Robotplatform

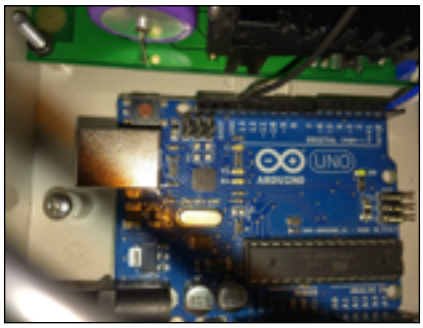
Jooris Matthias & Tom Nieskens

Elektronica-ICT Project 2 2013-2014

Index

Inleiding

Als opdracht voor ons project in het 2e jaar Elektronica-ICT kregen wij het Robotplatform. Dit was reeds een bestaand project van vorig jaar. Het platform bestaat uit een grote metalen plaat met daar onder 2 motoren van een elektrische step. We werken met 4 12V batterijen om het platform te voeden. De sturing wordt gedaan door een Arduino Uno. Op deze Arduino Uno zijn 2 PWM modules aangesloten om de motoren aan te sturen.




Doel

Toen we het project kregen werkte deze met een joystick die met 4 losse draden verbonden was met het platform. Omdat dit niet erg praktisch was en ook niet mooi uit zag was onze opdracht om er voor te zorgen dat het platform draadloos bestuurbaar was.

Natuurlijk zijn er verschillende mogelijkheden om het platform draadloos bestuurbaar te maken. Omdat de hardware hier op school aanwezig was en dit compatibel was met de Arduino hebben wij uiteindelijk gekozen om te werken met bluetooth.

Ook is het de bedoeling om een eigen inbreng in het project te steken. In de mate van het mogelijke een paar extra's toe te voegen.

Vooruitgang

Bekabeling

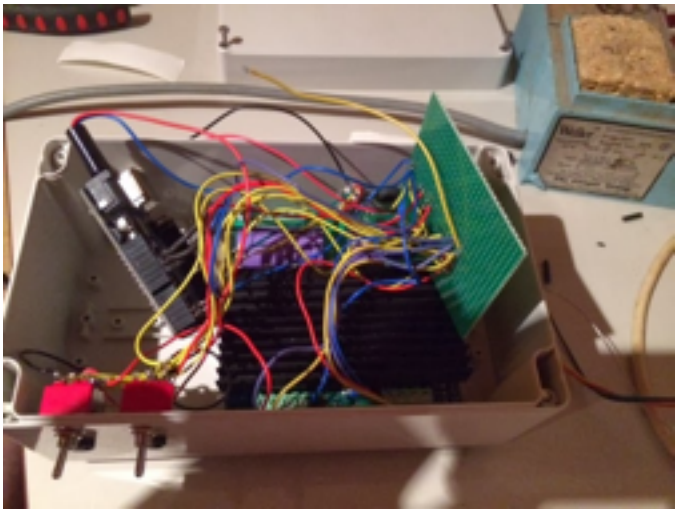
sturing kist

Toen we het project voor het eerst in handen kregen, viel het ons meteen al op dat de bekabeling niet netjes was. Er werden overal verschillende diameters gebruikt, kleurencombinaties die niet mogen. Er werd bijvoorbeeld een aardingskabel gebruikt om de batterijen met elkaar te verbinden. Kabels waren veel te lang en lagen slordig, solderingen die half „geplakt" werden.

Voor ons is het heel belangrijk als wij iets afgeven dat het ook netjes en in orde is. Daarom hebben wij er voor gekozen om meteen helemaal opnieuw te beginnen. We hebben alles uit elkaar gehaald, al de oude bekabeling verwijdert. Daarna zijn we begonnen met het her-organiseren van de sturing kist. Deze grijze kist is bevestigd onder het platform, in het midden tussen de beide motoren. In deze kist zet de Arduino als ook de 2 PWM modules en al de rest van de elektronica zoals een spanningsstabilisator.
De volledige bekabeling hier binnen hebben we dan opnieuw gedaan met de juiste kleuren.

Het resultaat mag er dan ook wel zijn als je gaat kijken naar het verschil:

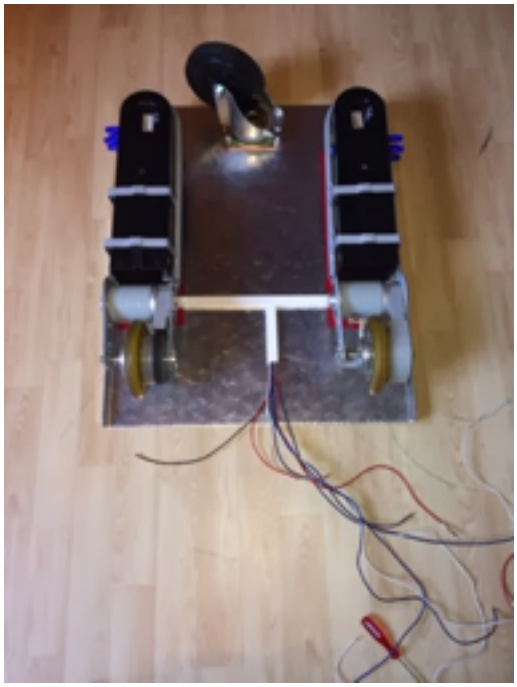
VOOR:



NA:

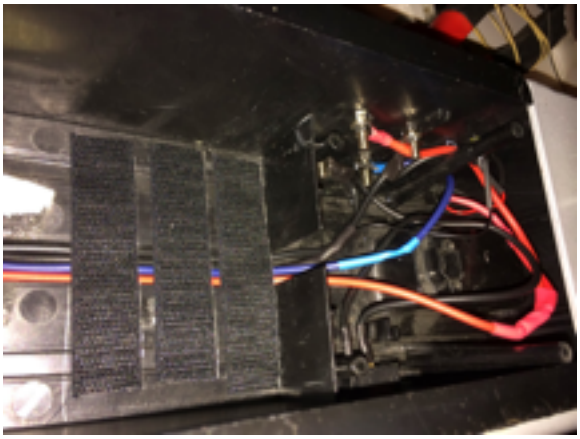
batterijen

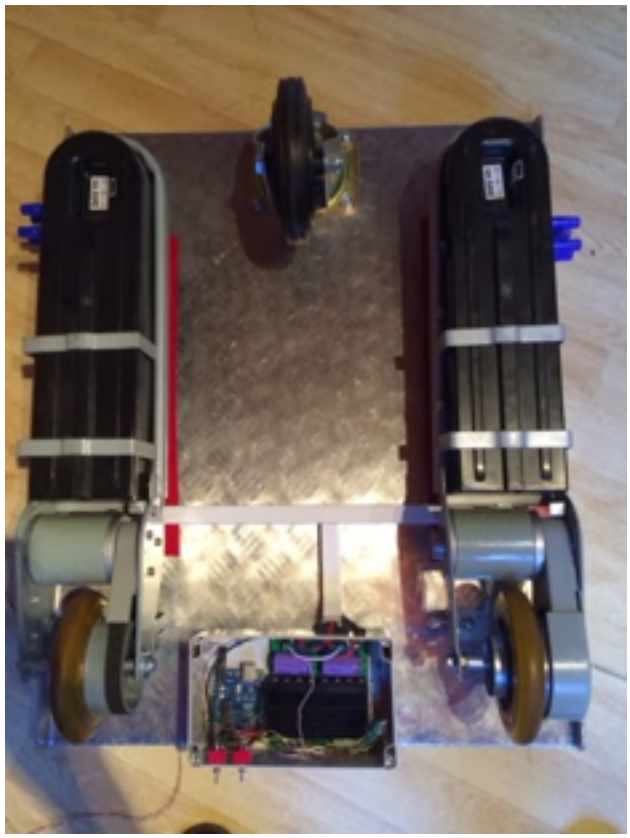
Aan de onderkant van het platform hebben we kabelgootjes bevestigd zodat de kabels die van de PWM's naar de batterijen en de motoren lopen niet zichtbaar zijn en al zeker niet over de grond zouden kunnen schuren tijdens het rijden of tussen de wielen geraken.



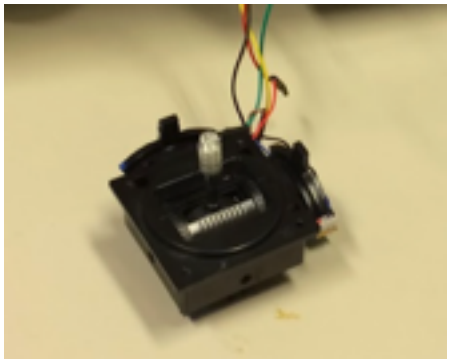
Ook de kabels die tussen de PWM en de batterijen lopen hebben we vervangen door dikkere kabels van de juiste kleur. Ook was er vroeger een schakeling met 2 relais voorzien om de schakeling te onderbreken om de batterijen op te laden. Deze schakeling was eigenlijk overbodig. Ze nam enkel veel plaats in beslag en veel kabels die er niet netjes uit zagen en de schakeling veel ingewikkelder maken. Daarom hebben wij dat stuk van de schakeling er uit gehaald en hebben wij een schakelaar tussen de batterijen gezet zodat je de serie schakeling voor de 24V te hebben gaat onderbreken en dat je de 12V laders gewoon kan aansluiten.

Onder de batterijen hebben we velcro tape geplakt zodat de batterijen blijven staan tijdens het rijden. Ook kunnen we daar onder de kabels netjes leggen. Onder de batterij zelf plakt uiteraard ook velcro zodat zeker alles op zijn plaats blijft staan.





Afstandsbediening



Oorspronkelijk was er een joystick bevestigd aan het platform. Dit zag er niet netjes uit en als we eventueel uitbreidingen wouden doen was het niet mogelijk met deze joystick. Of we moesten al een extra behuizing gaan maken voor deze uitbreiding. Daarom hebben we deze oude joystick vervangen door een andere afstandsbediening van een RC autootje.

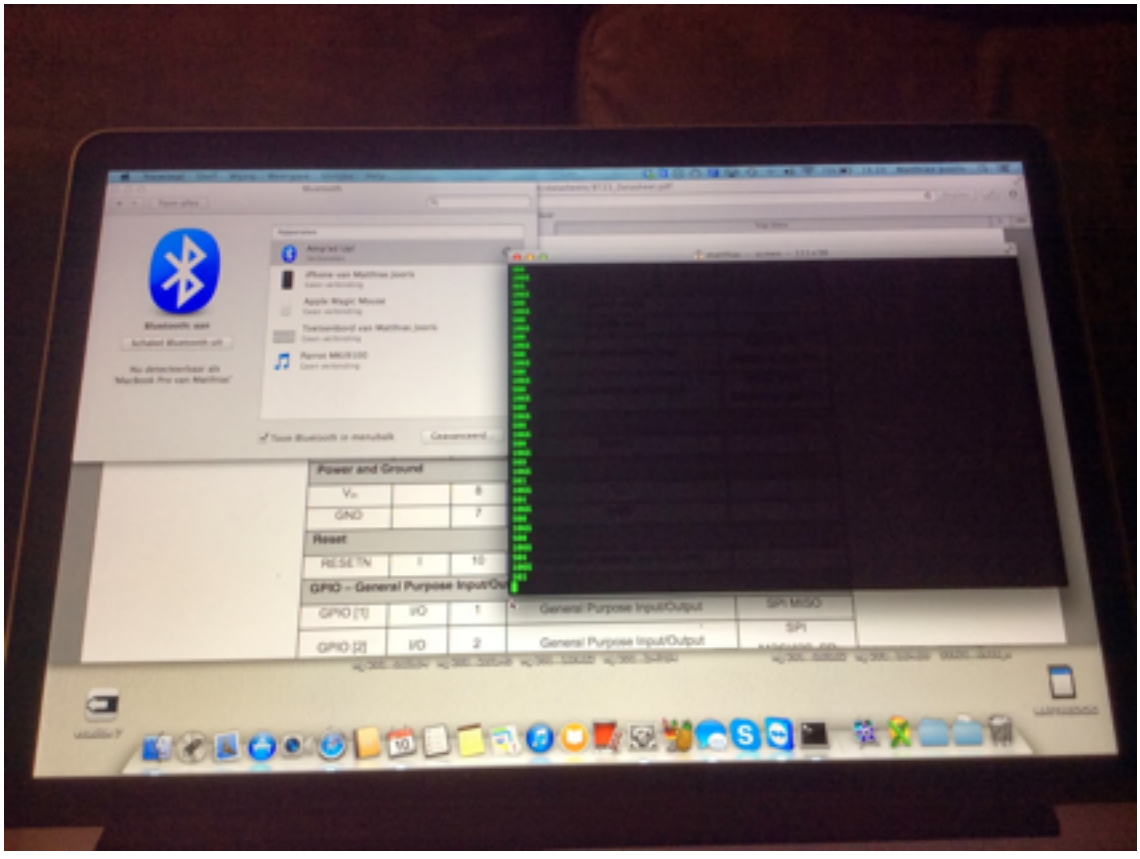
De hardware die hier in zat en de antenne konden wij op geen enkele manier gebruiken voor ons project. Daarom hebben we deze hardware er volledig uitgehaald en hebben we enkel de joystick laten zitten. Nu hebben we een lege behuizing waar nog veel plaats in over is. Zodat we hier later nog hardware in kunnen steken voor de bluetooth en dergelijke.

Bluetooth

De draadloze technologie die we gekozen hebben is Bluetooth. Op school hadden we de BT23 chips ter beschikking en de Bluegiga BLE112. Deze laatste had meer mogelijkheden maar deze was veel ingewikkelder in gebruik. Omdat de tijd erg beperkt was en de BT23 voldeed aan onze vereiste zijn we hiermee aan de slag gegaan.

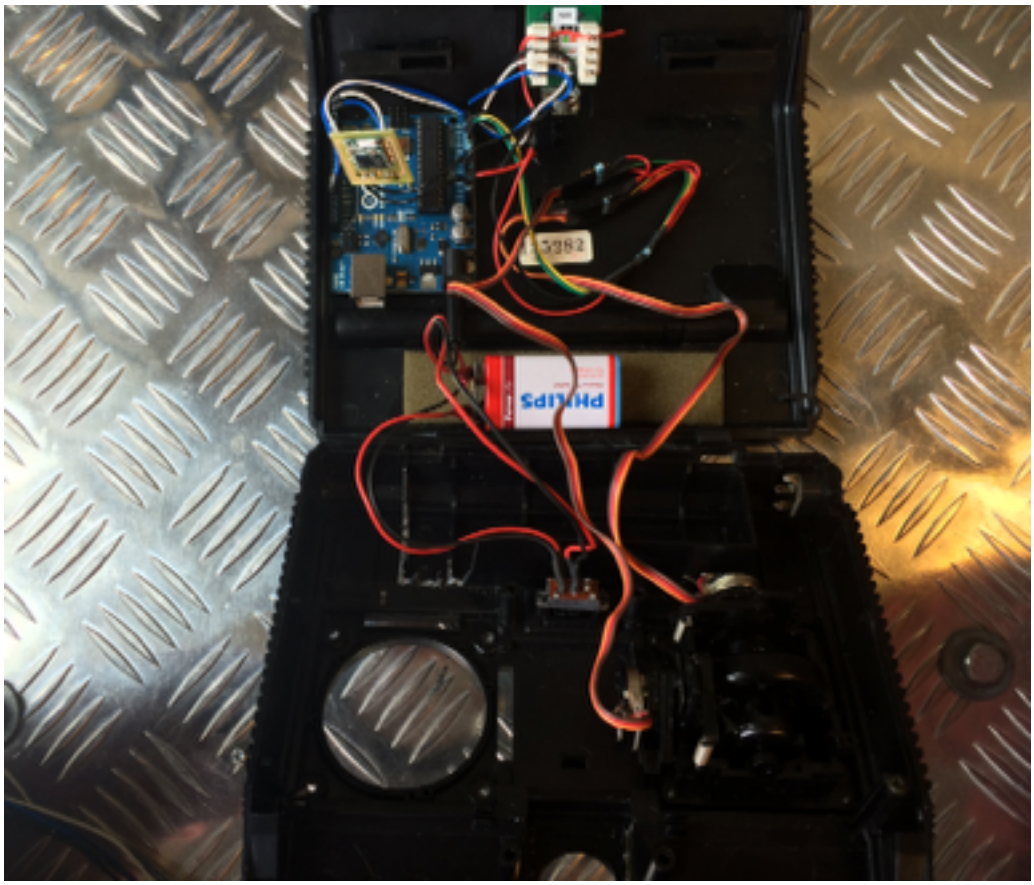
We hebben een klein printje ontworpen waar we de BT23 op gesoldeerd hebben. Het was belangrijk dat er geen koper onder de print door liep. Deze kon het bluetooth signaal gaan beïnvloeden.

Het aansluiten van deze print was eigenlijk vrij gemakkelijk. Wij hadden alle twee nog geen enkele ervaring met bluetooth. Na heel wat opzoekwerk hebben we dit eigenlijk vrij snel allemaal aan het werk gekregen. Deze chip moet aangesloten worden met 4 pinnen. 3.3V, Ground, Rx en Tx. Zijn Rx en Tx hebben we dan aangesloten op de Tx en Rx van de Ardiono in het platform. Deze is dan de ontvanger. Om te beginnen hebben we dit met mijn Mac getest, een bluetooth connectie gemaakt met deze ontvanger en in de terminal waardes verzonden. Dit werkte meteen, indien de juiste waardes werden verzonden begonnen de wielen van het platform te draaien.



Om de afstandsbediening ook met bluetooth te laten werken was er hier in ook een sturing nodig. Gelukkig hadden we in onze nieuwe afstandsbediening nog veel plaats over, daar we de oude schakeling er uit gehaald hebben. We hebben dus een 2e Ardiono Uno gebruikt en deze geplaatst in de behuizing van de afstandsbediening. Om deze te voeden hebben we een 9V batterij gebruikt. De Arduino is rechtstreeks met de 9V gevoed dus er was geen enkele spanningsstabilisator nodig. Op de 3.3V output van de Ardiono wordt de BT23 chip gevoed. De afstandsbediening had nog een power button ingebouwd, deze hebben we dus ook gebruikt om de Arduino aan en uit te laten schakelen zodat de batterij niet onnodig leeg zou gaan.

Omdat de afstandsbediening nog open moet gaan voor de batterij te vervangen en of eventuele wijzigingen, hebben we de kabels van de joystick zeker lang genoeg gelaten zodat hier nog aan gewerkt kan worden. De 2e joystick hebben we uit de afstandsbediening gehaald omdat deze niet gebruikt werd en er nu plaats genoeg is voor de Arduino en de bluetooth chip.

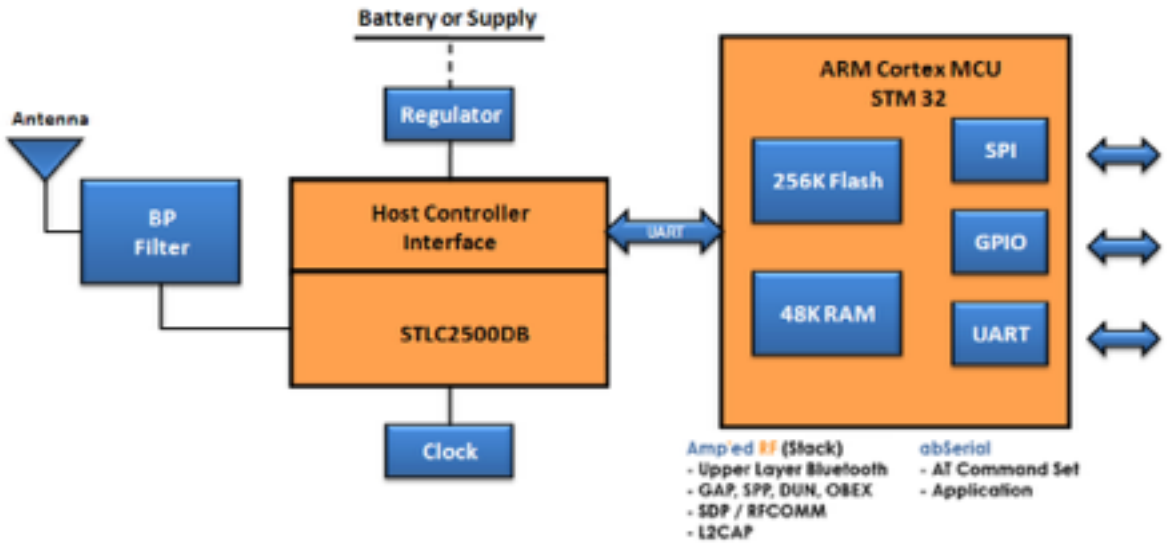


Data van de Amp'ed RF BT23:

Rating	Min	Typical	Max	Unit
Operating Temperature Range	-40	-	85	°C
Supply Voltage V_{IN}	2.7	3.0	3.6	Volts
Signal Pin Voltage	-	3.0	-	Volts
RF Frequency	2400	-	2483.5	MHz

Standard CPU Mode, 4 MHz
- UART supports up to 115 Kbps
- Data throughput up to 200 Kbps
- abSerial v1.3 (installed firmware)
- Shallow Sleep enabled

Modes (Typical Power Consumption)	Avg	Unit
ACL data 115K Baud UART at max throughput (Master)	25.6	mA
ACL data 115K Baud UART at max throughput (Slave)	25.8	mA
Connection, no data traffic, master	11.4	mA
Connection, no data traffic, slave	16.5	mA
Connection, 375ms sniff, slave (external LPO required)	3.6	mA
Standby, without deep sleep	11.5	mA
Standby, with deep sleep, no external LPO	0.5	mA
Standby, with deep sleep, with external LPO	80	uA
Page/Inquiry Scan, with deep sleep, no external LPO	1.4	mA
Page/Inquiry Scan, with deep sleep, with external LPO	0.9	mA
Bluetooth power down / CPU standby	25	µA



BT 23 Bluetooth Module Block Diagram

Parameters	Conditions	BT Spec	Typical	Unit
Antenna load			50	ohm
Radio Receiver				
Sensitivity level	BER < .001 with DH5	≤ -70	-85	dBm
Maximum usable level	BER < .001 with DH1	≥ -20	-9	dBm
Input VSWR			2.5:1	
Radio Transmitter				
Maximum output power	50 Ω load	-6 to +4	+6	dBm
Initial Carrier Frequency Tolerance		± 75	0	kHz
20 dB Bandwidth for modulated carrier		≤ 1000	932	kHz

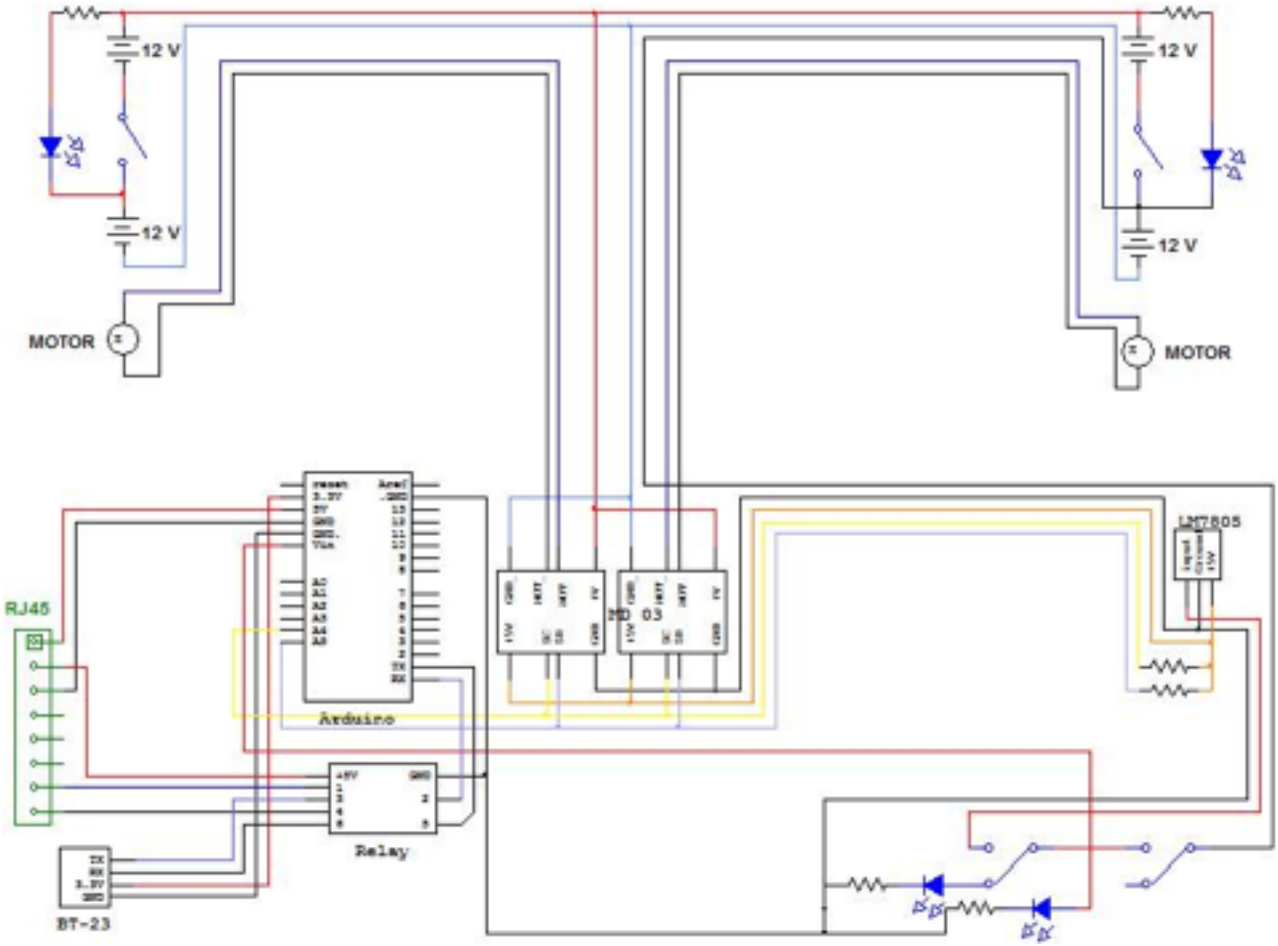
Bekabelde verbinding.

Het is fijn dat het nu allemaal draadloos werkt. Maar we vonden het belangrijk dat het platform altijd werkt. Ook in het geval dat de batterij van de afstandsbediening plat zou gaan, of net te plat is voor een stabiele verbinding te hebben. Het is niet praktisch om iedere keer alles open te draaien om een kabel in te gaan steken. Daarom hebben we besloten om te werken met een UTP kabel, op zowel het platform als op de afstandsbediening hebben we een RJ45 connector ingebouwd.
Iedereen heeft wel een UTP kabel liggen. Dus iedereen heeft wel een kabel liggen om hiervoor te gebruiken. in de afstandsbediening wordt er gewerkt met een software serial er wordt dus continu de zelfde waarde verzonden op zowel de RJ45 connector als op de bluetooth module. Dit hebben we gedaan om snel te kunnen switchen. Zodra je de kabel uit trekt zal de robot blijven werken maar dan via bluetooth en andersom.

In het platform zelf werken we met een kleine relais. De relais gaat er voor zorgen dat op het moment dat de kabel ingestoken wordt het signaal dat de bluetooth ontvangt niet meer zal aankomen op de Arduino. Zodat je via de kabel kan werken en niet dat er iemand toch nog via bluetooth de controle zou kunnen overnemen als de kabel in steekt. Van het moment dat de kabel uitgetrokken wordt zal het relais terug de data van de bluetooth naar de Arduino sturen (de bluetooth wordt niet uitgeschakeld, enkel de data wordt niet door gelaten) Dat zorgt ervoor dat er zeer snel omgeschakeld kan worden tussen kabel en bluetooth. Wij hebben van 1 UTP kabel het lipje afgebroken voor de veiligheid. Op die manier wordt de kabel uit de afstandsbediening getrokken als je te snel rijdt, zo kan de afstandsbediening niet vallen. Dan gaat er niets defect en wordt de robot automatisch stop gezet.

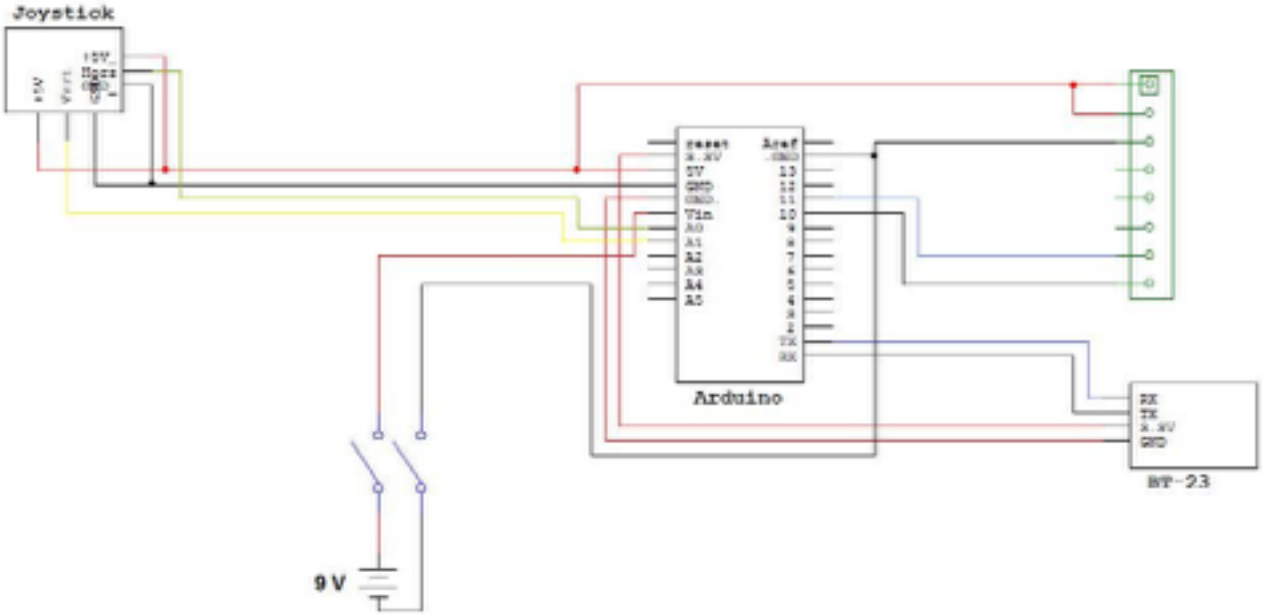
Schema's

Platform



- De 2 schakelaars in die tussen de batterijen staan dienen voor het opladen. Zoals je links op de afbeelding kan zien, als het ledje brandt zijn de batterijen in serie geschakeld. Nu kan je rijden met de robot. Als je de schakelaar om zet gaat deze led uit en kan je het platform opladen. Er hangen een + en een - sticker boven de klemmen zodat je de laders niet verkeerd zou aansluiten.
- Het relais gaat er voor zorgen dat als de UTP kabel in steekt, de bluetooth zijn data niet meer doorgezonden wordt maar enkel nog data over de kabel.
- De 2 schakelaars onderaan dienen om de voeding van de Arduino en de PWM modules apart te kunnen schakelen.

Afstandsbediening



- Op de bovenste pen van de RJ45 connector (groene connector op het schema) is de rode kabel aangesloten die naar de joystick gaat. Deze wordt terug gekoppeld naar de 2e pen. Deze zorgt ervoor dat in de ontvanger opgemerkt zal worden dat de kabel insteekt. Deze 2e pen geeft de voedingsspanning mee om het relais te laten aantrekken zodat de data van de bluetooth van de ontvanger uitgeschakeld wordt. Dit is nodig omdat het niet mogelijk is om het parseInt commando te gebruiken in de software serial.

- De schakelaar dient om de voeding van de Arduino en op zijn beurt de bluetooth module uit te schakelen. Om via kabel te werken maakt de stand van deze schakelaar niet uit want de kabel zal automatisch voeding mee sturen. Op die manier kan je ook gebruik maken van de afstandsbediening als de batterij plat moest zijn.

Software

Code platform

Hier onder kan u de software vinden die in de Arduino in de ontvanger zit. In het platform dus.
Wat we eigenlijk doen in de code is:
Krijgt hij 1066 binnen volgt daarna de waarde voor vooruit/achteruit te gaan.
Krijgt hij 1065 binnen volgt daarna de waarde voor links/rechts te gaan.

Als de waardes 500 zijn staat de robot stil, dat is eigenlijk het „nulpunt" op het assenstelsel.
Om er voor te zorgen dat elk apparaat kan verbinden met het platform wordt er niks gedaan met de bluetooth in deze code. Enkel in de afstandsbediening wordt er voor gezorgd dat er een connectie wordt gemaakt met het platform. De bluetooth module in het platform krijgt zijn voeding dus zend een netwerk uit. Wat hij dan ontvangt wordt in de Rx van de Arduino ingestuurd. Met deze waardes gaat de Arduino dan beslissen hoe de motoren moeten bewegen.
In de code kan u zien dat deze waardes omgevormd worden en uiteindelijk worden opgeslagen in Speed1 en Speed2 Deze waardes worden dan via I2C verzonden naar de bijbehorende PWM module.

Voor de rest staat in comment in de code al de info die nodig is om de code volledig te begrijpen.

```
#include <Wire.h>
#define MD03_1    0x58
#define MD03_2    0x59
#define CMD_reg   0x00
#define SPEED_reg 0x02
#define ACC_reg   0x03

 int valueFB = 500; //variable value forward backward
 int valueLR = 500; //variable value left right



 unsigned long speed1;
```

```
 unsigned long speed2;
 int speed;
 int recievedChar = 0;
void setup()
{
  Serial.begin(115200);// begin the serial communicaton


  {
  Wire.begin(); // join i2c bus (address optional for master)
}
}

void loop()
{
  recievedChar =Serial.parseInt();
  if (recievedChar > 0)
  {
   if(recievedChar==1066){// if it sees 1066 then it knows that valueFB is comming
   valueFB = Serial.parseInt();  //Reads integers as integer rather than ASCI. Anything else returns 0
   Serial.println(valueFB);// prints valueFB so I could see it int the serial monitor

   }
   if(recievedChar==1065){// if it sees 1066 then it knows that valueLR is comming
   valueLR = Serial.parseInt();  //Reads integers as integer rather than ASCI. Anything else returns 0
   Serial.println(valueLR);// prints valueLR so I could see it int the serial monitor

   }
  }else{
  valueLR = 500;
  valueFB = 500;
  }

 if(valueFB>510){                                          //forwards
     if(valueLR>505){                                       //forwards+left
     speed1 =((valueFB-400)/2);
     speed2 =((valueFB-400)/2)-((valueLR-450)/2) ;       // left motor slows down
     }
     else if(valueLR<495){                                  //forwards+right
     speed1 = ((valueFB-400)/2)-((550-valueLR)/2);      // right motor slows down
     speed2 =((valueFB-400)/2) ;
     }
     else{                                                 //forwards
     speed1 = ((valueFB-400)/2);
     speed2 = ((valueFB-400)/2);
```

```
    }

  Wire.beginTransmission(MD03_1);
  Wire.write(SPEED_reg);
  Wire.write(speed1);
  Wire.endTransmission();
  Wire.beginTransmission(MD03_2);
  Wire.write(SPEED_reg);
  Wire.write(speed2);
  Wire.endTransmission();

  Wire.beginTransmission(MD03_1);
  Wire.write(ACC_reg);
  Wire.write(27);
  Wire.endTransmission();
  Wire.beginTransmission(MD03_2);
  Wire.write(ACC_reg);
  Wire.write(27);
  Wire.endTransmission();
  Wire.beginTransmission(MD03_1);
  Wire.write(CMD_reg);
  Wire.write(1);              // sets motor forward
  Wire.endTransmission();
  Wire.beginTransmission(MD03_2);
  Wire.write(CMD_reg);
  Wire.write(1);              // sets motor forward
  Wire.endTransmission();

}
else if(valueFB<490){                                  //backwards
    if(valueLR>505){                                   //backwards+left
   speed1 =((600-valueFB)/2);
   speed2 =((600-valueFB)/2)-((valueLR-450)/2);      // left motor slows down
   }
   else if(valueLR<495){                               //backwards+right
   speed1 =((600-valueFB)/2)-((550-valueLR)/2);      // right motor slows down
   speed2 =((600-valueFB)/2);
   }
   else{                                              //backwards
   speed1 =((600-valueFB)/2);
   speed2 =((600-valueFB)/2);
   }

  Wire.beginTransmission(MD03_1);
  Wire.write(SPEED_reg);
```

```
  Wire.write(speed1);
  Wire.endTransmission();

  Wire.beginTransmission(MD03_2);
  Wire.write(SPEED_reg);
  Wire.write(speed2);
  Wire.endTransmission();

  Wire.beginTransmission(MD03_1);
  Wire.write(ACC_reg);
  Wire.write(27);
  Wire.endTransmission();
  Wire.beginTransmission(MD03_2);
  Wire.write(ACC_reg);
  Wire.write(27);
  Wire.endTransmission();
  Wire.beginTransmission(MD03_1);
  Wire.write(CMD_reg);
  Wire.write(2);              // sets motor backwards
  Wire.endTransmission();
  Wire.beginTransmission(MD03_2);
  Wire.write(CMD_reg);
  Wire.write(2);              // sets motor backwards
  Wire.endTransmission();

}
else if (valueFB>490 and valueFB<510){                    //left and right
   if(valueLR>505){                                       //left
  speed1 =((valueLR-400)/2);

  Wire.beginTransmission(MD03_1);
  Wire.write(SPEED_reg);
  Wire.write(speed1);
  Wire.endTransmission();
  Wire.beginTransmission(MD03_2);
  Wire.write(SPEED_reg);
  Wire.write(speed1);
  Wire.endTransmission();

  Wire.beginTransmission(MD03_1);
  Wire.write(ACC_reg);
  Wire.write(27);
  Wire.endTransmission();
  Wire.beginTransmission(MD03_2);
  Wire.write(ACC_reg);
```

```
Wire.write(27);
Wire.endTransmission();

Wire.beginTransmission(MD03_1);
Wire.write(CMD_reg);
Wire.write(1);              // sets motor forwards
Wire.endTransmission();
Wire.beginTransmission(MD03_2);
Wire.write(CMD_reg);
Wire.write(2);              // sets motor backwards
Wire.endTransmission();

 }
 else if(valueLR<495){                                  //right
speed1 =((600-valueLR)/2);

Wire.beginTransmission(MD03_1);
Wire.write(SPEED_reg);
Wire.write(speed1);
Wire.endTransmission();
Wire.beginTransmission(MD03_2);
Wire.write(SPEED_reg);
Wire.write(speed1);
Wire.endTransmission();

Wire.beginTransmission(MD03_1);
Wire.write(ACC_reg);
Wire.write(27);
Wire.endTransmission();
Wire.beginTransmission(MD03_2);
Wire.write(ACC_reg);
Wire.write(27);
Wire.endTransmission();
Wire.beginTransmission(MD03_1);
Wire.write(CMD_reg);
Wire.write(2);              // sets motor backwards
Wire.endTransmission();
Wire.beginTransmission(MD03_2);
Wire.write(CMD_reg);
Wire.write(1);              // sets motor forwards
Wire.endTransmission();

 }
 else{                                                  //stops
```

```
    Wire.beginTransmission(MD03_1);
    Wire.write(SPEED_reg);
    Wire.write(0);
    Wire.endTransmission();
    Wire.beginTransmission(MD03_2);
    Wire.write(SPEED_reg);
    Wire.write(0);
    Wire.endTransmission();

    Wire.beginTransmission(MD03_1);
    Wire.write(ACC_reg);
    Wire.write(27);
    Wire.endTransmission();
    Wire.beginTransmission(MD03_2);
    Wire.write(ACC_reg);
    Wire.write(27);
    Wire.endTransmission();
    Wire.beginTransmission(MD03_1);
    Wire.write(CMD_reg);
    Wire.write(0);              // stops motor
    Wire.endTransmission();
    Wire.beginTransmission(MD03_2);
    Wire.write(CMD_reg);
    Wire.write(0);              // stops motor
    Wire.endTransmission();

      }
  }
}
```

Code afstandsbediening

Hieronder volgt de code die zit in de Arduino van de afstandsbediening. In deze code wordt er voor gezorgd dat de bluetooth connectie tot stand wordt gemaakt, en blijft.
De regel: Serial.println("AT+AB SPPConnect 00043e26120c"); zorgt hiervoor.
00043e26120c is de code van het mac adres van de bluetooth chip in het platform.

We hebben er voor gekozen om dit niet in het begin van de code te doen. Want als je aan het rijden bent en de bluetooth connectie gaat verloren, dan ben je de verbinding kwijt tot je de afstandsbediening opnieuw zou opstarten.
Daarom hebben wij deze regel in de code gezet zodat bij iedere cyclus (elke keer bij het verzenden van de waarden FB en LR) opnieuw wordt gezegd tegen de module dat hij moet verbinden met het platform. Op die manier wordt meerdere keren per seconde de connectie opnieuw tot stand gebracht. Als je dan te ver van het platform weg bent, wordt de verbinding meteen weer tot stand gebracht als je weer dichter in de buurt komt.

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX
int joyFB = A1 ; //joystick forward and backward
 int joyLR = A2 ; //joystick left and right
 int valueFB = 500; //variable value forward backward
 int valueLR = 500; //variable value left right
void setup()
{
  Serial.begin(115200);
  pinMode(joyFB,INPUT);
  pinMode(joyLR,INPUT);

  mySerial.begin(115200);
}
void loop()
{
valueFB = analogRead(joyFB);
valueLR = analogRead(joyLR);

Serial.println(1066);
mySerial.println(1066);
Serial.println (valueFB); // send valueFB to rx arduino
mySerial.println(valueFB);
delay(80);
```

```
Serial.println(1065);
mySerial.println(1065);
Serial.println (valueLR); / / send valueLR to rx arduino
mySerial.println(valueLR);
delay(80);
Serial.println("AT+AB SPPConnect 00043e26120c");
delay(80);
}
```

Android applicatie

De laatste weken zijn we bezig geweest met het ontwikkelen van een Android applicatie. Waarom we voor Android gekozen hebben en niet voor IOS is omdat een Android app gemakkelijk op een toestel te zetten is en een IOS app kan je enkel op je eigen toestel zetten als je een Mac hebt, wat bij ons wel het geval is maar toch. In de toekomst kan de volgende groep hier dan misschien niet meer verder.

Wij hadden geen enkele ervaring met Android. We zijn gaan googelen naar software en zijn er mee aan de slag gegaan.

We zijn begonnen met het maken van een joystick, de rode bol kan je doen bewegen binnen in de grijze bol. Deze gaat dan een X en een Y waarde genereren die we later zouden moeten doodsuren via bluetooth naar onze robot (nadat deze waardes zijn omgevormd natuurlijk)

Helaas zijn we op dat moment een beetje vast gelopen. Het lukt om een connectie te maken met de robot via bluetooth. De app is verbonden maar vanaf het moment dat we data beginnen te verzenden crasht de app. We hebben uren gespendeerd om dit proberen op te lossen. Ook hebben we hulp gekregen van Mr Palmaers, leerkracht programmeren voor Android en IOS maar helaas had hij niet veel tijd en op de periode dat hij heeft kunnen helpen is het ook hem niet gelukt om dit probleem op te lossen.
Op de volgende pagina kan je het stukje code zien dat er voor zorgt dat er een bluetooth verbinding/communicatie is. De fout zit in de code „os.write(1065)" Deze regel staat in comment. Haal deze uit comment en de app blijft werken (natuurlijk wordt er dan geen data verzonden) maar helaas hadden we geen tijd meer over dit jaar om dit nog verder uit te werken wat we erg jammer vinden omdat we toch al zo ver geraakt zijn met deze app. De app kan je terug vinden op github. de hele code is te lang om hier in te zetten.

```
        /*BluetoothAdapter ba = BluetoothAdapter.getDefaultAdapter();
        Set<BluetoothDevice> devices = ba.getBondedDevices();
        for(BluetoothDevice device : devices) {
           Log.i("BT", "device: " + device.getAddress() + "=> " + device.getName());
           if(device.getAddress().equals("00:04:3E:26:12:0C")) {
              Log.i("platform","gevonden");
              UUID uuidSPP = UUID.fromString("00001101-0000-1000-8000-00043E26120C");
              try {
                 BluetoothSocket sock =
device.createRfcommSocketToServiceRecord(uuidSPP);
                 Log.i("BT", "socket ok **" + sock + "**");
                 OutputStream os = sock.getOutputStream();

                 Log.i("BT", "outputstream **" + os + "**");
                 if(os == null)
                    Log.i("BT", "os null");
                 else
                 Log.i("BT", "os niet null");


                 //os.write(1065);
                 /*os.flush();

                    try {
                       Thread.sleep(80);
                    } catch (InterruptedException e) {
                       e.printStackTrace();
                    }
                 os.write(1065);
                 os.write(600);
                 os.write(1065);
                 os.write(600);
```

(de code in het groen is waar het mis gaat in de app)