecriteria, conform de eisen van de stakeholder(s).
- Requirements worden beschreven als **functioneel/niet-functioneel** en krijgen een **prioriteit** mee met de MoSCoW methode.

3.3 Technische Vereisten

3.3.1 Details



3.3.2 Pinout



Pin	Component	Functie
13	NeoPixel LED strip	Visuele feedback
12	Servo Gripper	Grijpen van pion
6, 11, 10, 5	Motor Driver	Aansturing motoren
A0-A7	Lijnsensorarray	Zwart-wit detectie
9	Echo	Geluidssignaal
8	Ultrasonische Trigger	Afstand meten
7	Ultrasonische Echo	Afstand meten
	Fysieke doolhof	
3,2	Ultrasonische links	Geluidssignaal/Afstand meten
7,4	Ultrasonische rechts	Geluidssignaal/Afstand meten

3.3.3 Beschrijving Systeem

Het embedded systeem van de BattleBot is gebouwd op een Arduino, een microcontrollerplatform dat we programmeren met C/C++ in de Arduino IDE. De motoren zorgen voor de aandrijving van de robot, terwijl de servo's de gripper aansturen. Met de afstandssensor kan de BattleBot meten hoe ver objecten in de buurt zijn. De lijnarray-sensor helpt de robot bij het herkennen en volgen van lijnen op het parcours. De rotatiesensor houdt bij hoe snel de wielen draaien en hoeveel afstand er is afgelegd. Al deze onderdelen zijn aangesloten op de Arduino volgens het hardware-ontwerp.

3.3.4 Werking van Sensoren en Actuatoren

De ultrasone afstandssensor werkt door een geluidssignaal uit te zenden en te meten hoe lang het duurt voordat dat signaal terugkaatst vanaf een object. Op basis van die tijd berekent de robot hoe ver het object weg is.

De lijnarray-sensor bestaat uit meerdere kleine sensoren die naar de grond kijken. Sensoren boven een lichte ondergrond (zoals wit) vangen meer licht op dan die boven een donkere lijn (zoals zwart). Door de waarden van al die sensoren uit te lezen, weet de robot precies waar de lijn onder hem loopt.

De rotatiesensor meet hoe vaak het wiel draait. Die geeft pulsen of een analoge signaal af dat overeenkomt met de rotatie. Door dat te meten, kun je bepalen hoe snel de robot gaat en hoeveel afstand hij heeft afgelegd.

De motoren worden aangestuurd via motor drivers die verbonden zijn met de Arduino. Door de spanning en stroomrichting te regelen, kun je bepalen hoe snel en in welke richting de wielen draaien – zo kan de robot zich voortbewegen.

De servo-motoren kunnen heel precies naar een bepaalde hoek draaien. De Arduino stuurt pulsen om die hoek te bepalen. Zo kan de gripper bijvoorbeeld openen, sluiten of een object oppakken.

3.4 Samenvatting van Impacts

Voor dit project heb je kennis nodig van embedded programmeren in C++ en een basisbegrip van elektronica en sensortechnologie. Omdat de robot autonoom moet kunnen rijden en verschillende spellen moet uitvoeren, kan de software best ingewikkeld worden. Daarom is het belangrijk dat je goed begrijpt hoe de sensoren en actuatoren werken, zodat je de BattleBot op de juiste manier kunt aansturen. Ook het werken in subgroepen – elk verantwoordelijk voor een eigen gameonderdeel – en goed samenwerken binnen het team zijn essentieel om het project tot een succes te maken.

4. Vereisten voor prestatie

- De BattleBot moet het parcours zo snel mogelijk afleggen tijdens de estafette.
- De lijnvolging moet nauwkeurig genoeg zijn zodat de robot binnen de lijnen blijft.
- Obstakels moeten betrouwbaar worden gedetecteerd om botsingen te voorkomen.
- De software moet snel reageren op de gegevens van de sensoren, zodat de robot goed kan sturen en snel kan inspelen op veranderingen in de omgeving.
- Zowel de hardware als de software moeten betrouwbaar zijn om storingen of uitval tijdens de wedstrijd te voorkomen.
- De rotatiesensor moet nauwkeurige informatie geven over snelheid en afstand, zodat de robot zich precies kan verplaatsen.

5. Extra Systeem Vereisten

5.1 Beschrijving systeem

Het BattleBot-systeem bestaat uit de fysieke robot (de hardware) die wordt aangestuurd door software op de Arduino-microcontroller. De software leest de data van verschillende sensoren om een goed beeld van de omgeving te krijgen. Op basis daarvan stuurt de robot zijn actuatoren aan om de juiste bewegingen en acties uit te voeren. De samenwerking tussen hardware en software is hierbij superbelangrijk – zonder die wisselwerking kan de robot niet autonoom functioneren.

5.2 Samenwerking Systemen

De verschillende hardwareonderdelen, zoals sensoren en actuatoren, worden via bekabeling aangesloten op de Arduino volgens het hardware schema. De software schrijven we in de Arduino IDE en uploaden we daarna naar de Arduino. Die software leest de sensordata uit en stuurt de actuatoren aan via de juiste I/O-pinnen. Alles komt samen in de integratiefase: de hardware moet correct zijn aangesloten en de software moet goed samenwerken met die onderdelen om de gewenste functies van de robot mogelijk te maken.

5.3 Mogelijke aanpassingen

De software moet op een modulaire manier worden opgebouwd, zodat we makkelijk aanpassingen kunnen doen als de eisen veranderen. Door te werken met variabelen en parameters – bijvoorbeeld voor snelheden of drempelwaardes van sensoren – blijft de code flexibel en makkelijk aan te passen. Ook is het handig om er bij het ontwerp alvast rekening mee te houden dat er in de toekomst misschien extra sensoren of actuatoren toegevoegd kunnen worden (uiteraard in overleg met de opdrachtgever en via goedgekeurde leveranciers).

5.4 Duurzaamheid code

De programmacode wordt geschreven volgens de C++ Style Guide voor Arduino-projecten, zodat de code goed leesbaar, onderhoudbaar en consistent blijft. Duidelijke codecommentaren worden toegevoegd om uit te leggen hoe de verschillende onderdelen van de code werken. We gebruiken versiebeheer met Git om veranderingen in de code bij te houden en de samenwerking soepel te laten verlopen.

5.5 Systeem Documentatie

Dit technisch ontwerp maakt onderdeel uit van de systeemdokumentatie. De programmacode zal verder goed van commentaar worden voorzien. Het BattleBot Portfolio bevat de ontwikkelde code, aantekeningen, resultaten en evaluaties. Een aparte gebruikershandleiding voor het bedienen van de

BattleBot is niet specifiek nodig voor het eindproduct, maar de documentatie in het portfolio moet genoeg informatie bieden om de werking van de robot te begrijpen.

5.7 instelbare parameters

Het is handig om belangrijke parameters, zoals de snelheden van de motoren, de gevoeligheid van de lijnarray-sensor en de afstanden voor obstakeldetectie, makkelijk aan te passen in de software. Dit kan worden bereikt door constanten of variabelen aan het begin van het programma te definiëren, zodat deze eenvoudig gewijzigd kunnen worden zonder dat de hele code doorzocht hoeft te worden.

6. Equipment en Software

6.1 Equipment

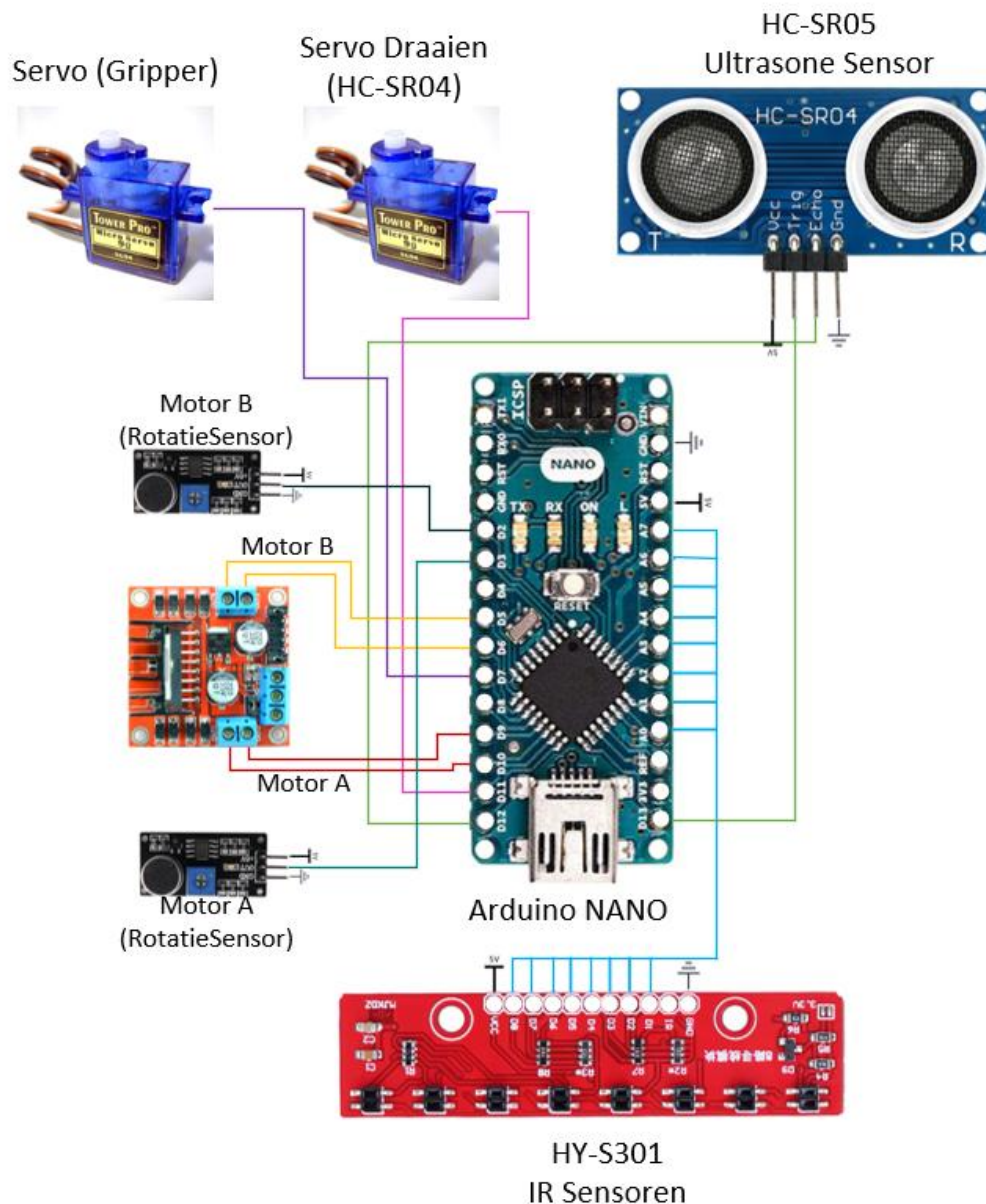
- BattleBot.
- Arduino microcontroller.
- Motoren (voor aandrijving).
- Servo-motoren (voor gripper).
- Afstandssensor.
- Lijnarray-sensor.
- Rotatiesensor.

6.2 Software

- Arduino IDE.
- C++ programmeertaal.
- NeoPixels Arduino library.

7. Hardware Ontwerp

7.1 Schematische weergave Battlebot



7.2 Interpretatie van hardware en schema's

Het technisch ontwerp bevat schemas, maar ook een **beschrijving van de sensoren en actuatoren**. Ook de **interpretatie van de hardware componenten en schema's** worden gegeven. De BattleBot met verschillende componenten, waaronder een **microcontroller (Arduino)**, **motoren voor de aandrijving**, **servo-motoren voor de gripper**, een **afstandssensor**, een **lijnnarray-sensor** en een **rotatiesensor** om de snelheid en afstand van de wielen te meten. De verbinding tussen deze componenten, zoals hoe de sensoren informatie doorgeven aan de microcontroller en hoe de microcontroller de actuatoren aanstuurt op basis van die informatie, dienen te worden uitgelegd aan de hand

van het schema. De handleiding '**The Ultimate BattleBot Guide**' die bij de apparatuur wordt meegeleverd, zal hierbij waarschijnlijk een belangrijke bron van informatie zijn.

10. Metingen en Testen

10.1 Lijnsensor

Om er voor te zorgen dat de lijnsensoren goed werken in alle situaties, moet er worden gekalibreert. Dit zorgt ervoor dat de lijn sensor goed het verschil kan zien tussen non reflectief en reflectieven kleuren.

Het kalibreren doet de robot door over zwarte en witte lijnen heen te rijden, om daarmee een drempel waarden te krijgen.

Dan wordt er een hysteresis aan toegevoegd om er voor te zorgen dat kleine verschillen in de reflectie geen grote impact hebben op de robot.

Ook kan er worden gebruik gemaakt van tijdens het rijden te kalibreren.

Dit kan worden gedaan door het lezen van een gemiddelde waarden en hier mee een bovenste en onderste drempel waarden uit te rekenen.

10.2 Rotatiesensoren

De rotatiesensoren worden gebruikt om de draaisnelheid te berekenen.

De sensor berekend de snelheid door pulsen te geven tijdens de rotatie.

Hier mee kunnen we enige snelheids verschillen tussen de wielen berekenen en deze corrigeren.

10.3 Meerdere metingen

Om er voor te zorgen dat de metingen van de sensoren accuraat zijn moeten (als het mogelijk is) de sensoren meerdere keren de zelfde meting geven.

Hier mee kunnen wij er voor zorgen dat er geen fout positief ontstaat wanneer de robot bezig is.

Dit wordt niet altijd gedaan zoals bij de lijnsensoren omdat het niet mogelijk is vanwege de snelheid waar de robot op moet kunnen reageren.

