**dia 1:**

Afgelopen semester kregen we ons bachelorproject voorgeschoteld. De opdracht bestond

erin een robot te maken die zelfstandig een traject binnen twee witte lijnen op een zwarte mat kon afleggen.

Onze robot zou tijdens het rijden bepaalde RFID-tags moeten kunnen inlezen en versturen via Bluetooth naar een Raspberry Pi. RFID [9] staat voor Radio-frequency identification en gebruikt het elektromagnetisch veld om een object te identificeren.

Ook de snelheid van de robot dient geregistreerd en verstuurd te worden via Bluetooth.

De bedoeling was dat onze robot zo snel mogelijk een volledige ronde kon afleggen. Deze tijd werd dan vergelijken met die van de medestudenten.

Verder dienden we onze eigen pcb te routen en te vervaardigen. Deze pcb

moest bestaan uit de microcontroller, in dit geval een Arduino, en een motershield die

het wagentje aandrijft.

Op deze manier werd onze robot Stampertje gedoopt, onze robot in motion.

**dia 2:**

Hoe start je nu aan zo'n project. Op het internet vonden we een cyclus die een ingenieur doorloopt bij het maken van een project. En inderdaad deze cirkel hebben we meermaals doorlopen. (vandaar de lambda in de cirkel).

We merkten op dat in het schema design en analyze soms omgewisseld werden. Vaak analyseer je een probleem om vervolgens iets te ontwikkelen dat het probleem oplost. Maar ook situaties waarbij het design voor de analyze staat kwamen voor. Bijvoorbeeld bij het gebruik van de RFID-lezer of HC05 module (module die instaat voor de bluethoot), daar was de design (de module) reeds ontwikkeld, deze dienden we dan te analyzeren alvorens we een softwaredesign konden implementeren. Om de reden dat de volgorde van design en analyse soms wisselen, nemen we ze samen voor de bespreking die volgt.

In het ontwerpen van onze robot zijn er twee grote peilers: de softwarepeiler en de hardwarepeiler dienden ontwikkeld te worden.

**dia 3:**

We starten met het uitleggen van de softwarepijler. Voor we met het schrijven van de code konden beginnen moesten we eerst een goed idee hebben van het design. We zijn dus begonnen met het overleggen hoe we de grote lijnen van het project gingen invullen. Met hoeveel sensoren werken we? Rijden we op de volle lijnen of op de stippenlijn? Lezen we analoog of digitaal in?   
Uiteindelijk hebben we besloten dat we gingen beginnen met redelijk simpele sensormodules met twee TRCT-5000 sensoren op. Die sensoren gaan we analoog inlezen omdat we dan veel preciezer kunnen werken dan bij digitale inlezing. Zo konden we bijvoorbeeld de sensor naast zwart en wit een tussenstaat laten inlezen waarbij de sensor overgaat van 1 kleur naar de andere. Dit gaf ons meer ruimte om de robot te laten reageren op het parkour.   
In het begin zijn we ook begonnen met beschrijvende code ipv een PID-regelaar. Dit was vooral omdat ik bij een vorig robotprojectje op een gelijkaardige manier had gewerkt en we dus daarom daar meer vertrouwen in hadden.

**dia 4:**

Op basis van de eerste designversie hebben we gewerkt met een extra klasse genaamd Motor. In deze klasse schreven we enkele basisbewegingen zoals vooruit rijden, achteruit, stilstaan en daarnaast ook enkele methodes voor het echt volgen van de lijn waarin we aan de hand van onze sensorwaarden een bepaalde hoek berekende die de robot zou moeten maken en die daarna dan doorgeeft naar de motoren. Deze methode van werken is eigenlijk niet zo goed. Hij kan veel gestructureerder en stijgt nogal rap in complexiteit naarmate we het gedrag meer willen beschrijven. Dat is de reden waarom we alles hebben omgegooid en overgeschakeld hebben naar een tweede aanpak waarin we vooral digitaal inlazen, extra sensoren toevoegden en de PID-regelaar gebruikten. We maakten ook enkele extra klassen aan om een betere structuur in onze code te krijgen. Zo maakten we de classes SensorModule, Motor en PID aan die we in de main op elkaar afstellen. Deze methode was veel overzichtelijker en makkelijker om de verscheidene parameters van de PID-regeling en de errormap in te stellen. Eens dat die ingesteld waren voor de eerste sensormodule was het ook eenvoudig om een tweede module toe te voegen.

**dia5:**

Op deze dia kan je zien hoe onze loop er in de grote lijnen uitziet voor de aansturing van de motoren. In de setup declareren we alle parameters en geven we ook een basissnelheid mee. Deze basissnelheid is de snelheid waarmee de robot zou rijden op een rechte lijn. Na de setup beginnen we met een refreshValues-methode waarin we de sensorwaarden van de twee sensormodules eens updaten. Daarna kijken we aan welke kant van de baan we aan het rijden zijn en welke sensormodule dus voorrang heeft op de andere. Als een bepaalde sensor van de module met prioriteit wit aantoont komt dat overeen met een bepaalde waarde uit de errormap. Die waarde stijgt naarmate dat de sensor dichter bij het midden van de baan staat. Dit doen we omdat naarmate de robot verder over de lijn aan het rijden is, er harder gecorrigeerd zal moeten worden.   
De voorlaatste stap is dan een berekening via de PID-regelaar. Eigenlijk een PD-regelaar. Want de Integrator hebben we buiten beschouwing gelaten omdat het instellen van de parameters dan te lang zou duren en de PD-regelaar al een goed resultaat gaf. De waarde die we dan uiteindelijk bekomen is de correctie die we op de basissnelheid toepassen. “Toepassen” betekent dat we de correctie bij de snelheid van motor 1 optellen en bij de snelheid van motor 2 aftrekken. Zo krijgen we dus de correcties die onze robot maakt bij het volgen van het parkour.

**dia6:**

Voor het tweede gedeelte van de opdracht, het detecteren van RFID-tags en communiceren via bluetooth, maken we gebruik van een aparte arduino om onze code op te plaatsen. Voor deze aparte lijn van code hebben we ook enkele aparte classes aangemaakt: voor de RFID en bluetooth om onze code overzichtelijk te houden. De eerste stap in deze lijn is ook weer het declareren van al het benodigde in de Setup. De volgende stap is het berekenen van de snelheid van ons robotje. Dit doen we met behulp van 1 TCRT-5000 sensor die een witte plakker detecteert op de zijkant van een van de wielen. Na elke omwenteling houden we de tijd bij tussen opeenvolgende passages van de plakker. Met behulp van die tijd en de omtrek van de wielen kunnen w8e de snelheid bereken. Als we de snelheid hebben kijken we vervolgens ook of er een RFID-tag te vinden is. Indien niet sturen we enkel de snelheid door via de bluetooth-communicatie. Als er wel een aanwezig is lezen we die in, en geven we zowel het nummer van de tag als de snelheid door. Als dit is afgerond starten we terug vanaf het begin van de loop.

**dia7:**

Voor het PCB-ontwerp zijn we vertrokken van een arduinoschema en een schema van

de gebruikte motershield dat was meegeleverd met de slides op toledo. Beide schema's

voegden we samen tot één printplaat en dus één kicad project. Zaken die irrelevant waren voor

onze printplaat hielden we achterwege. Zo werd in het arduinoschema bijvoorbeeld de usbconnectie achterwege gelaten. In eerste instantie besloten we niet te veel te wijzigen

aan de bestaande schema's, zo behielden we bij een eerste ontwerp bijvoorbeeld alle

pinnen. Een eerste cyclus was na veel werk doorlopen. We verkregen een eerste printplaat waarbij aanvankelijk alles leek te werken. Het led blink programma werkte op de arduino en de motershield reageerden op lichtwijzigingen van de led.

**dia 8:**

Maar toen stootten we op een probleem, de RFID-lezer die we wenste te gebruiken maakte gebruik van de SPI-communicatie pinnen. SPI staat voor Serial Peripheral Interface en is een synchrone seriële datalink tussen ten minste twee mediums, bij ons de arduino en de RFID-module. Enkel de pinnen 10-11-12 en 13 beschikten over zo'n interface. We hadden pin 10,11 en 12 nodig voor de RFID lezer, maar pin 10 en 11 beschikten ook over PWM (Pulse Width Modulation) en we gebruikten deze reeds om de motershield aan te sturen. Op deze manier hadden we geen pinnen meer over voor de communicatie met de RFID lezer en overwogen we twee mogelijkheden.

1. Het maken van een nieuwe printplaat
2. Het gebruiken van een andere Arduino

We kozen ervoor een nieuwe printplaat te vervaardigen en doorliepen terug de cyclus van de vorige slides. Nu hadden we bij het experimenteren echter minder succes. De arduino van de printplaat werkte niet en we kregen hem maar niet aan de praat. De spanningsniveau's leken te kloppen, maar het geheel werkte niet. We hebben nog geprobeerd de afsluitcondensatoren van het kristal te wijzigen, in de hoop dat we de klok stabiel kregen, maar dat is niet gelukt. Uiteindelijk zijn de dan maar teruggevallen op optie twee: het gebruik van 2 Arduino's.

**dia9:**

De eerste printplaat die aanvankelijk werkte kreeg problemen met de motershield, bij het maken van de printplaat hebben we onder de kroonsteentjes moeten foefelen met kleine draadjes, bij het aansluiten van de draden, OUT1 en OUT2 waren aan elkaar verbonden. Geen probleem, we vervingen de kroonsteentjes door dikke draadjes zoals te zien op de foto en testten opnieuw. Hier liep het echter mis, bij het monteren van de batterij is heel even de GND met GND kortgesloten, waarna alle spanninsniveau's niet meer klopten en de ATMEGA stuk was.

Aan een nieuwe pcb zijn we niet meer begonnen.

**dia10:**

Het uiteindelijke resultaat was een wagentje dat de baan volgde en dat de RFID tags

en zijn snelheid doorstuurde naar de Raspberry Pi. Het is ons niet gelukt de printplaat

te integreren op ons wagentje.

Uit de opdracht hebben we concreet electronica leren ontwerpen en ontwikkelen. Dit is

een enorme meerwaarde geweest. We leerden eveneens fouten opsporen op de printplaat,

ook al konden we niet met zekerheid zeggen wat de fout op de tweede printplaat was (we

vermoeden onstabiele klok), toch leerden we na te gaan of alle spanningen en signalen al

dan niet klopten. We leerden het theoretische concept van PID regeling in praktijk toe te

passen en te integreren in software. Het lezen van datasheets, het zelfstandig opzoeken

van informatie, het bespreken over een project,. . . zijn allemaal zaken die dit project

ons bijbrachten. Een project is zeker onmisbaar aan het einde van een bacheloropleiding

om een groot deel van de verworven elektronica kennis in praktijk te brengen.