**P&O 2:**

**100/1**

**analoge reflectiesensor**

**poorten nakijken**

**batterijen**

**Online shop:** <https://irkulak.be/po2>

-> Pull from DanielWangKUL :Epic-Waggie

**TO DO**

takenstructuur! → DONE

teamkalender → NOG AANPASSEN IN LATEX

Gantt chart => uitbreiden → IEDEREEN NAKIJKEN

samenvatting klanten vereisten → IEDEREEN NAKIJKEN

Overzicht ontwerpspecificaties → IEDEREEN NAKIJKEN

Model schetsen → DONE

wekelijkse verslagen→ Latex? docs?

materiaallijst +- afwerken

bieden

HANDIGE SITES:

<https://www.ni.com/pdf/manuals/376047c.pdf>

<https://www.youtube.com/watch?v=8VT8DBMc4Bw&ab_channel=LabVIEW%26MULTISIM>

<https://forums.ni.com/t5/Showcasing-Student-Innovation/Using-LabVIEW-and-NI-myRio-to-create-a-Robot-Car-Wi-Fi/ta-p/3596844>

<https://www.youtube.com/watch?v=UCqFck0CLpc&ab_channel=NTS>

<https://www.youtube.com/watch?v=BFgelRQxJ_E&ab_channel=NTS>

**Takenstructuur**

CODE:

bocht nemen, drie richtingen(links ,rechts ,rechtdoor)

lijndetectie(onderscheid tussen volglijn en stoplijn, zwart-wit)

verkeer herkennen ⇒ auto herkennen(obstakel definiëren) en eventueel stoppen (stoppen voor obstakels), ook afstand kunnen houden van het obstakel

lichten herkennen ⇒ groen doorrijden (bocht nemen of voorwaarts, rood wachten tot groen)

stoplijn herkennen ⇒ afhankelijk van licht (stoplijn negeren bij groen licht en bij rood stoppen aan de stop lijn)

snelheid controleren (interface) ⇒ niet te traag

interface maken ⇒ Om de richting te kiezen + stop/start

draadloze besturing ⇒ vanop afstand informatie overbrengen van en naar de wagen

BOUWEN:

3D model opstellen

demo maken

model schetsen

technische tekeningen ⇒ hoe concreet de wagen eruit ziet en gebouwd moet worden

berekeningen ⇒ berekeningen voor het maken van bochten, afstand nodig om te stoppen,

MATERIAAL:

onderdelenlijst

versiering

PRESENTATIE:

tussentijds verslag

eindverslag

tussentijdse presentatie

eindpresentatie

wekelijkse vooruitgang

ALGEMEEN:

bieden voor bij de verkoop

opgave lezen

eerste deadline

**Verantwoordelijkheidsstructuur**

CODE: Aimé en Alycia

PRESENTATIE: Prathap

MATERIAAL: Daniel

BOUWEN: Jonas en Alycia

**Klantenvereisten**

Zorg voor een zelfrijdend voertuig dat een voorgeprogrammeerde weg kan afleggen in een modelstad. Daarnaast moet het ook de verkeerslichten kunnen detecteren en correct interpreteren. Vervolgens moet het ook de andere voertuigen/obstakels kunnen detecteren en indien nodig stoppen.

**Ontwerpspecificaties**

We willen een gemotoriseerd voertuig op vier wielen, dat in verschillende richtingen draadloos kan sturen. Bovendien moet het voertuig obstakels en rode lichten kunnen detecteren en hierop anticiperen. Indien een rood licht gedetecteerd wordt zal hij de nodige afstand voor te stoppen moeten bepalen en remmen. Bij groen licht mag de auto verder rijden en eventueel een bocht inslaan. Wanneer het autootje een obstakel tegenkomt (zoals een andere auto), zal dit opnieuw door de sensor gedetecteerd worden en zal het autootje hierop anticiperen door te remmen (en stoppen). Het zal automatisch een voorgeprogrammeerde route op een lijn moeten afleggen met uitzondering van de kruispunten, met eventueel nieuwe informatie die draadloos naar het voertuig gecommuniceerd zal worden. We maken gebruik van een vierwielaandrijving zonder versnellingen en een stevige chassis. Het maximaal budget voor het wagentje bedraagt 3500 elementen

* 4 wielen
* motorisch aangedreven
* sturen: links en rechts
* remmen en stoppen ⇒ rood licht of obstakel
* sensoren(verkeerslicht, objecten)
* lichten detecteren: groen, rood
* maximaal budget : 3500 elementen
* automatisch rijden met draadloze besturing
* lijnen volgen
* beweging in het straatverkeer kunnen detecteren
* stoppen als er een obstakel in de weg staat
* bocht nemen of rechtdoor verder rijden na kruispunt (met stoplicht)
* versnellingen? niet nodig
* vierwielaandrijving?
* Stevige chassis

MATERIALENLIJST

Micro Metal Gear Motor 100:1 HP 2

Wiel 42x19mm 2

Ball Caster 1

TCS34725 Kleur sensor BOB 1

QTR-8A analoge reflectie sensor array 1

JST Connector 2p 2mm M haaks 1

Breadboard Full-size 1

Universele Hub voor 3mm assen 2

NI MyRio 1

Optische Afstandssensor (digitaal) 1

Oplaadbare LITHIUM-ION 4

Motorschield IC 2

Male headers 10 4

Micro metal gear motor beugel 2

bedrag: 2100

ball caster (60 credits) 1

(Rpi camera (150 credits) 1 mss beter LED kleursensor ipv camera (fourier transformatie ingewikkeld te implementeren) )

Powerbank 1

NI MyRio 1

Micro Metal Gear Motor 50:1 HP 2

Wiel 42x19 mm 2

breadboard full size 1

universele hub → 2

QTR digitaal sensor → 1 (lijnsensor)

optische afstandssensor analoog→ 1 (

ADC → 1 (analoog naar digitaal converter)

motorshield → 2

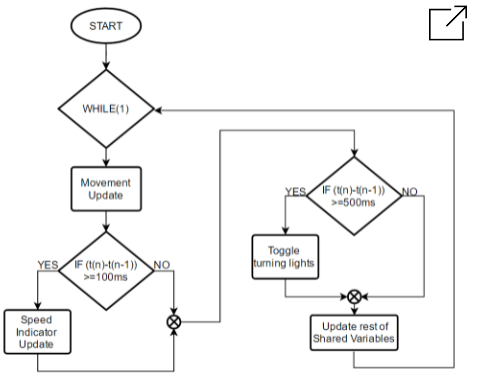
male headers → 4 (breadboard verbinden aan controller

kleursensor BOB → 1 (voor de lichten)

ONTWERP

We maken gebruik van een tweewielaandrijving zonder versnellingen en een stevige chassis. 2 wielen en balcaster

De kruispunten zullen knipperen aan 1Hz. licht op 7,5 cm hoog



WEKELIJKSE VERSLAGEN

TODO

Bieding

CAD-ontwerp

verschillende onderdelen in solid edge

individuele touch: medium → chassis ontwerpen

brainstormen over algoritme

(hoe gaan we het kruispunt oversteken?)

schakeling breadboard

financieel rapport voor eindverslag

penningmeester

keuzes materialen verantwoorden

verslag → latex document opstellen, structuur => uitleg geven over elke keuze,

presentatie

verloop baan

9/03

TO DO

Alles in gitkraken overzetten → labview verbinden?

chassis → aanpassen aan juiste afmetingen

labview code concreet uitwerken

BAAN

1m straat

9 kruispunten

→ te overbruggen afstand = 50 cm

7.5 cm hoogte verkeerslicht → kleursensor op hoogte hangen

stopafstand → stoppen aan lichten 0,5 m oversteken en dan 1 m rijden en weer stoppen→ kijken naar hoelang de auto nodig heeft om te stoppen → stopt redelijk snel

CHASSIS

PLA filament

SOLID EDGE

Afmetingen

Breadboard

volg lijn —>

*verbetering verslag*

**Klantenvereisten**

De modelstad bestaat uit wegen waarlangs verkeerslichten staan, na de verkeerslichten is er een kruispunt met vier richtingen.

**Ontwerpspecificaties**

De wagen moet rechtdoor kunnen rijden en bovendien links en rechts afslaan.

Licht

De kleursensor kijkt welke kleur het licht heeft en op basis hiervan zal de wagen vertrekken of wachten. *Actuoren? Technische vereisten?*

‘Indien een rood licht gedetecteerd wordt zal hij de nodige afstand voor te stoppen moeten bepalen en vertragen en stoppen bij de stoplijn.’ => deze zin klopt niet

Verkeerslicht staat rechts è stoppen aan de streep en kleursensor detecteert het licht.

**Takenstructuur**

Implementeren reflectie sensor

Implementeren afstandssensor

Implementeren kleursensor

V**is**ualisatie

Sensor inputs => while loop (1 grote while loop waarin alle functies gebundeld zitten)

***Functies:***

*1.* *Reflectie sensor (lijndetectie) input =>*

o Acht minisensoren controleren of de wagen op de lijn zit (wit of zwart)

o Reactie: rechtdoor blijven rijden als de wagen recht (parallel) op de lijn rijdt of wagen X graden draaien (links of rechts) indien schuin op de lijn (corrigeren als een van de sensoren de lijn niet meer ziet)

o Als de wagen X meter van de lijn is afgeweken, wordt de wagen Y graden gedraaid zodat de auto terug in de juiste richting rijdt

o Na een kruispunt moet er mogelijks een bocht genomen worden:

De wielen moeten gedurende X seconden Y graden draaien.

o Daarnaast moet ook nog de stopstreep herkend worden:

Er moet worden afgeremd door middel van de motor stil te leggen.

Dit moet gebeuren vanaf de stopstreep wordt opgemerkt door de sensor voorop de wagen.

**While loop**:

§ **Zolang** de **stopstreep** **niet** wordt ge**zien** (0) => **doorrijden**

§ Vanaf de **sensor** de **streep** **detecteert** (1) => **motor stilleggen**

*2.* *Kleursensor input =>*

Vanaf de wagen aan de stopstreep staat, kijkt de kleurensensor of het verkeerslicht op rood of groen staat, dit wordt aangeduid met een Boolean => rood of groen:

o Als het licht op rood staat, loopt er een while loop en duidt de Boolean rood aan:

o Zolang het licht op rood staat => stil blijven staan

o Vanaf het licht op groen springt => motor starten (versnellen) tot aan een bepaalde snelheid

o **If licht == rood:**

§ **While not OK (Boolean: rood):**

· **If licht == rood:**

o **Doe niets**

· **Else:**

o **Start de motor en versnel tot X km/h**

o **OK (Boolean: groen) => functie zal stoppen**

o Als het licht op op groen staat:

o Motor starten (versnellen) tot aan een specifieke snelheid

o **If licht == groen:**

§ **Start de motor en versnel tot X km/h**

*3.* *Afstandssensor input =>*

Deze functie detecteert voertuigen of eventuele obstakels op de baan. Als dit het geval is, moet er worden geanticipeerd door het obstakel te ontwijken.

o Eerst gebruiken we de if/else structuur om te kijken of er een obstakel is

o Er wordt vervolgens gebruik gemaakt van een while loop: er moet namelijk voortdurend in de gaten gehouden worden of het obstakel zich nog op de baan bevindt.

o Als er zich binnen 10 cm een stilstaand obstakel bevindt => motor stilleggen

o (Meer specifiek: als er binnen 10 cm een voorwerp trager beweegt dan de auto => motor stil)

o Het gevolg van de while loop die voortdurend controleert of er een obstakel is op 10 cm:

o Vanaf het obstakel verdwenen is, kan de wagen terug vertrekken

o **If obstakel op < 10 cm:**

o **Motor stilleggen en stoppen**

o **While obstakel op < 10 cm:**

§ **Doe niets**

o **Motor aanzetten en vertrekken**

o **Else:**

o **Verder rijden (niets doen)**

*Naar links/ rechts sturen*

Lijnsensor geeft informatie => we zitten niet meer op de lijn

Daarnaast vertelt de sensor hoeveel graden naar welke kant de wagen afwijkt: bv left 20°: de wagen is links van de lijn en is twintig graden afgeweken van de lijn.

Nu zal de wagen reageren door naar links of rechts bij te sturen.

**If info == afgeweken:**

**If zijde == left:**

**Draai X graden naar rechts.**

**If zijde == right:**

**Draai X graden naar links.**

*Motoren aansturen*

Signaal van de sensoren dat de wagen mag vertrekken:

We gebruiken een boolean: stilstaan/rijden

Wanneer de bool stilstaan aangeeft, moet de auto stil blijven staan of de motoren stilleggen indien de wagen aan het rijden is. Als de bool zegt rijden, blijft de wagen verder rijden of vertrekt de auto.

Motoren stilleggen (stoppen):

**If bool1 == stilstaan and bool2 == motorenAan:**

**Bool2 == motorenUit:**

**Leg de motoren stil**

Motoren aanleggen (vertrekken):

**If bool1 == rijden and bool2 == motorenUit:**

**Bool2 == motorenAan:**

**Start de motoren**

*Snelheid controleren*

Vanaf de sensor aangeeft dat de wagen moet stoppen, moet er gemeten worden hoeveel de uitbolafstand bedraagt.

**uitbolAfstand = constante x snelheid km/h**

**If bool==stoppen and uitbolAfstand == afstandTotStopstreep:**

**Bool2 == motorenUit:**

**Leg de motor af**

????? stopt de wagen niet onmiddellijk, de sensoren werken toch niet vanop afstand