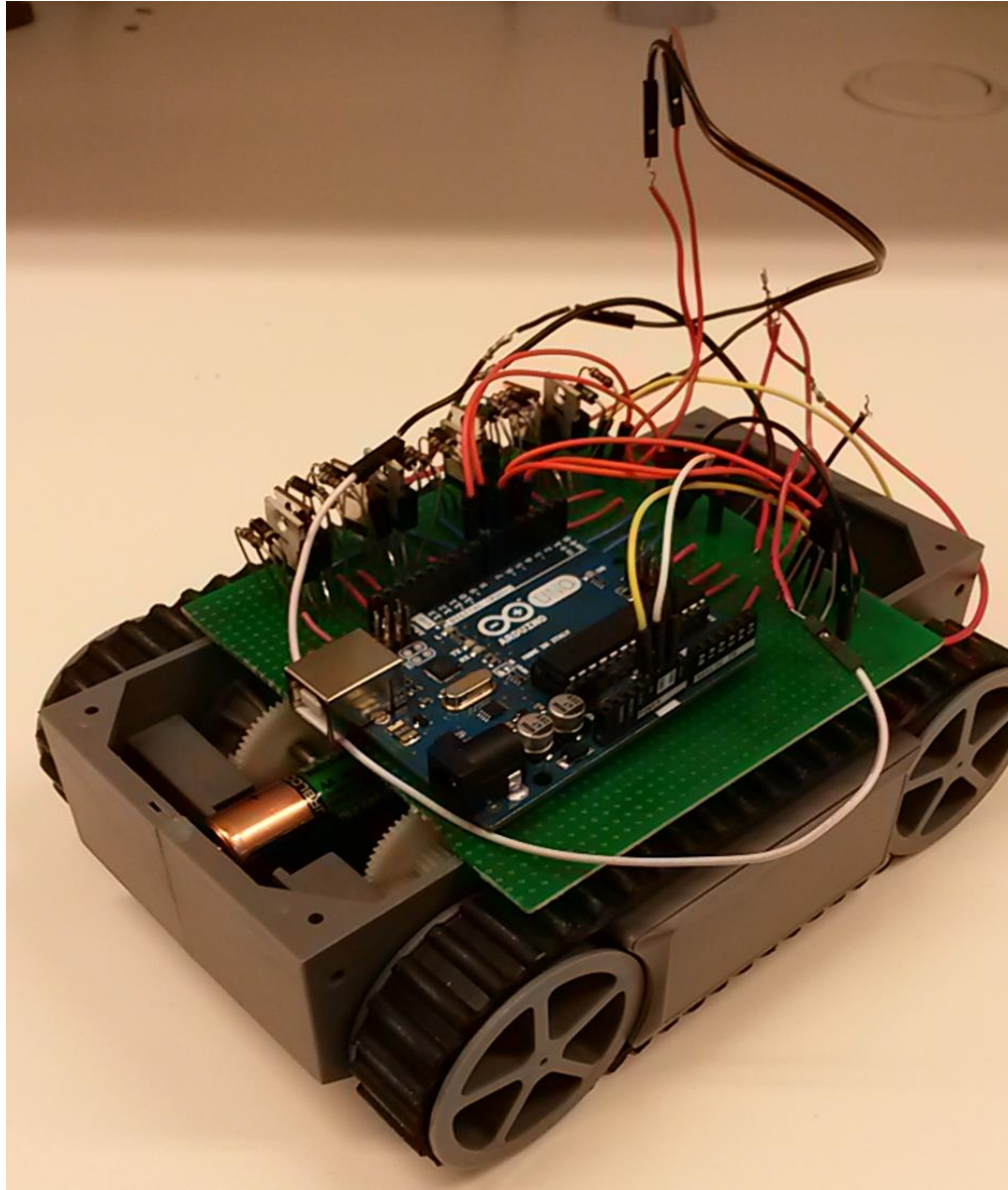


# Project elektronica 3

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

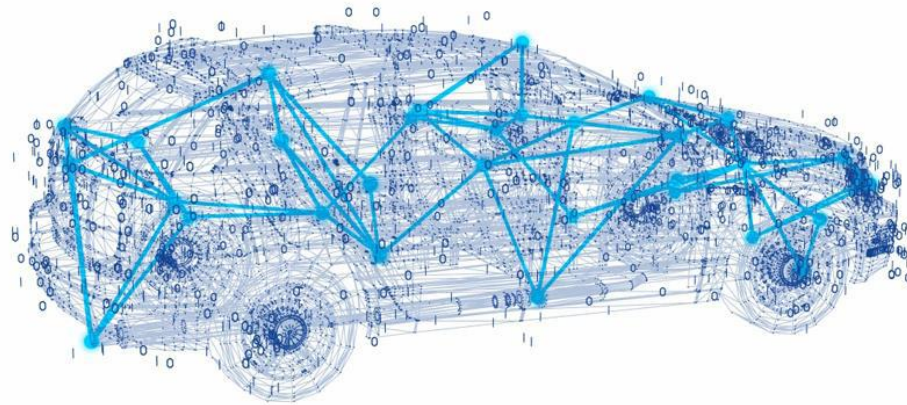
- Inleiding
- Probleemstelling
- Probleemoplossing
  - Projectplan
  - Informatie & componenten
  - Schema ontwerpen
  - Testschakeling breadboard
  - Opstellen code
  - Ontwerpen PCB
- Conclusie

# I. Inleiding

- Project Elektronica 3
- Auto
  - Remote control
  - Vooruit en achteruit
  - Links en rechts
  - Stoppen
  - (Brug)

# 2.Probleemstelling

- Auto



Project

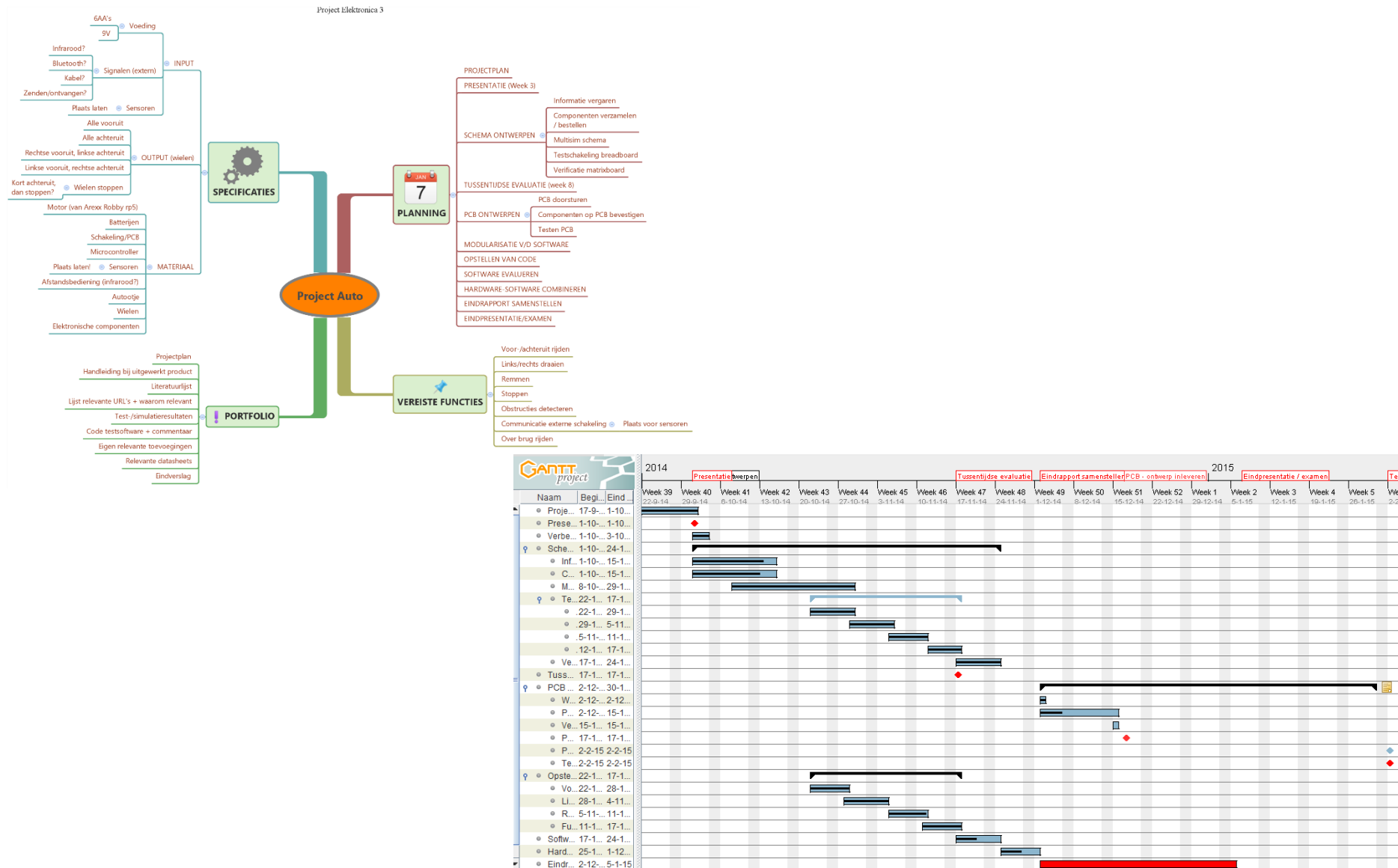


*“Hoe vervolledigen we een project dat erin bestaat een auto geautomatiseerd te laten functioneren?”*

# 3. Probleemoplossing - outline

- Projectplan
- Informatie & componenten
- Schema ontwerpen
- Testschakeling breadbord
- Opstellen code
- Ontwerpen PCB

# 3.1 Projectplan & Gantt- chart





# Overzicht verdeling

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

- |                                      |         |
|--------------------------------------|---------|
| • Projectplan                        | Beide   |
| • Info & componenten                 | Beide   |
| • Multisim schema                    | Beide   |
| • Testschakeling breadbord           | Beide   |
| • H-bruggen                          | Michiel |
| • IR-schakeling + combineren         | Lorenz  |
| • Matrixboard                        | Michiel |
| • PCB                                | Michiel |
| • Opstellen code                     | Lorenz  |
| + Functies & bibliotheken integreren | Lorenz  |

## 3.2 Informatie vergaren + componenten verzamelen

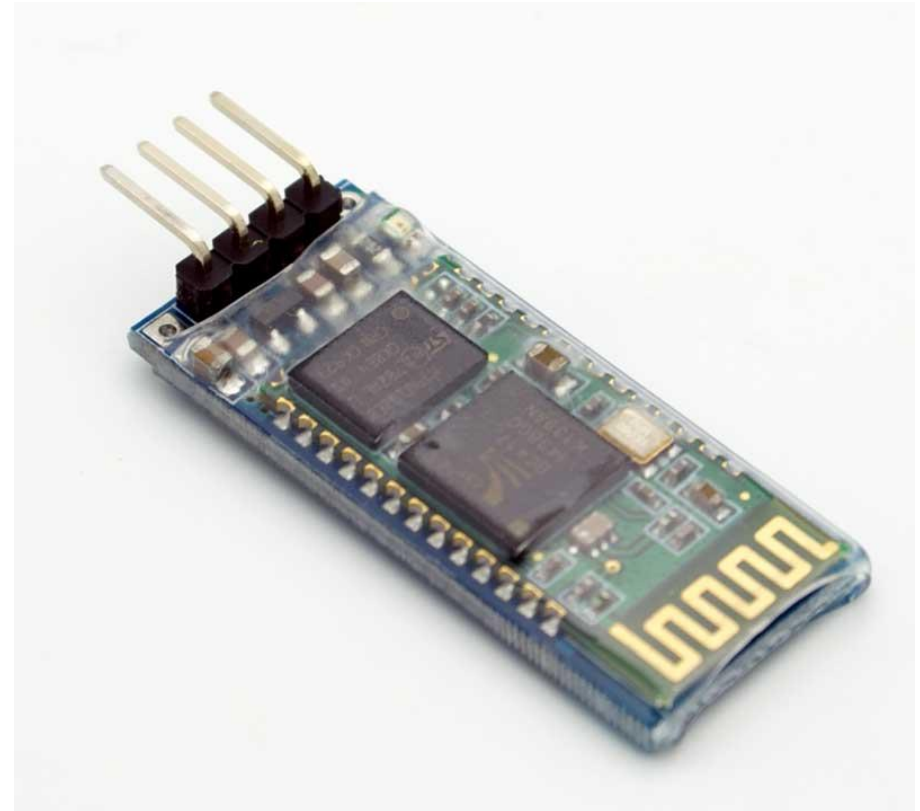
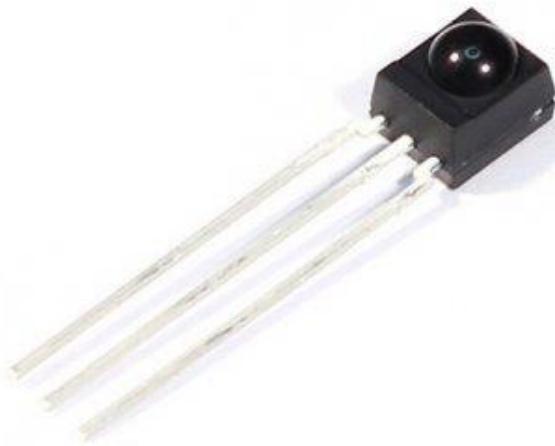


- Keuze tussen

Infrarood

of

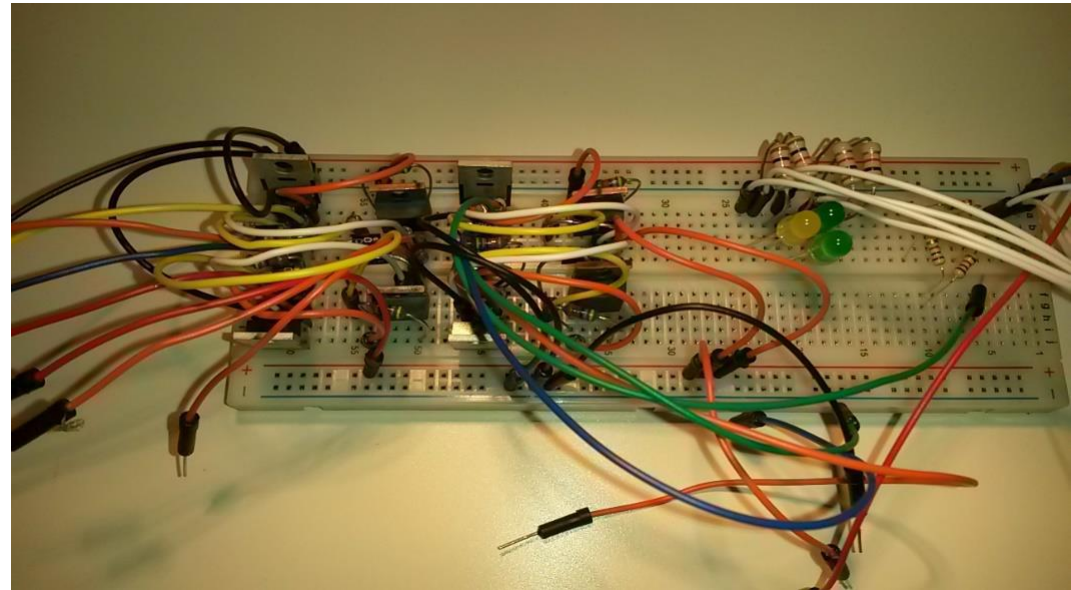
Bluetooth



## 3.2 Informatie vergaren + componenten verzamelen

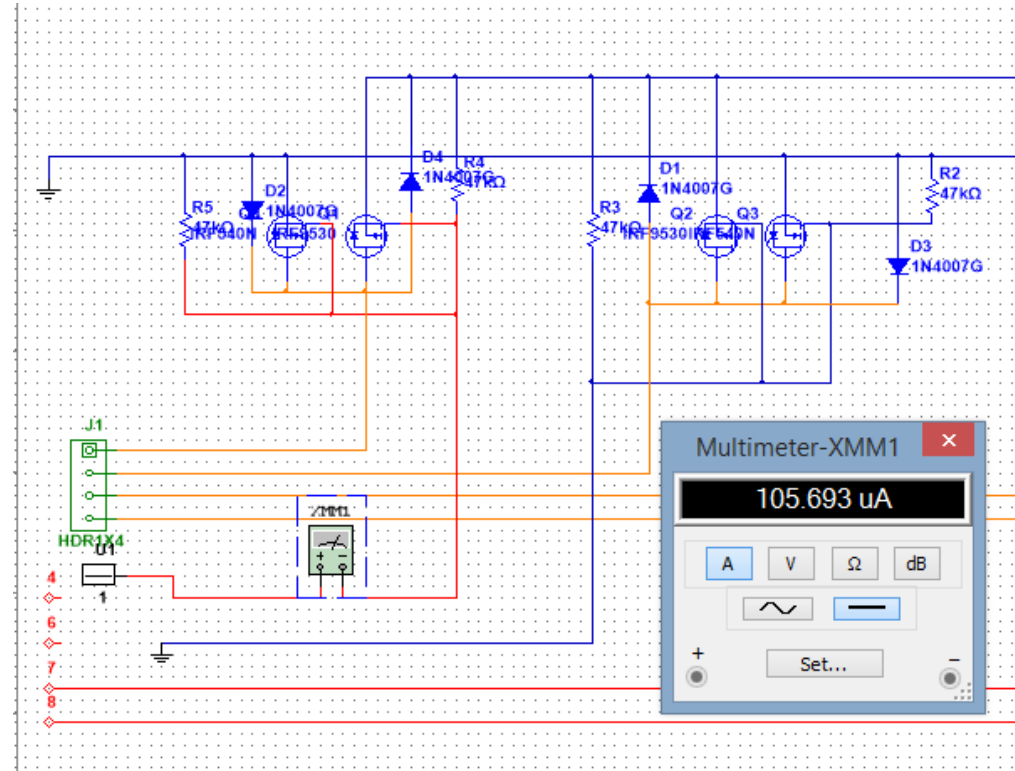
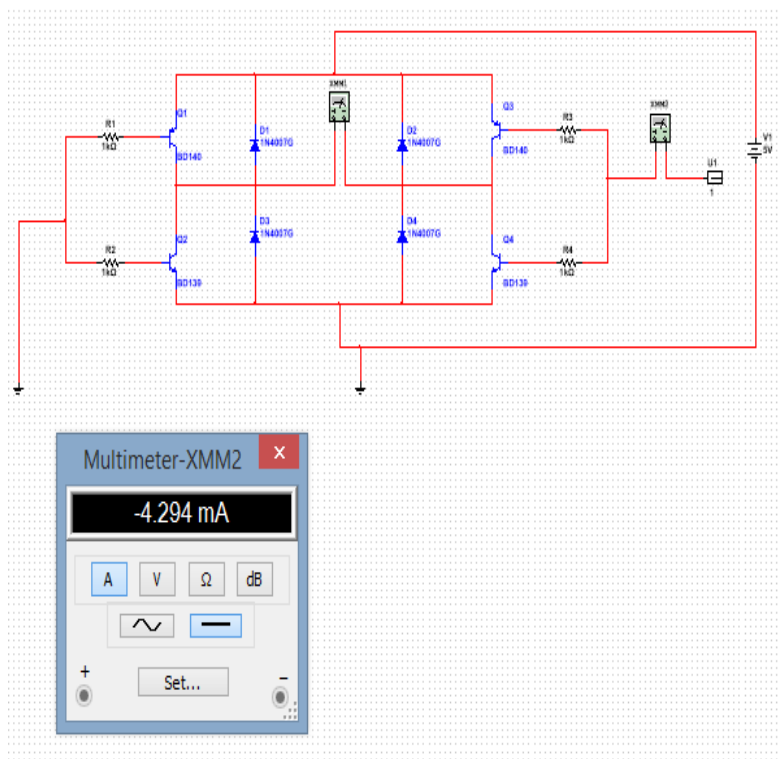


- Keuze tussen  
IC of Zelfgebouwde H-bruggen

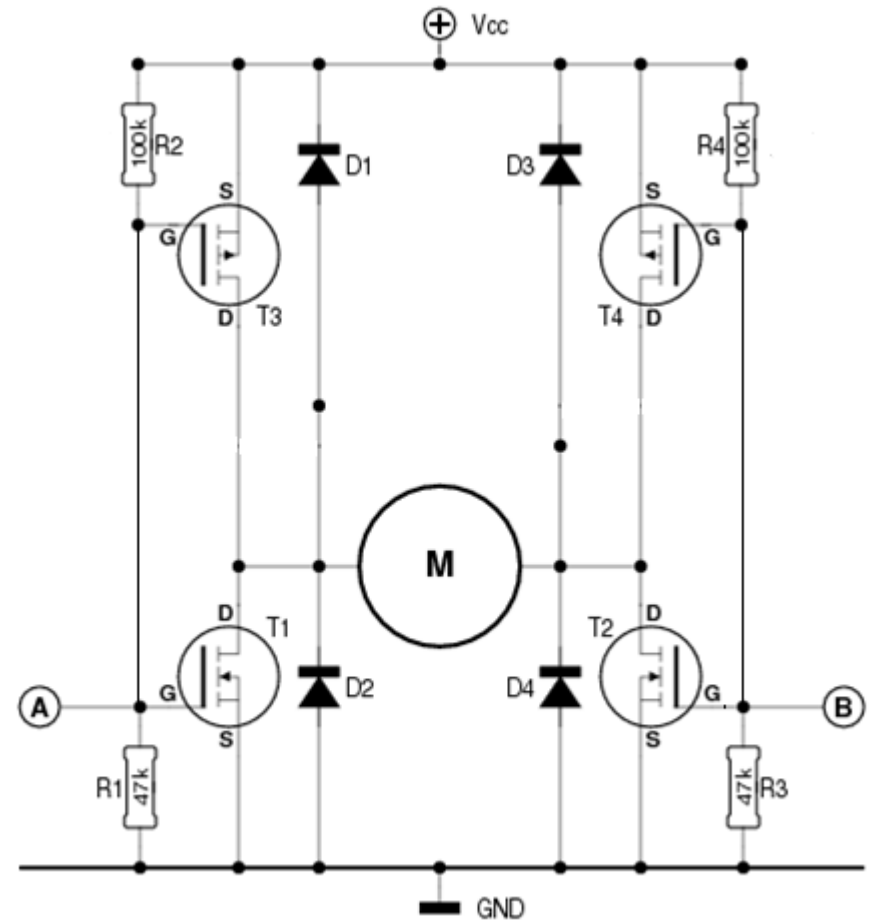


# 3.3 Schema ontwerpen

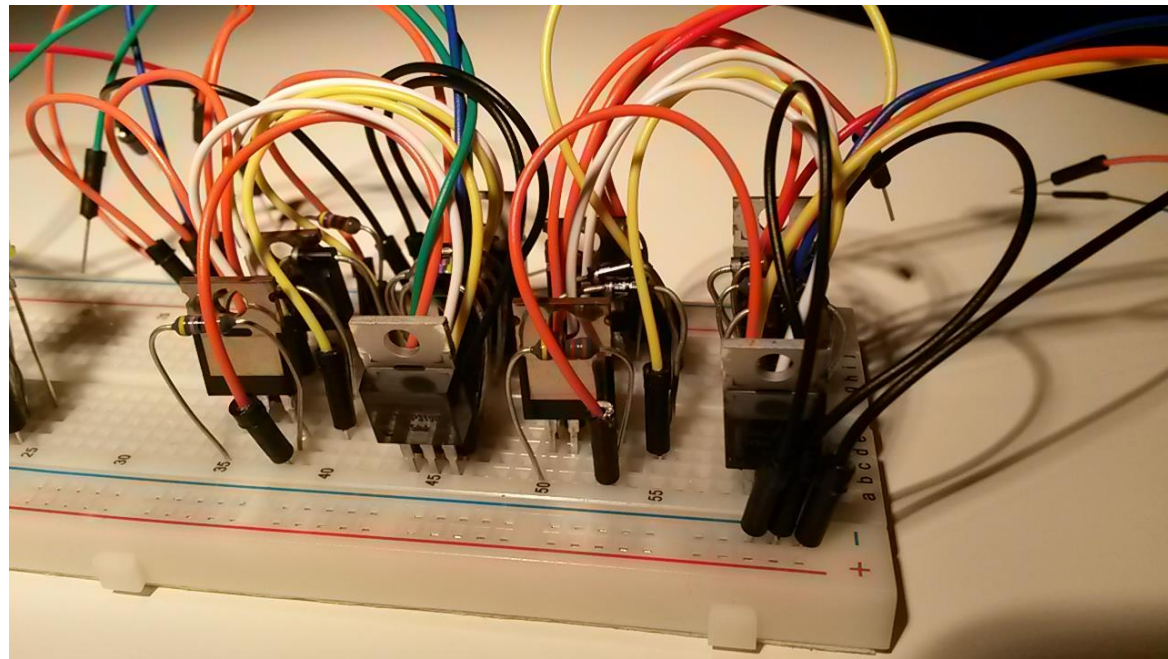
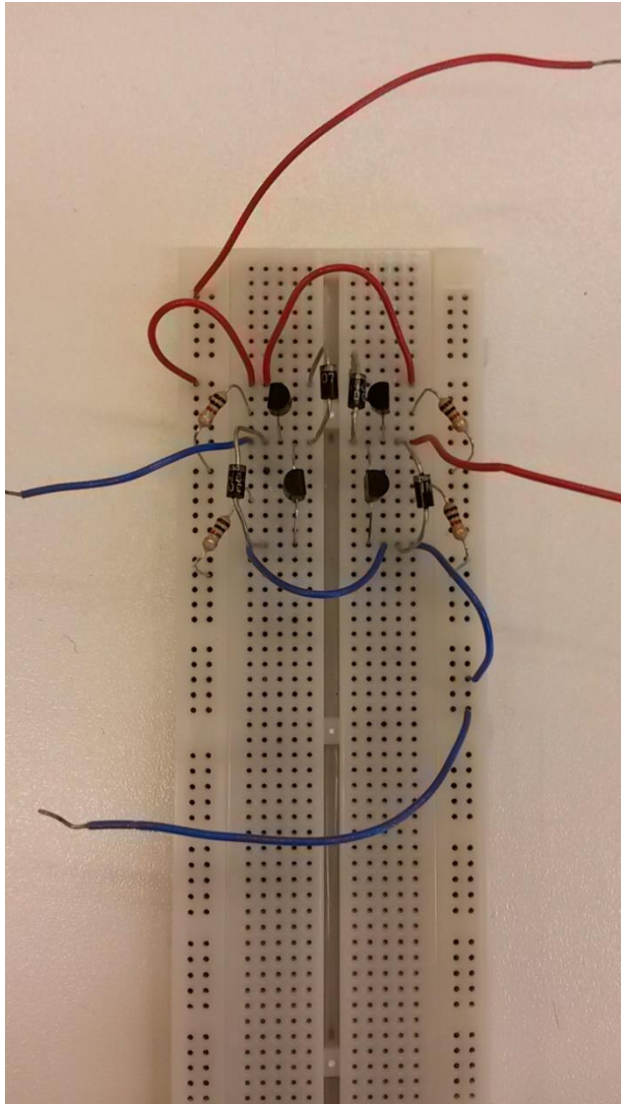
- Transistoren of MOSFET's



**Bewerkt**



## 3.4 Testschakeling breadbord





## 3.5 Opstellen code – Test H-brug(gen)

### MotoraansturingTest\$

/\*Dit programma moet de motor vooruit & achteruit laten rijden\*/

```
const int leftOne = 2;  
const int leftTwo = 3;
```

```
void setup()  
{  
  Serial.begin(9600);  
  pinMode(leftOne, OUTPUT);  
  pinMode(leftTwo, OUTPUT);  
}
```

//Dit programma zet 2 pinnen op hoog of op laag.

//Dit werd gebruikt om de H-bruggen te testen.

```
void loop()  
{  
  digitalWrite(leftOne, LOW);  
  digitalWrite(leftTwo, HIGH);  
}
```

### MotoraansturingTest\$

/\*Dit programma moet de motor vooruit & achteruit laten rijden\*/

```
const int leftOne = 2;  
const int leftTwo = 3;  
const int rightOne = 4;  
const int rightTwo = 5;
```

```
void setup()  
{  
  Serial.begin(9600);  
  pinMode(leftOne, OUTPUT);  
  pinMode(leftTwo, OUTPUT);  
  pinMode(rightOne, OUTPUT);  
  pinMode(rightTwo, OUTPUT);  
}
```

//Dit programma zet 4 pinnen op hoog of op laag.

//Dit werd gebruikt om de H-bruggen te testen.

```
void loop()  
{  
  digitalWrite(leftOne, LOW);  
  digitalWrite(leftTwo, HIGH);  
  digitalWrite(rightOne, LOW);  
  digitalWrite(rightTwo, HIGH);  
}
```

# 3.5 Opstellen code - Toetsenbord

- Auto besturen a.d.h.v. Input toetsenbord

Motoraansturing\_InputPcFull\$

```
void loop()
{
    //A.d.h.v. de letter die hier ingelezen wordt uit
    //de seriële monitor zal het programma een bepaalde functie uitvoeren.
    Inputkey = Serial.read();

    if(Inputkey == 'f')
    {
        Forward();
    }
    if(Inputkey == 's')
    {
        Stop();
    }
    if(Inputkey == 'l')
    {
        TurnLeft();
    }
    if(Inputkey == 'r')
    {
        TurnRight();
    }
    if(Inputkey == 'b')
    {
        Backward();
    }
}
```

//Dit zijn de verschillende methoden die in de loop aangeroepen worden

```
void Forward()
{
    digitalWrite(leftOne, HIGH);
    digitalWrite(leftTwo, LOW);
    digitalWrite(rightOne, LOW);
    digitalWrite(rightTwo, HIGH);
}

void Backward()
{
    digitalWrite(leftOne, LOW);
    digitalWrite(leftTwo, HIGH);
    digitalWrite(rightOne, HIGH);
    digitalWrite(rightTwo, LOW);
}

void TurnLeft()
{
    digitalWrite(leftOne, HIGH);
    digitalWrite(leftTwo, LOW);
    digitalWrite(rightOne, HIGH);
    digitalWrite(rightTwo, LOW);
}

void TurnRight()
{
    digitalWrite(leftOne, LOW);
    digitalWrite(leftTwo, HIGH);
    digitalWrite(rightOne, LOW);
    digitalWrite(rightTwo, HIGH);
}

void Stop()
{
    digitalWrite(leftOne, HIGH);
    digitalWrite(leftTwo, HIGH);
    digitalWrite(rightOne, HIGH);
    digitalWrite(rightTwo, HIGH);
}
```



## 3.5 Opstellen code – IR remote

- Uitlezen van de specifieke codes van de IR remote

### IRReceiverTest

```
Pin 1 naar Vout (pin 11 van Arduino)
Pin 2 naar GND
Pin 3 naar Vcc (+5v van Arduino)

*/

#include <IRremote.h>

int IRpin = 11;
IRrecv irrecv(IRpin);
decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}

void loop()
{
  if (irrecv.decode(&results))
  {
    Serial.println(results.value, DEC); // Print the Serial 'results.value'
    irrecv.resume(); // Receive the next value
  }
}
```

## 3.5 Opstellen code – IR remote

- Implementeren van IR-codes

```
void loop()
{
  if(irrecv.decode(&results))
  {
    //In plaat van dat er hier een letter binnen komt van de seriële monitor komt er een rijfer binnen.
    //Dit rijfer wordt dan vergeleken met waarden die via een ander programma bekomen zijn.
    //Indien de waarde gelijk zijn, wordt er een bepaalde functie uitgevoerd.
    //Deze functies staan onderaan het programma gedefinieerd.
    Serial.println(results.value, DEC);
    if(results.value == 16736925)
    {
      Forward();
    }
    if(results.value == 16712445)
    {
      Stop();
    }
    if(results.value == 16720605)
    {
      TurnRight();
    }
    if(results.value == 16761405)
    {
      TurnLeft();
    }
    if(results.value == 16754775)
    {
      Backwards();
    }

    irrecv.resume();
  }
}
```

# 3.5 Opstellen code - bibliotheek

- Methoden omzetten naar bibliotheek

Motor.h

```

#ifndef Motor_h
#define Motor_h

#include "Arduino.h"
/*Dit bestand geeft een template weer van welke variabelen de cpp file zal bevatten */
class Motor
{
public:
    // De constructor
    Motor(int leftOne, int leftTwo, int rightOne, int rightTwo);
    //de functie forward, turnleft, turnright, achterruit en stop.
    void Forward();
    void TurnLeft();
    void TurnRight();
    void BackWards();
    void Stop();
private:
    // de private variabelen _leftOne _leftTwo, _rightOne, _rightTwo;
    int _leftOne;
    int _leftTwo;
    int _rightOne;
    int _rightTwo;
};
#endif

```

Motor.cpp

```

#include "Arduino.h"
#include "Motor.h"
/*De constructor van de motor.
Hier worden de pinnen gedefinieerd*/
Motor::Motor(int leftOne, int leftTwo, int rightOne, int rightTwo)
{
    pinMode(leftOne,OUTPUT);
    pinMode(leftTwo,OUTPUT);
    pinMode(rightOne,OUTPUT);
    pinMode(rightTwo,OUTPUT);
    /*de pinnummers worden door gegeven aan private variabelen*/
    _leftOne = leftOne;
    _leftTwo = leftTwo;
    _rightOne = rightOne;
    _rightTwo = rightTwo;
}
/*de pinnen worden zo aan gestuurd dat de auto voorruit rijdt.*/
void Motor::Forward()
{
    digitalWrite(_leftOne, HIGH);
    digitalWrite(_leftTwo, LOW);
    digitalWrite(_rightOne, LOW);
    digitalWrite(_rightTwo, HIGH);
}
/*de pinnen worden zo aan gestuurd dat de auto naar links rijdt.*/
void Motor::TurnLeft()
{
    digitalWrite(_leftOne, HIGH);
    digitalWrite(_leftTwo, LOW);
    digitalWrite(_rightOne, HIGH);
    digitalWrite(_rightTwo, LOW);
}

```

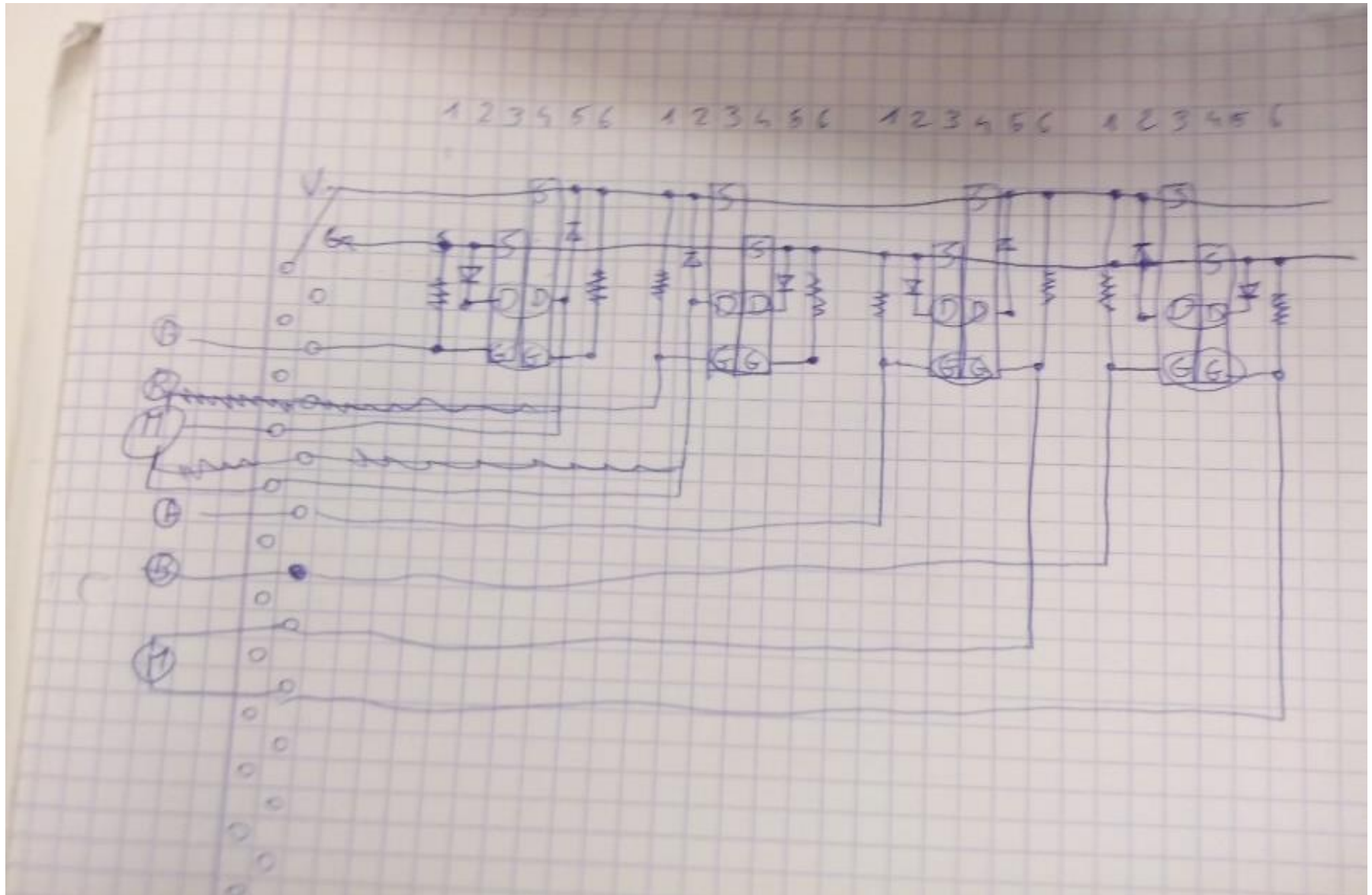
## 3.5 Opstellen code - bibliotheek

- Top – level
  - Black box
- Gemakkelijk te combineren met sensoren

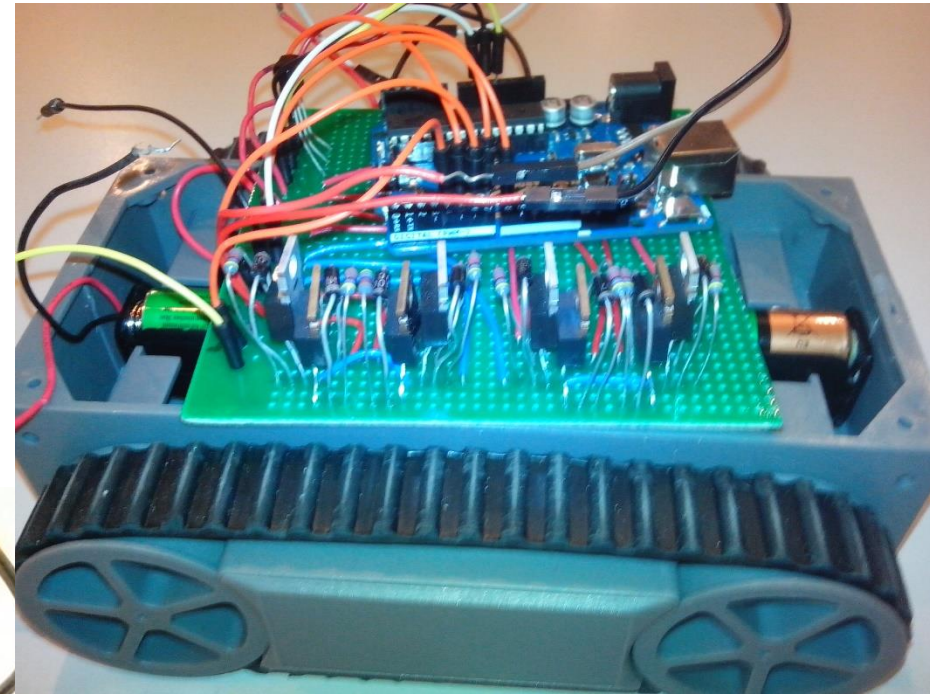
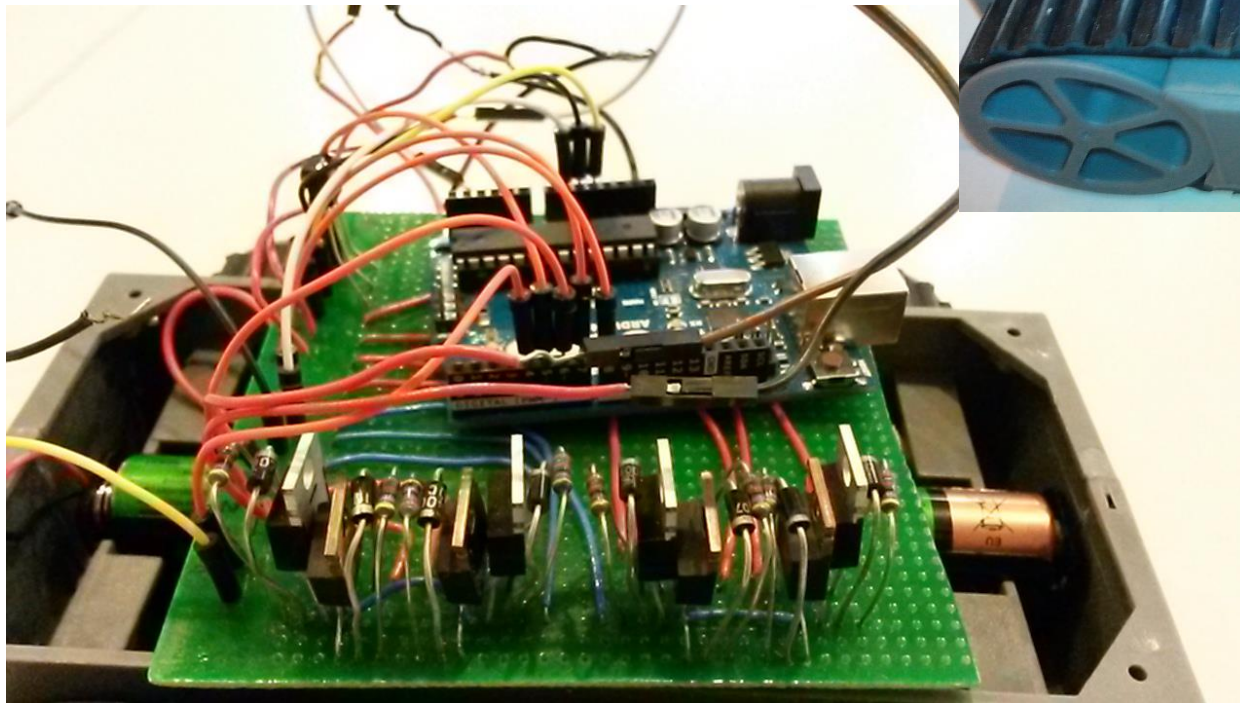
Definitieve\_versie \$

```
void loop()
{
  /*Afhankelijk van het signaal dat de IR ontvanger binnen krijgt,
  zal de if structuur er voor zorgen dat een bepaalde functie
  uit de motor bibliotheek aangeroepen zal worden.
  */
  if(irrecv.decode(&results))
  {
    Serial.println(results.value, DEC);
    if(results.value == 16736925)
    {
      motor.Forward();
    }
    if(results.value == 16712445)
    {
      motor.Stop();
    }
    if(results.value == 16720605)
    {
      motor.TurnRight();
    }
    if(results.value == 16761405)
    {
      motor.TurnLeft();
    }
    if(results.value == 16754775)
    {
      motor.BackWards();
    }
    //Er wordt gekeken of de IR receiver een nieuwe waarde heeft ontvangen.
    irrecv.resume();
  }
}
```

## 3.6 Matrixboard

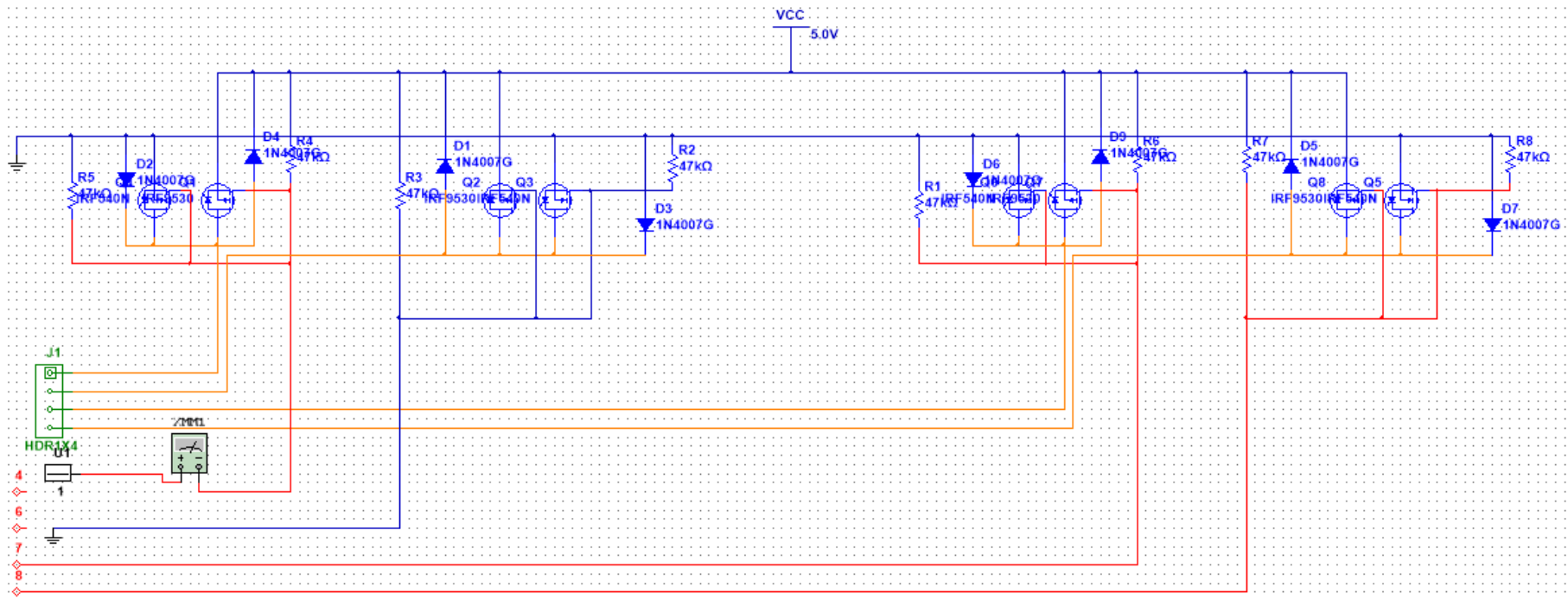


## 3.6 Matrixboard

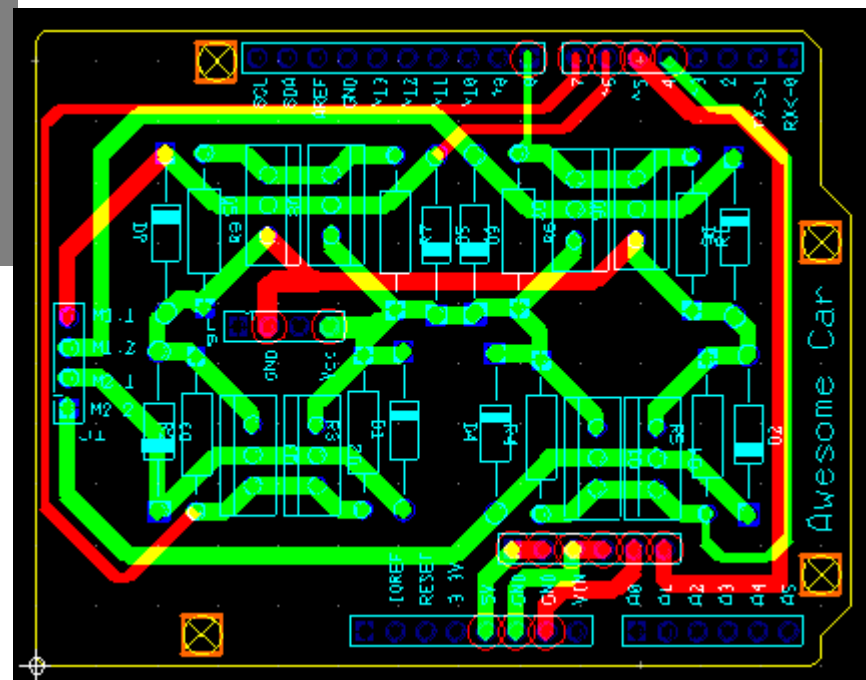
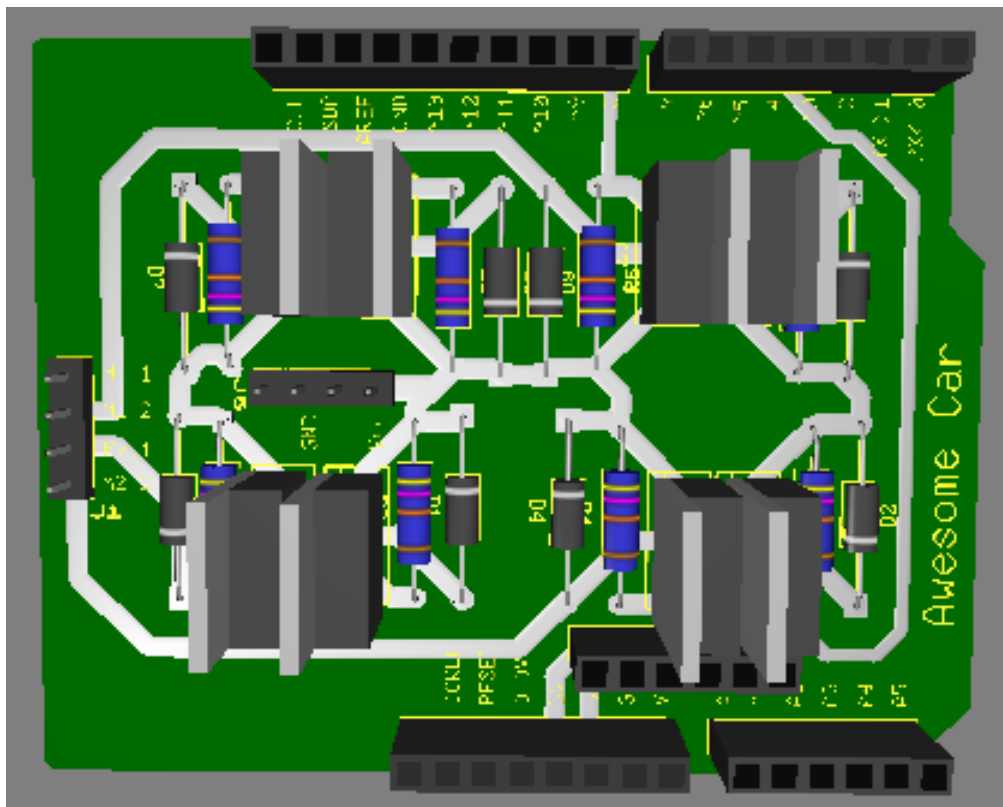




# 3.7 PCB ontwerpen



# 3.7 PCB ontwerp





# 3.7 PCB ontwerpen

## PCB Trace Amperage Chart

WWW.JOEZGARAGE.COM PCB TRACE AMP CHART REV. 1.1

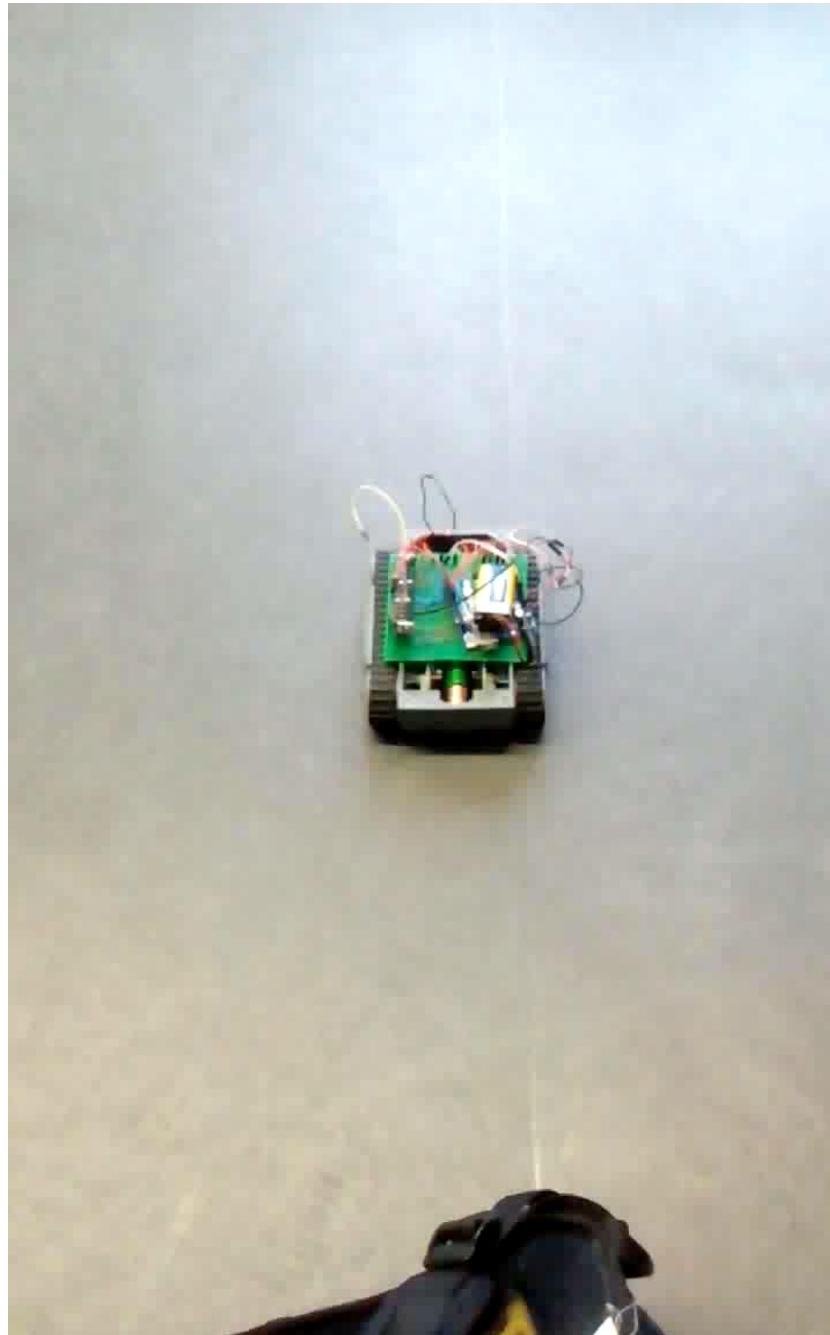
The following ratings are based on 2 OZ per square foot of copper and an ambient temperature of 90 degrees F (32 deg. C)

Acceptable temperature rise: 50 degrees F (10 deg. C)			
	MAX AMPERAGE		
Trace width	INTERNAL LAYER	EXTERNAL LAYER (open air)	Resistance per inch
0.005 (5 mils)	.360 amps (360ma)	1.013 amps (1013ma)	0.0535 ohms
0.010 (10 mils)	.602 amps (602ma)	1.616 amps (1616ma)	0.0268 ohms
0.015 (15 mils)	.811 amps (811ma)	2.124 amps	0.0178 ohms
0.020 (20 mils)	1.0025 amps	2.578 amps	0.01338 ohms
0.030 (30 mils)	1.350 amps	3.387 amps	0.00892 ohms
0.050 (50 mils)	1.966 amps	4.777 amps	0.00535 ohms
0.060 (60 mils)	2.248 amps	5.401 amps	.00446 ohms
0.070 (70 mils)	2.517 amps	5.991 amps	.00382 ohms
0.080 (80 mils)	2.777 amps	6.555 amps	.003346 ohms
0.100 (100 mils)	3.272 amps	7.617 amps	.002676 ohms
0.125 (125 mils) 1/8"	3.855 amps	8.852 amps	.002141 ohms
.250 (250 mils) 1/4"	6.415 amps	14.1154 amps	.00107 ohms
.500 (500 mils) 1/2"	10.676 amps	22.509 amps	.000535 ohms

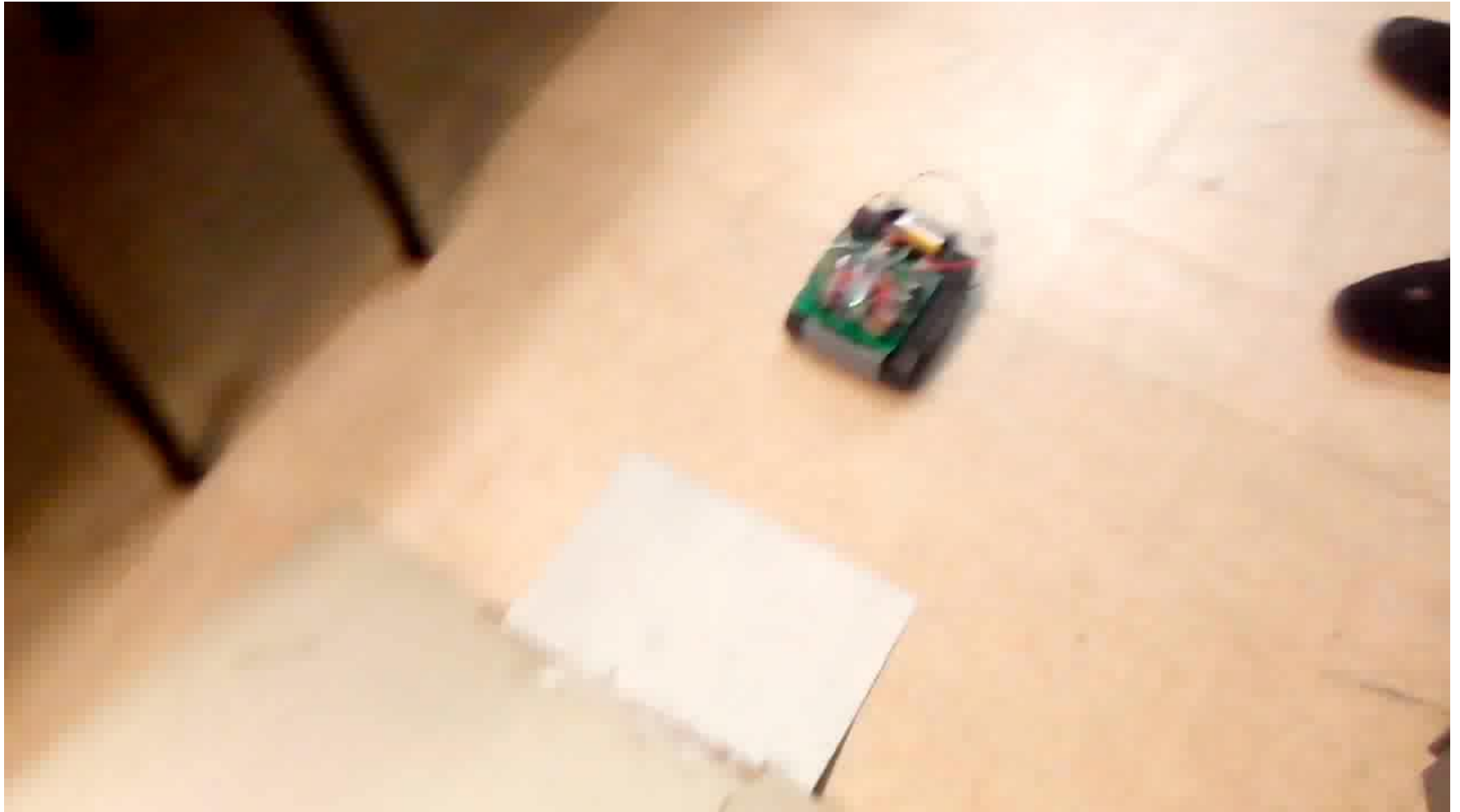
## 4. Conclusie

- De auto is functioneel
- Veel bijgeleerd:
  - Project en samenwerking
  - Werking H-brug
  - Toepassing Arduino
- Aandachtspunt:
  - Planning!

# Demo



# Demo - Brug



# Vragen?

**Bedankt voor u aandacht !**