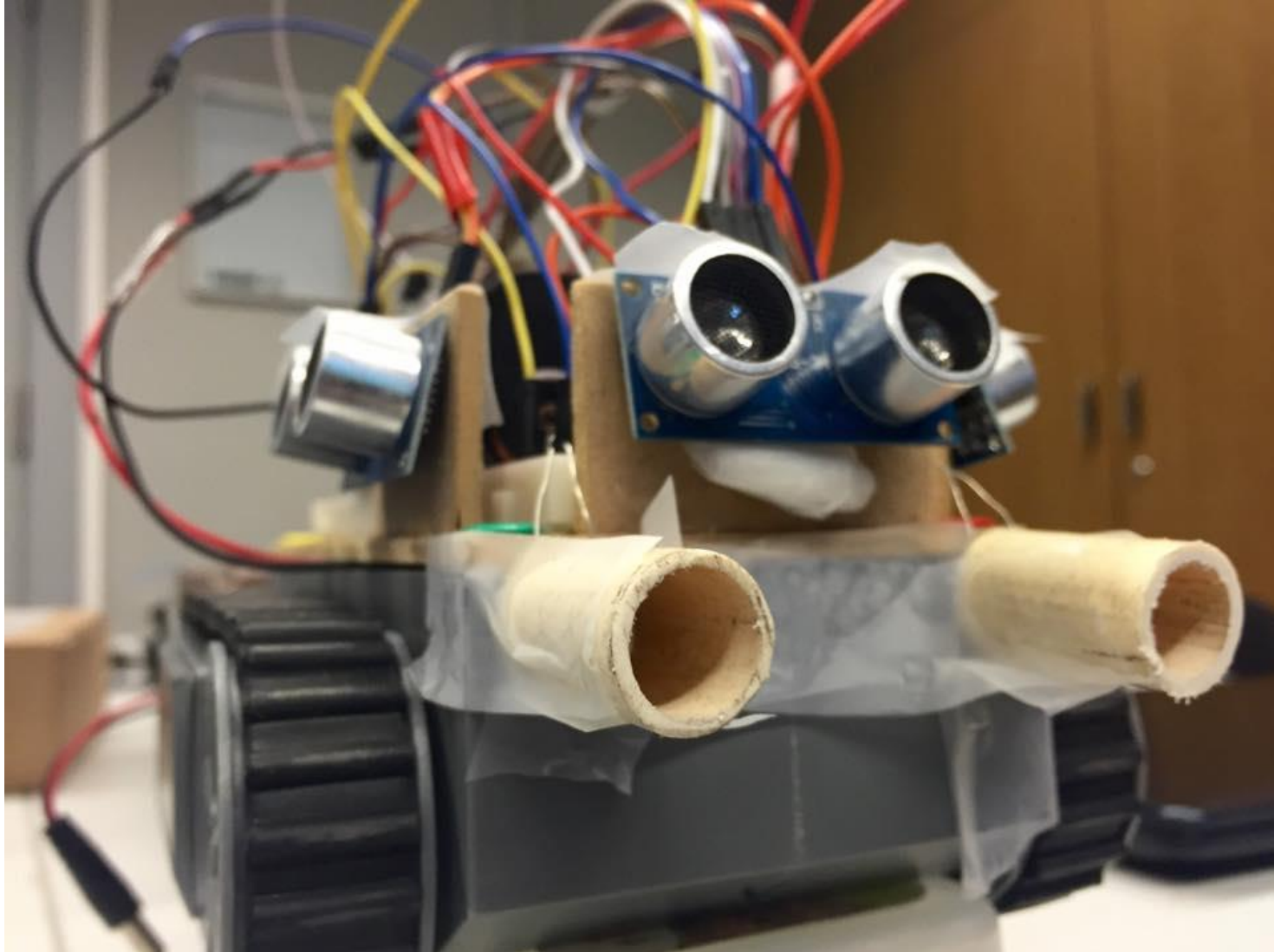


# Project elektronica 4

Lorenz Put: [Lorenz.Put@student.ap.be](mailto:Lorenz.Put@student.ap.be)

Michiel Mulder: [Michiel.Mulder@student.ap.be](mailto:Michiel.Mulder@student.ap.be)

# Wat we hebben bereikt



# Outline

- Inleiding
- Probleemstelling
- Probleemoplossing
  - Projectplan
  - Informatie vergaren en componenten verzamelen
  - Hardware
    - PCB
    - Plaatsing sensoren
  - Software
    - Uitlezen sensoren
    - Level 1
    - Level 2
    - Level 3
- Conclusie

# I. Inleiding

- Project 3 → Project 4
- Autonoom
- Level 1: Bekend parcours met hindernissen
- Level 2: Onbekend hindernissenparcours
- Level 3: Zoek object in open ruimte
- Level 4: Zoek object in open ruimte met hindernissen
- Level 5: Zoek object in onbekend hindernissenparcours met tijdslimiet 2

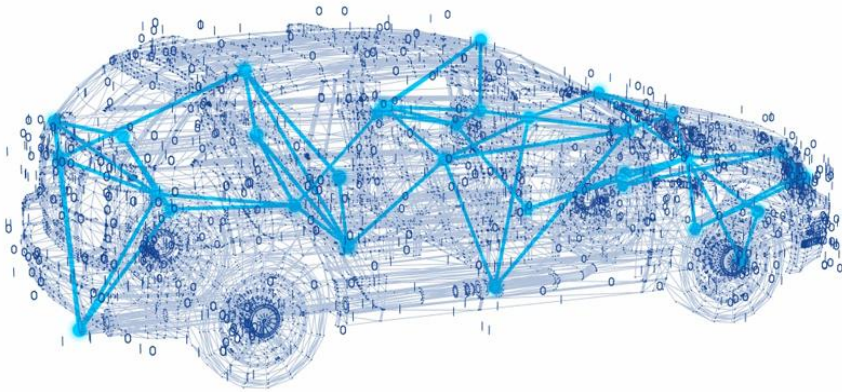
→ ***SENSOREN***

## 2. Probleemstelling

Auto

|

Sensoren

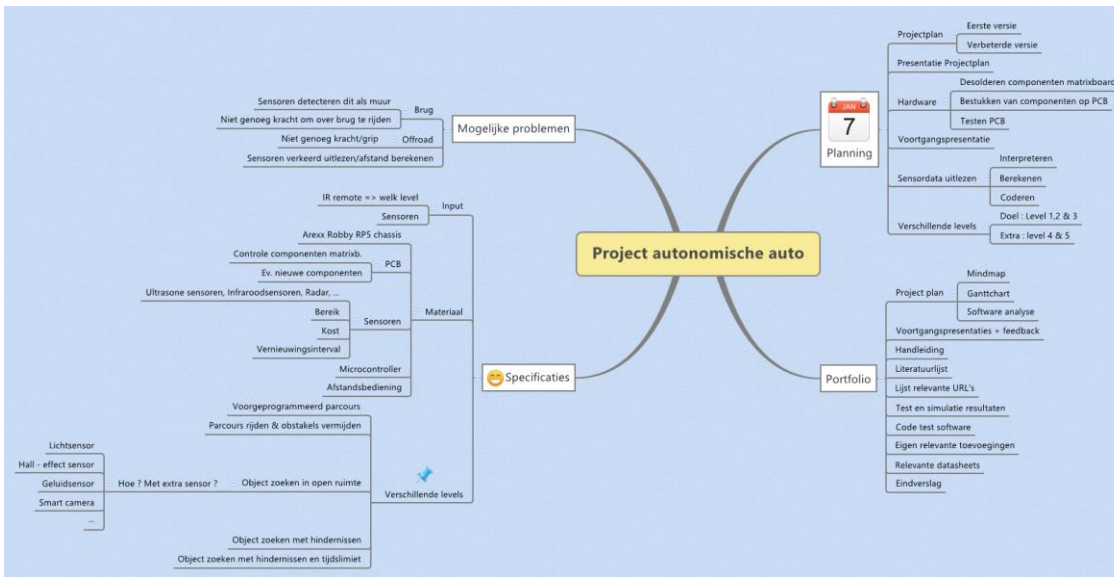


***“Hoe laten we onze auto  
autonoom zijn weg en/of objecten  
vinden in een vast of onbekend  
parcours/ruimte, zonder dat deze  
hinder ondervindt?”***

# 3. Probleemoplossing - outline

- Projectplan
- Informatie vergaren en componenten verzamelen
- Hardware
  - PCB
  - Plaatsing sensoren

# 3.1 Projectplan & Gantt- chart



## LEVEL 1

Variables:  
sFront = sensor voor  
sLeft = sensor links  
sRight = sensor rechts  
sBack = sensor achter  
corner = iterator

while (!sFront)

if (sFront)	
T	F
Go right	Go straight
End loop	

while (sFront != 'bridge-distance')

if (sFront == 'bridge-distance')	
T	F
Go straight	Go straight
End loop	

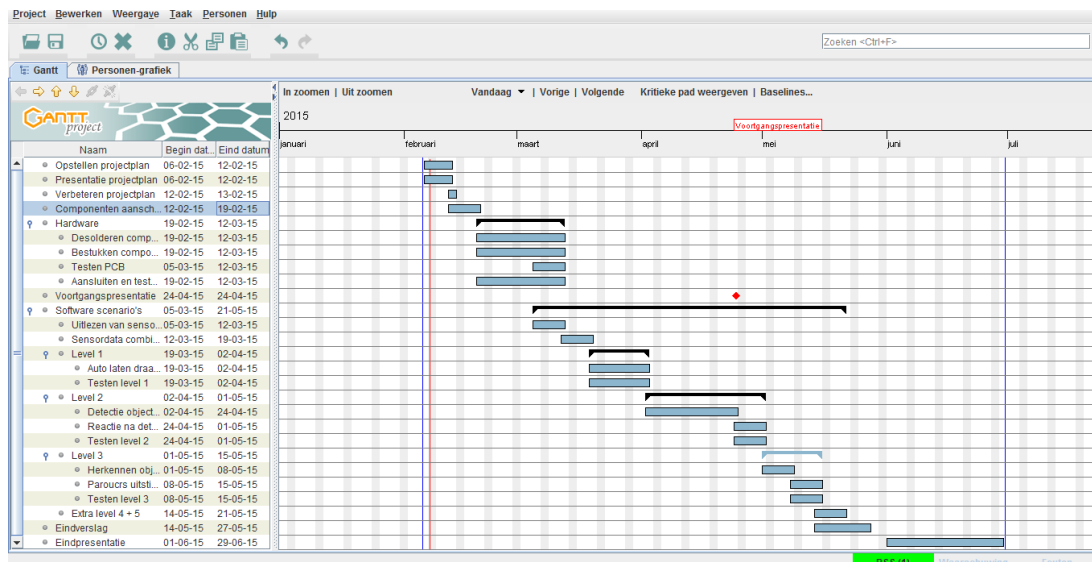
while (!sFront)

if (sFront)	
T	F
Go left	Go straight
End loop	
if (sFront)	
T	F
Go left	Go straight
End loop	

while (!sFront)

if (sFront)	
T	F
Go right	Go straight
End loop	

FINISH - end route





# 3.1 Overzicht taakverdeling

(Zie portfolio map '7. Eigen relevante toevoegingen')

- |                             |                 |
|-----------------------------|-----------------|
| • Projectplan en ganttchart | Beiden          |
| • Verbeteren projectplan    | Beiden          |
| • Hardware                  | Michiel         |
| • PCB                       | Michiel         |
| • Plaatsing sensoren        | Lorenz          |
| • Software scenario's       | Lorenz          |
| • Level 1                   | Lorenz          |
| • Level 2                   | Lorenz          |
| • Level 3                   | Michiel         |
| • Level 4, 5                | (niet aanwezig) |

## 3.2 Informatie vergaren + componenten verzamelen

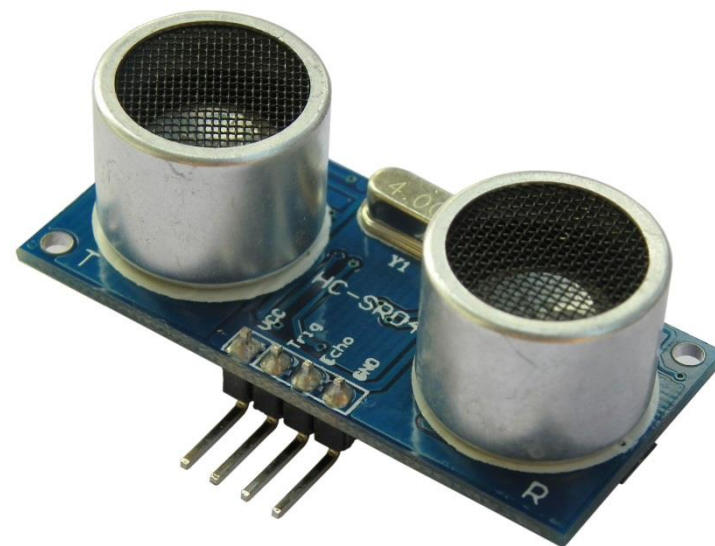


- Keuze tussen

Infrarood

of

Ultrasoon



## 3.2 Informatie vergaren + componenten verzamelen



- Keuze tussen

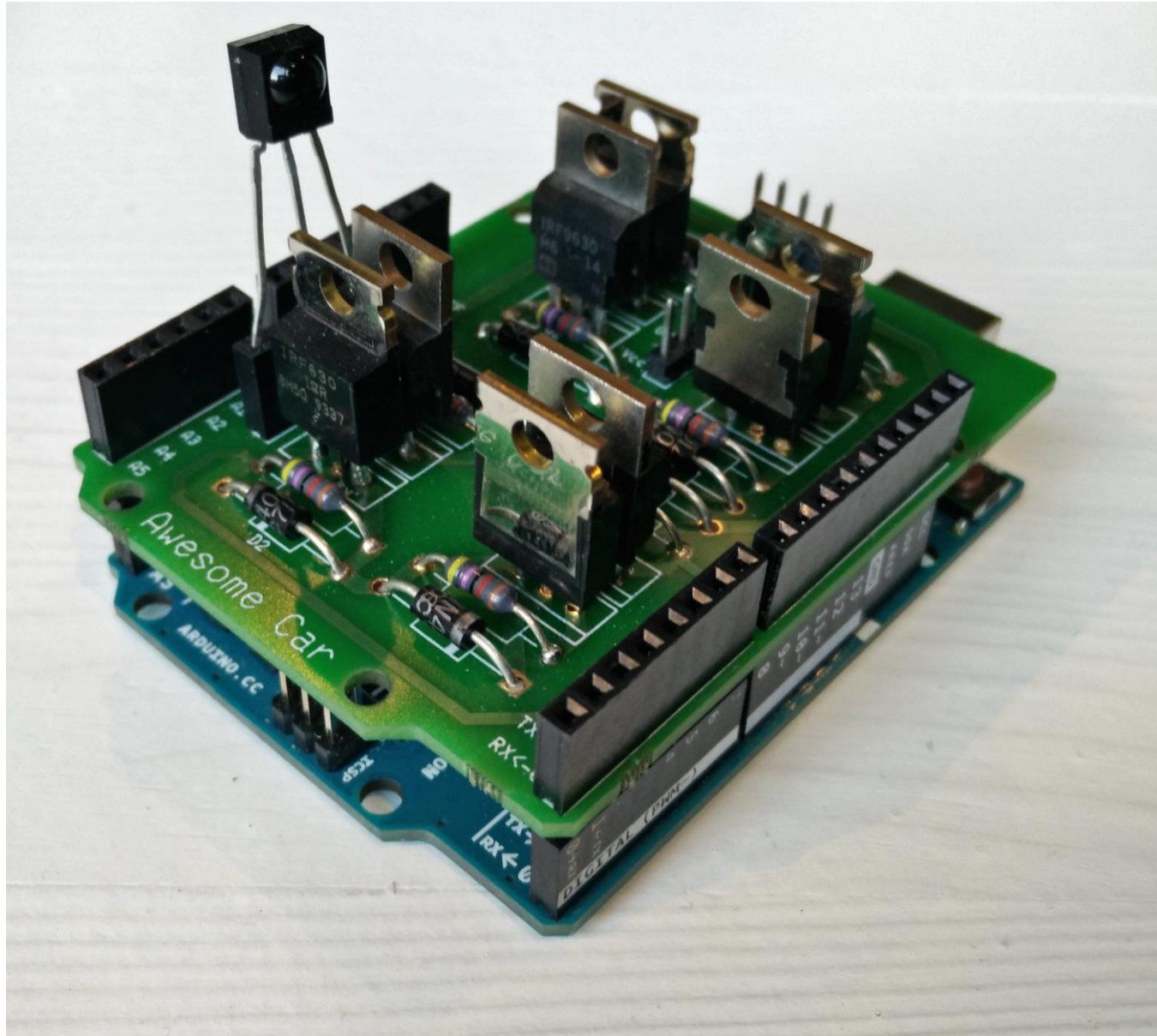
Hall - sensor

of

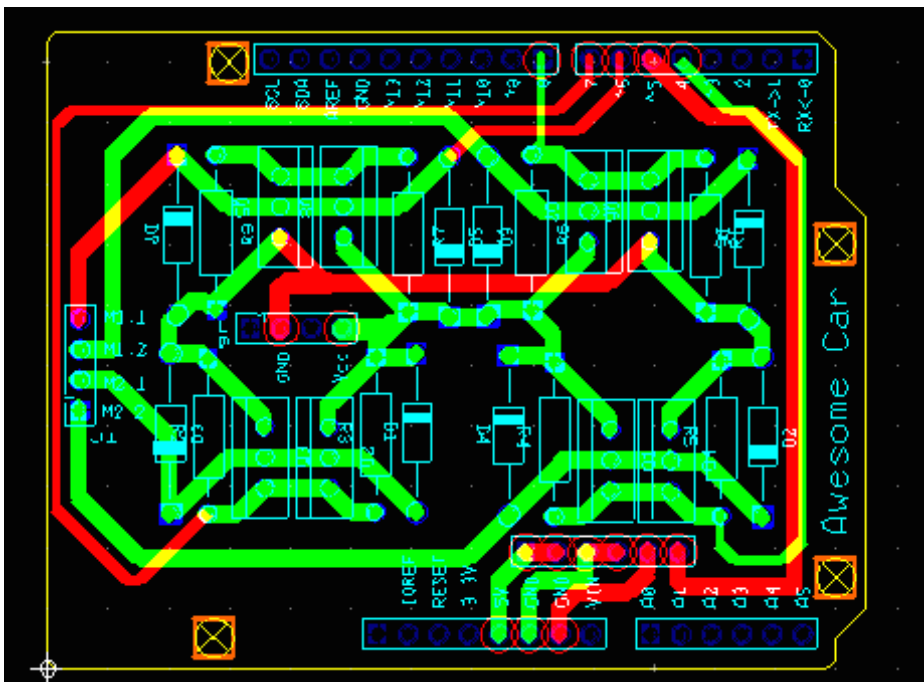
LDR - sensor



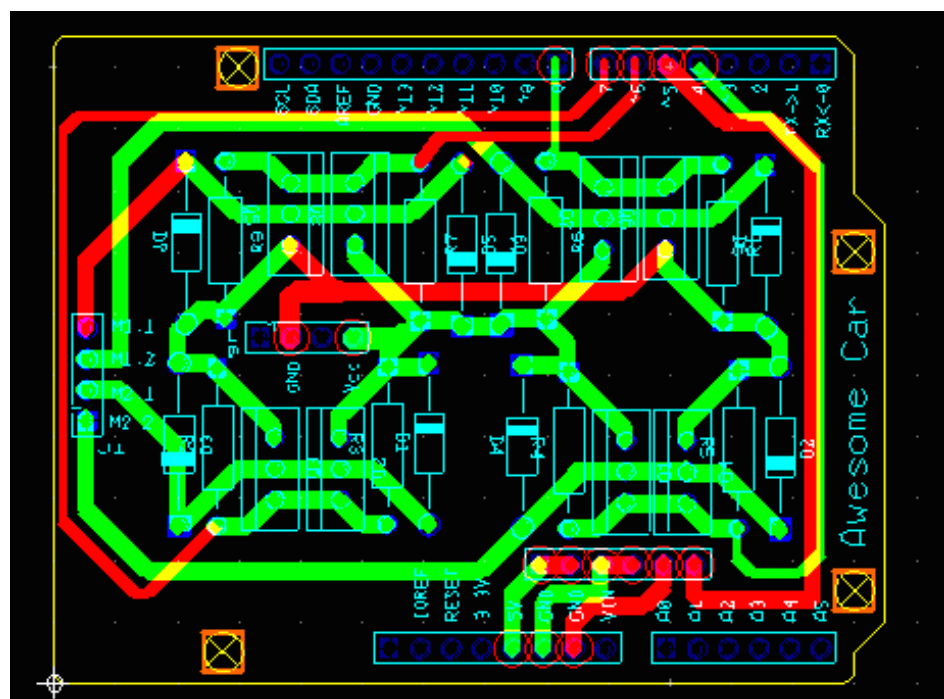
## 3.3 Hardware



Voor:



Na:

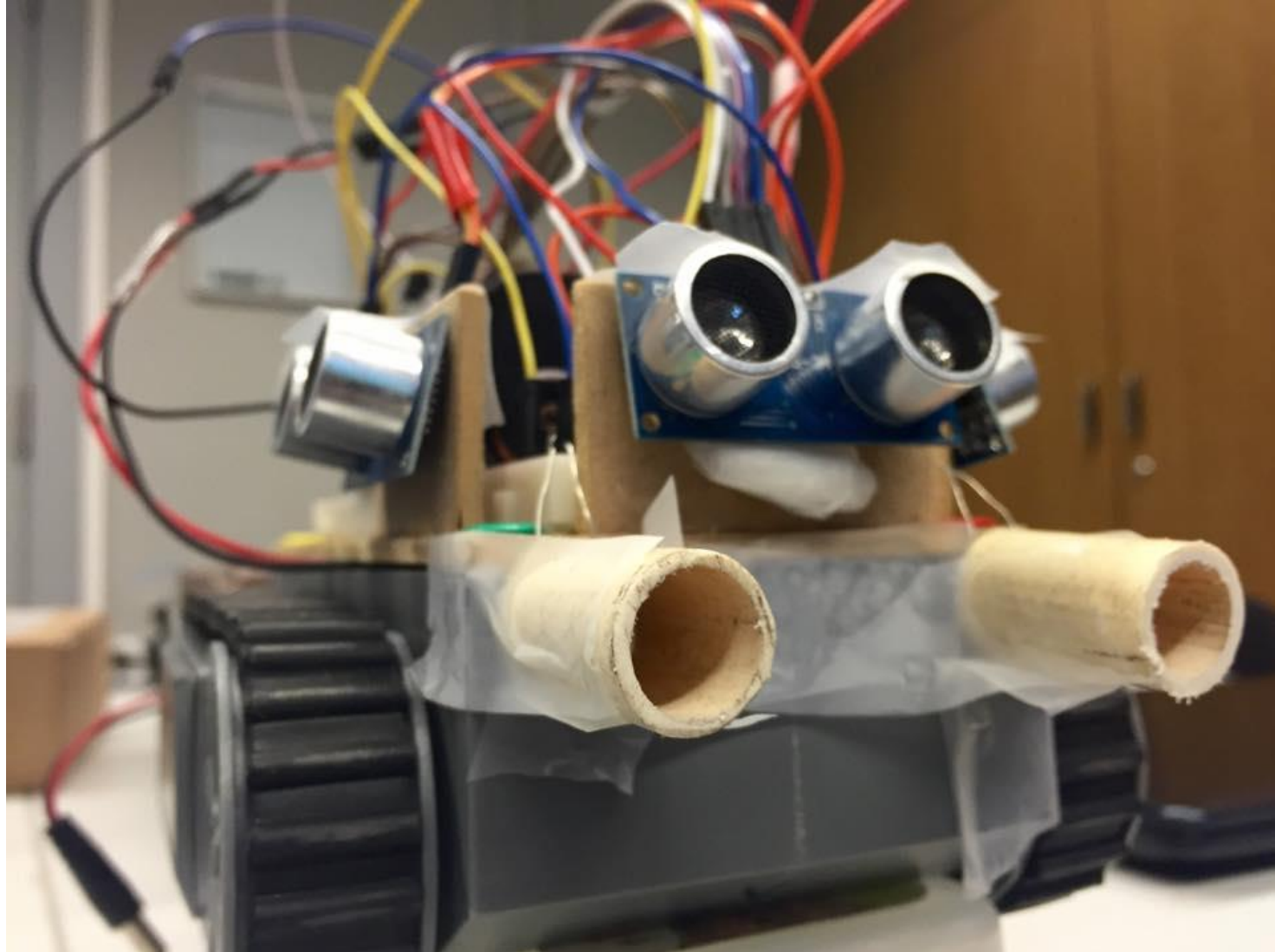




## 3.3.2 Plaatsing sensoren



## 3.3.2 Plaatsing sensoren





## 3.4 Software

## 3.4.1 Uitlezen sensoren

- Ultrasoonsensor uitlezen
- Black box principe

```

1  #include <Ultrasonsensor.h>
2
3  Ultrasonsensor s(A2,A3);
4  int afstand = 0;
5  void setup()
6  {
7      Serial.begin(9600);
8  }
9  void loop()
10 {
11     afstand = s.Measure();
12     Serial.println(afstand);
13
14 }
```

```

1  #include "Arduino.h"
2  #include "Ultrasonsensor.h"
3
4  Ultrasonsensor::Ultrasonsensor(int trigPin, int echoPin)
5  {
6      pinMode(trigPin,OUTPUT);
7      pinMode(echoPin,INPUT);
8
9
10     _trigPin = trigPin;
11     _echoPin = echoPin;
12 }
13 long Ultrasonsensor::Measure()
14 {
15     digitalWrite(_trigPin, LOW);
16     delayMicroseconds(2);
17     digitalWrite(_trigPin, HIGH);
18
19     delayMicroseconds(10);
20     digitalWrite(_trigPin, LOW);
21     _duration = pulseIn(_echoPin, HIGH);
22     _distance = (_duration/2) / 29.1;
23     return _distance;
24 }
```

## 3.4.2. Verschillende levels

```

47  if(irrecv.decode(&results))
48  {
49      Serial.println(results.value, DEC
50
51      if(results.value == 16738455)
52      {
53          level1 = true;
54          level2 = false;
55          carStop = false;
56      }
57      if(results.value == 16750695)
58      {
59          level1 = false;
60          level2 = true;
61          carStop = false;
62      }
63
64      if(results.value == 16712445)
65      {
66          carStop = true;
67      }
68
69
70      irrecv.resume();
    }

```

- Starten en stoppen levels a.d.h.v. IR remote
- Booleans houden level aan

```

72  if(!carStop)
73  {
74      if(level1)
75      {
76          Level1();
77      }
78      else if (level2)
79      {
80          Level2();
81      }
82  }
83  else
84  {
85      {
86          m.Stop();
87      }
88

```

## 3.4.2 Level I

- Programmatorisch parcours rijden
- Gebruik van delays

```
19 void loop() {
20     // put your main code here, to run repeatedly
21     if(irrecv.decode(&results))
22     {
23         Serial.println(results.value, DEC);
24         if(results.value == 16736925)
25         {
26             Levell();
27         }
28         irrecv.resume();
29     }
30
31 }
32
33 void Levell ()
34 {
35     m.TurnLeft(590);
36     Serial.println("Ik draai");
37     m.Stop();
38     Serial.println("Ik stop");
39     m.TurnLeft(590);
40     m.Stop();
41     m.Forward(1000);
42     m.Stop();
43     m.TurnRight(590);
44     m.Stop();
45     m.Forward(1000);
46 }
```

## 3.4.3 Level 2

- Stuur algoritme op basis van sensordata

```

171  if(forwarddistance <= distanceF)
172  {
173      m.Stop();
174      if(leftdistance > distanceS || rightdistance > distanceS)
175      {
176
177          if(leftdistance > rightdistance)
178          {
179
180              m.TurnLeft(draaitijd);
181              Printonscreen("Ik draai 90 graden naar links");
182          }
183
184
185
186      else if(rightdistance > leftdistance)
187      {
188          m.TurnRight(draaitijd);
189          Printonscreen("Ik draai 90 graden naar rechts");
190      }
191
192
193
194      else if (rightdistance == leftdistance)
195      {
196          m.TurnLeft(draaitijd);
197          Printonscreen("Ik draai 90 graden naar links");
198      }

```

```

201      else
202      {
203          m.Stop();
204          digitalWrite(13, HIGH);
205          Printonscreen("Ik zit vast");
206      }
207  }
208
209
210  else if(leftdistance < distanceS)
211  {
212      m.TurnRight(draaitijd);
213      Printonscreen("Ik stuur bij naar rechts");
214  }
215  else if(rightdistance < distanceS)
216  {
217      m.TurnLeft(draaitijd);
218      Printonscreen("Ik stuur bij naar links");
219  }
220
221  else
222  {
223      m.Forward(draaitijd);
224      Printonscreen("Ik rijd vooruit");
225  }
226  }
227

```

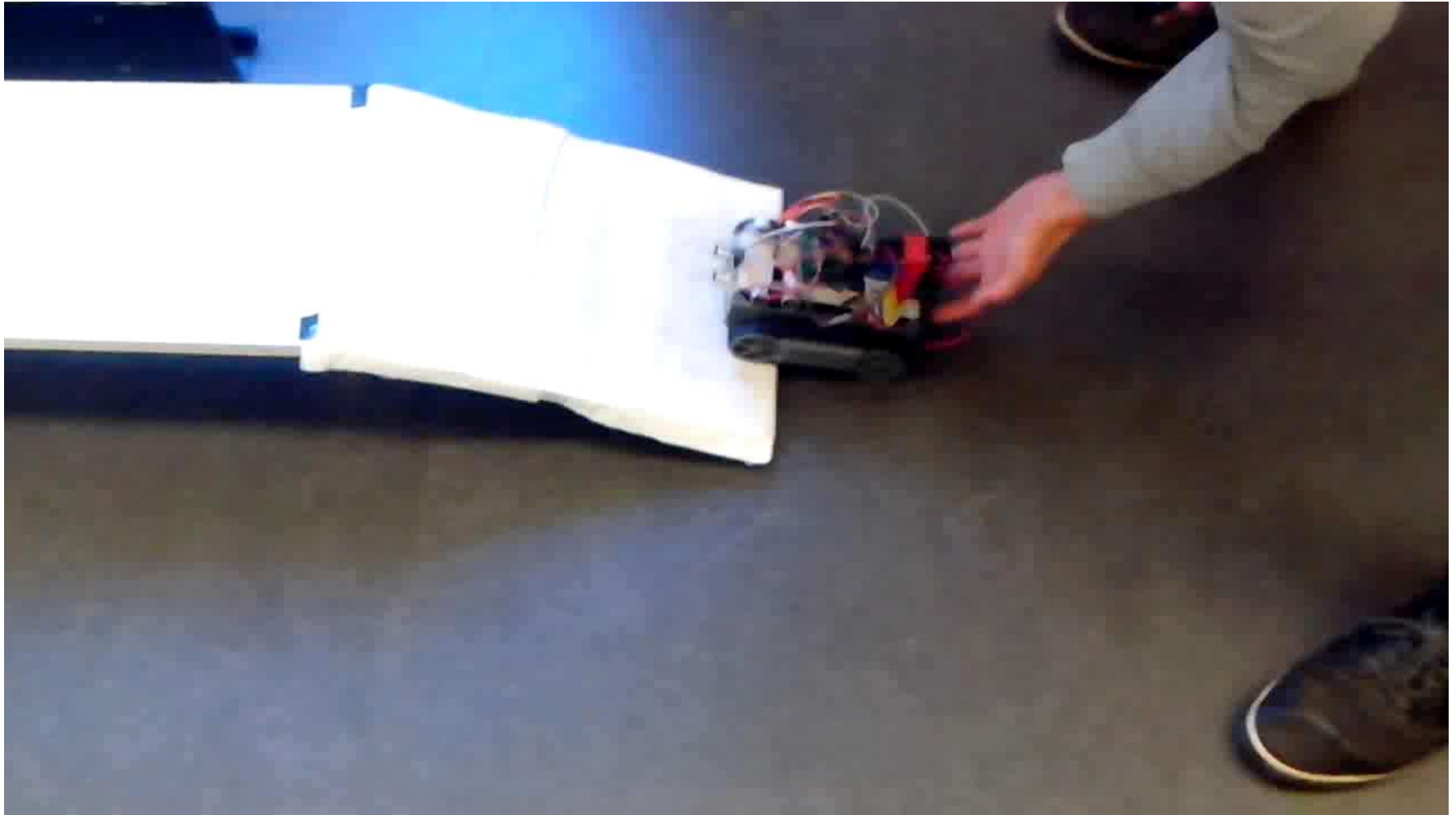
## 3.4.4 Level 3

- 2 Light Dependant Resistors of LDR's
- Links meer lichtinval → naar links draaien
- Zelfde voor rechts
- Lichtinval groter dan vb. 500 = gevonden!

## 4. Tegengekomen problemen

- Gewichtverdeling
- Kapotte batterijhouder
- PCB (reeds vermeld)
- Onbekend probleem voor de spanning

# Demo - Brug





# Vragen?

**Bedankt voor u aandacht !**