

# Eindverslag

---

**Steven Verheyen**  
**Tim Asscherickx**  
**Jeroen Van Wolput**

Projectverantwoordelijke: - M. Smets  
- M. Luyts

## Overzicht code

De nieuwste codes staan steeds op GitHub:

Uno: <https://github.com/VerheyenSteven/Smart-Systems/tree/master/programma>

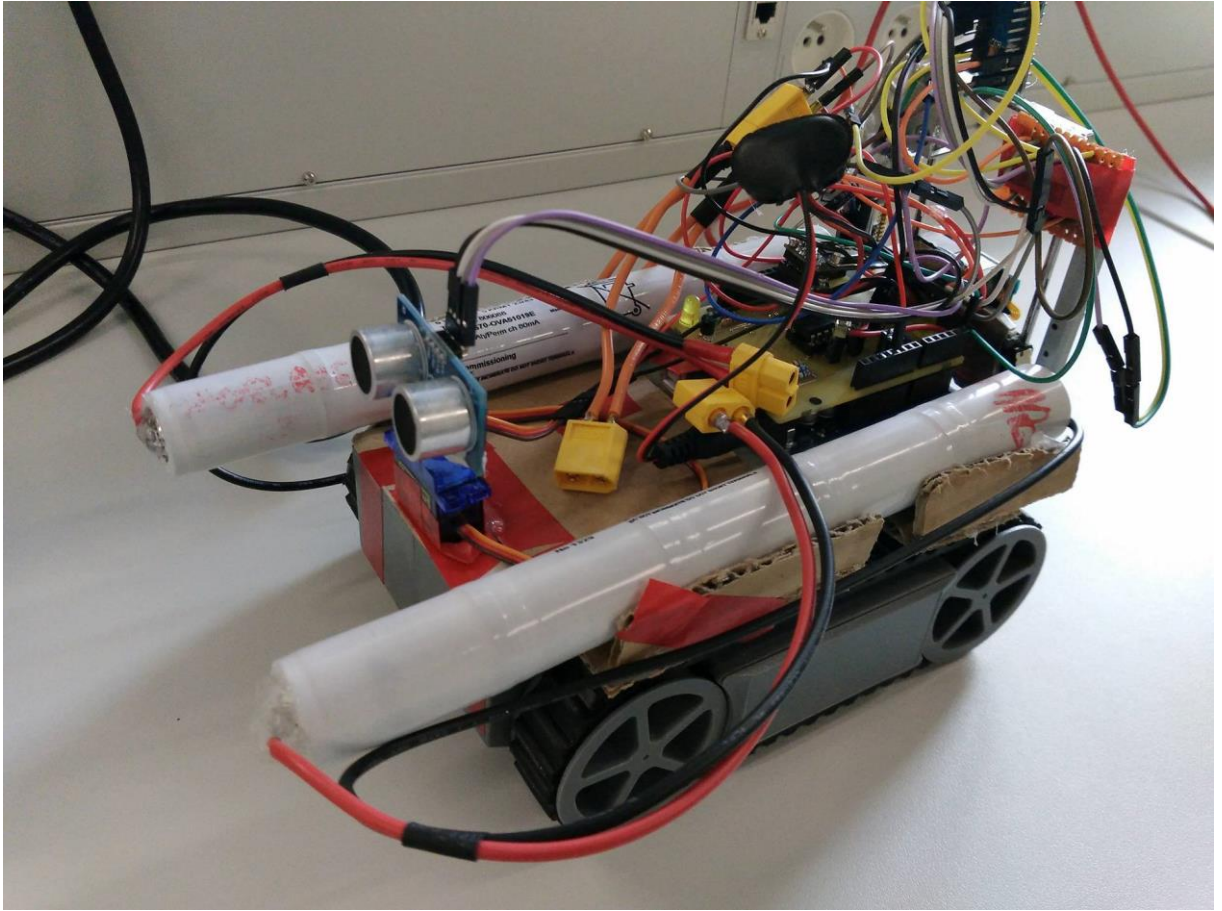
Nano: <https://github.com/VerheyenSteven/Smart-Systems/tree/master/programmaNano>

Joystick: [https://github.com/VerheyenSteven/Smart-Systems/tree/master/joystick\\_Transmittor](https://github.com/VerheyenSteven/Smart-Systems/tree/master/joystick_Transmittor)

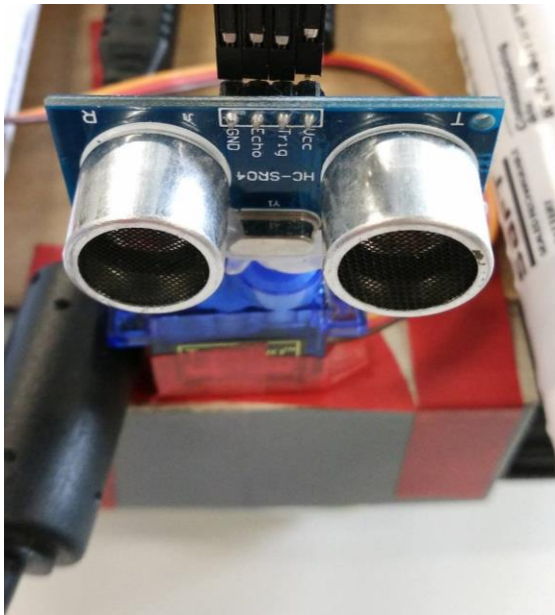
## Wat kan onze auto

1. Automatisch rijden: Onze auto kan zelfstandig rijden en een uitweg zoeken in een bepaald parcouer.
2. Rijden met een joystick: Via een RF module kan je draadloos onze wagen besturen.
3. Als onze auto naar links draait, dan zal de auto naar links pinken en hetzelfde gebeurd met rechts.
4. Als we automatisch rijden, dan krijgen we een liedje te horen. Als we via joystick rijden, dan zal onze auto race geluiden maken. Dit hebben we gedaan door een mp3-trigger.
5. Onze auto tekent een lijn als hij gewoon rijdt. Komt hij een object tegen, dan zal hij een stippellijn beginnen tekenen.
6. Onze auto kijkt rondt om een goede uitweg te zoeken uit een parcouer.
7. Zigzag patroon.

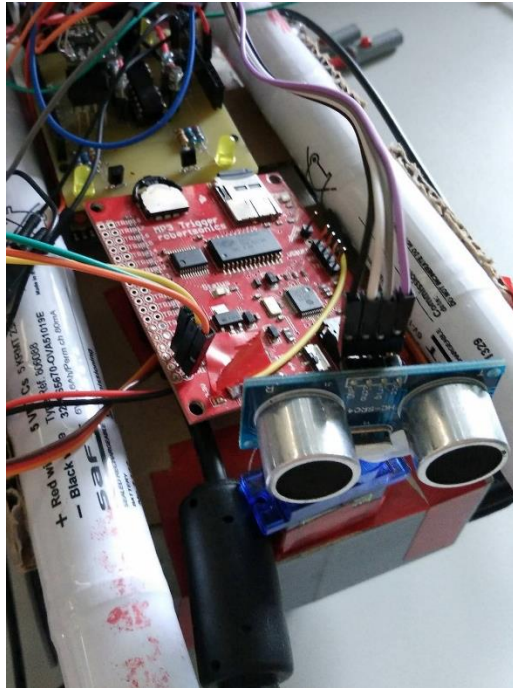
## Afbeeldingen en uitleg functies



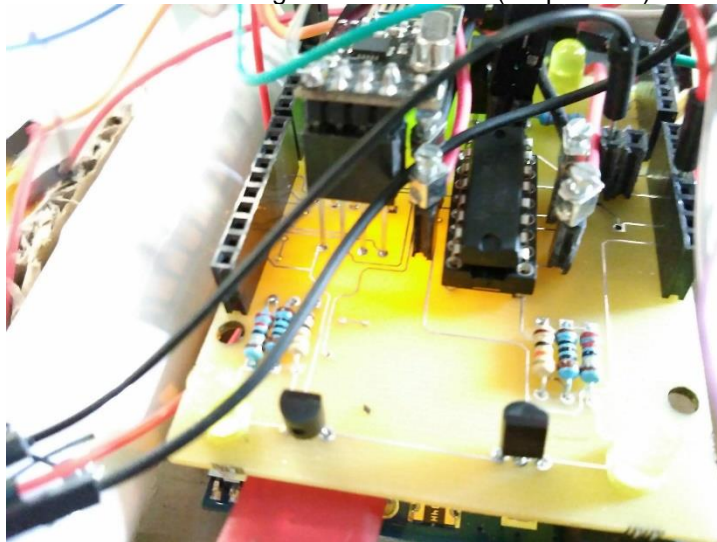
Bovenstaande foto toont onze wagen.



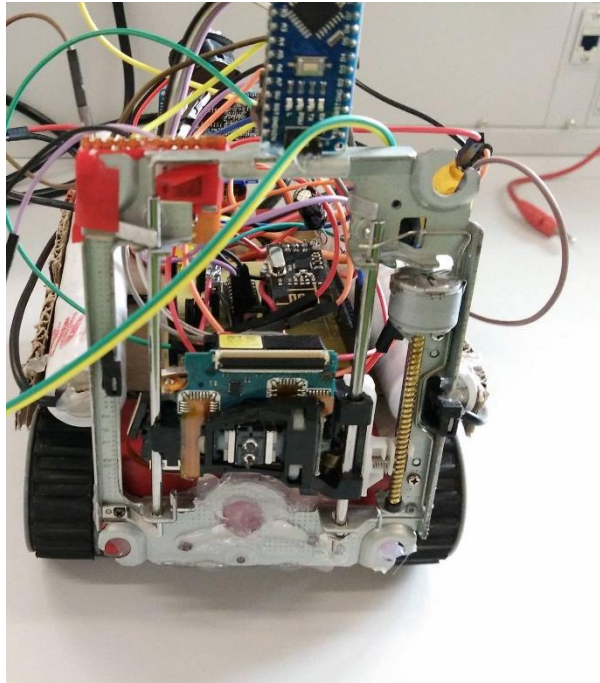
- De twee grote witte onderdelen op onze auto zijn de batterijen. Deze zijn oplaadbaar en geven beide een spanning van  $\pm 7$  volt. Ze zijn oorspronkelijk afkomstig van noodlichten die ze gebruiken in gebouwen.
- Vooraan vinden we de servo-motor terug met een sonische sensor bovenop. Dit hebben we nodig voor obstakels te detecteren. De servo-motor laat de sonische sensor rondkijken.
- In het midden van onze wagen staat de mp3-trigger (niet zichtbaar op bovenste foto). Deze beheert welk geluid wordt afgespeeld en wanneer.



- Achter de mp3-trigger vinden we onze Arduino Uno en pcb-design terug. Op de pcb vinden we 4 pinkers terug en de H-Brug. Dit is het hard van onze wagen. Hier worden de motors aangestuurd en meer. (zie punt 7.4)



- Het achterste deel van onze wagen is ingepalmd door onze optische drive van een Playstation 3. Deze beheert het stippelijnpatroon. De optische drive zijn stappenmotor wordt geactiveerd, wanneer onze wagen een obstakel ontdekt.

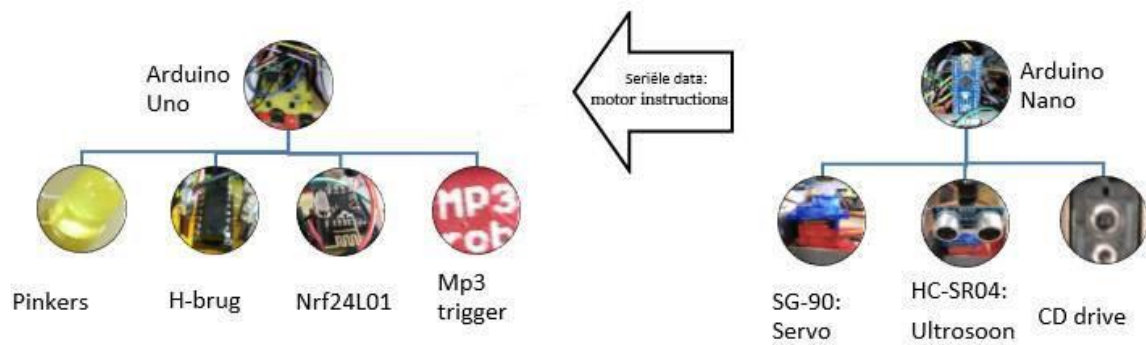


- Doordat we te weinig pwm's hadden, hebben we een extra arduino Nano moeten gebruiken. Deze staat op onze optische drive en communiceert met onze Uno via de seriële poort. De Nano beheert de optische-drive, sonische sensor, de servo en stuurt commando's door hoe de wagen moet rijden.



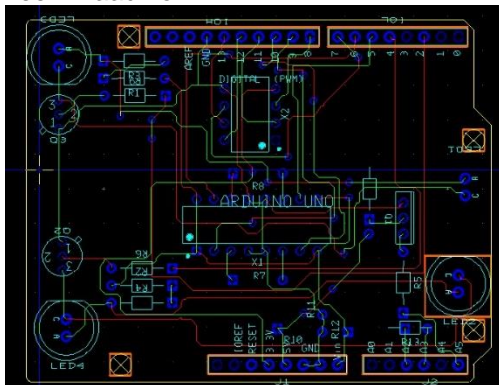


## Overzicht NANO en UNO



## PCB

Voor ons PCB te designen wilden we eerst EAGLE gebruiken, maar Meneer Smets had ons aangeraden om Ultiboard te gebruiken. Met dit in gedachten hebben we een Arduino Uno shield template gevonden voor Ultiboard. Onze hardware was klaar dus we konden ook direct beginnen met een PCB te designen. Na het indienen hadden we als opmerking gekregen dat sommige banen voor de motorbesturing te klein waren, maar het voor dit project niet zo heel veel kwaad kon.



## Eindervaringen en pijnpunten

Sprint 1 en 2 gingen heel goed bij onze groep. Alles verliep vlot en we namen snel teveel hooi op onze vork. De servo maakte vreemde bewegingen en we hadden teveel extra's bedacht, maar geen rekening gehouden met het aantal beschikbare pinnen van de Uno. De communicatie tussen de extra Nano en Uno verliep ook niet vlekkeloos. We hadden steeds het probleem dat onze auto rondjes reed. Via toeval zijn we er dan achter gekomen dat onze Uno te weinig stroom kreeg en geen communicatie had met de Nano, met de rondjes draaien als gevolg. We dachten dat we onze auto volledig hadden kunnen klaar krijgen voor de voorstelling. Op het eerste moment ging onze auto heel goed en werkte zoals we zelf wilden, tot hij een obstakel tegenkwam. Hij draaide steeds 90 graden naar links en dan weer naar recht en zo bleef hij doorgaan, tot 15 seconden en dan ging hij weer verder zoals eigenlijk de bedoeling was. We dachten dat dit weer aan de communicatie tussen Nano en Uno lag. We zijn die dag snel beginnen zoeken naar een oplossing. Deze hebben we helaas niet gevonden

en tegen het einde van de dag deed onze wagen bijna niets meer. We hebben dan rustig na de voorstelling eens gaan kijken naar de oorzaak en toen bleek dat alle waardes werden opgeslagen in een Seriële buffer. Onze wagen kreeg in korte tijd teveel waardes binnen en sloeg deze op. Dat verklaarde het vreemde rijgedrag. Nu hebben we onze auto toch verder in orde gekregen en werkt hij zoals we in het begin gehoopt hadden.