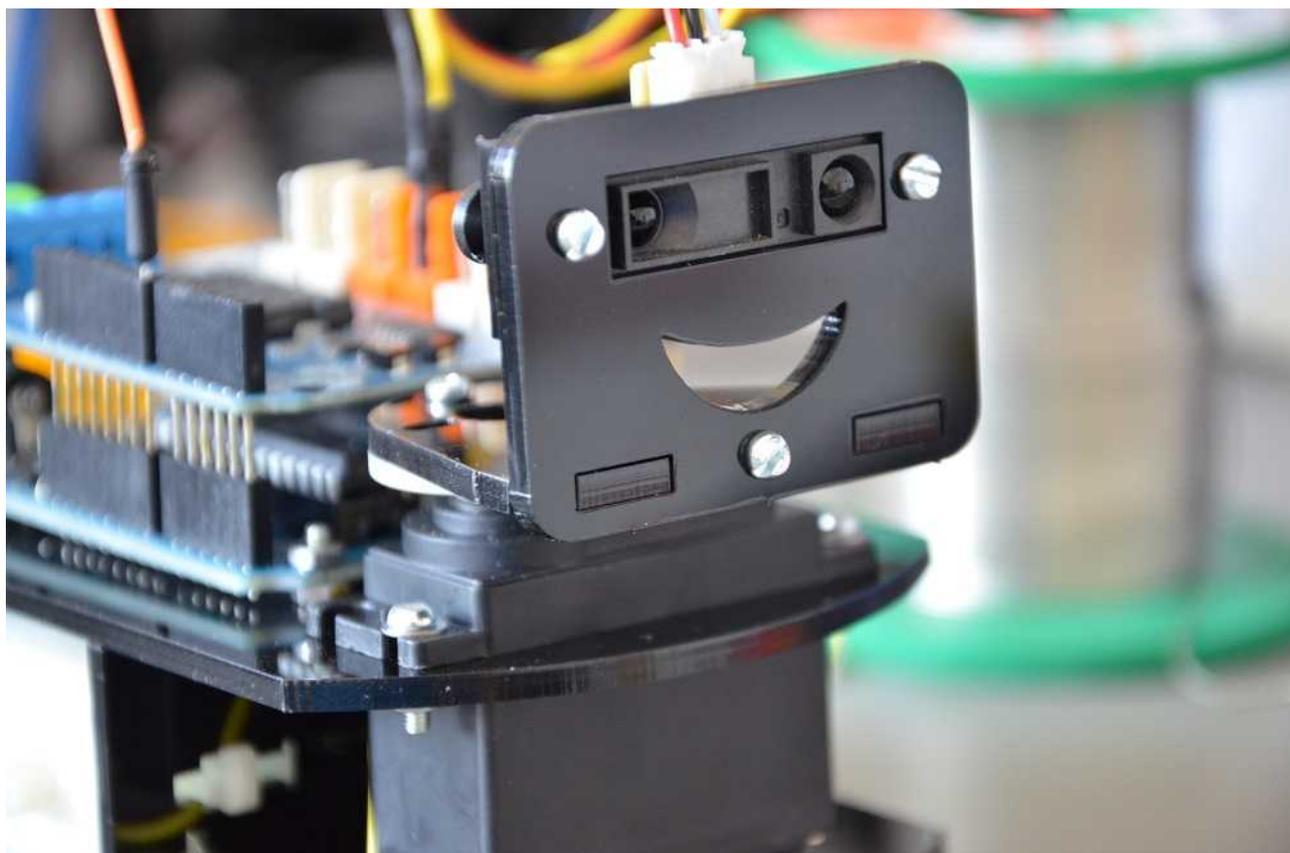


Starter Kit Roboter V2



Die Stückliste

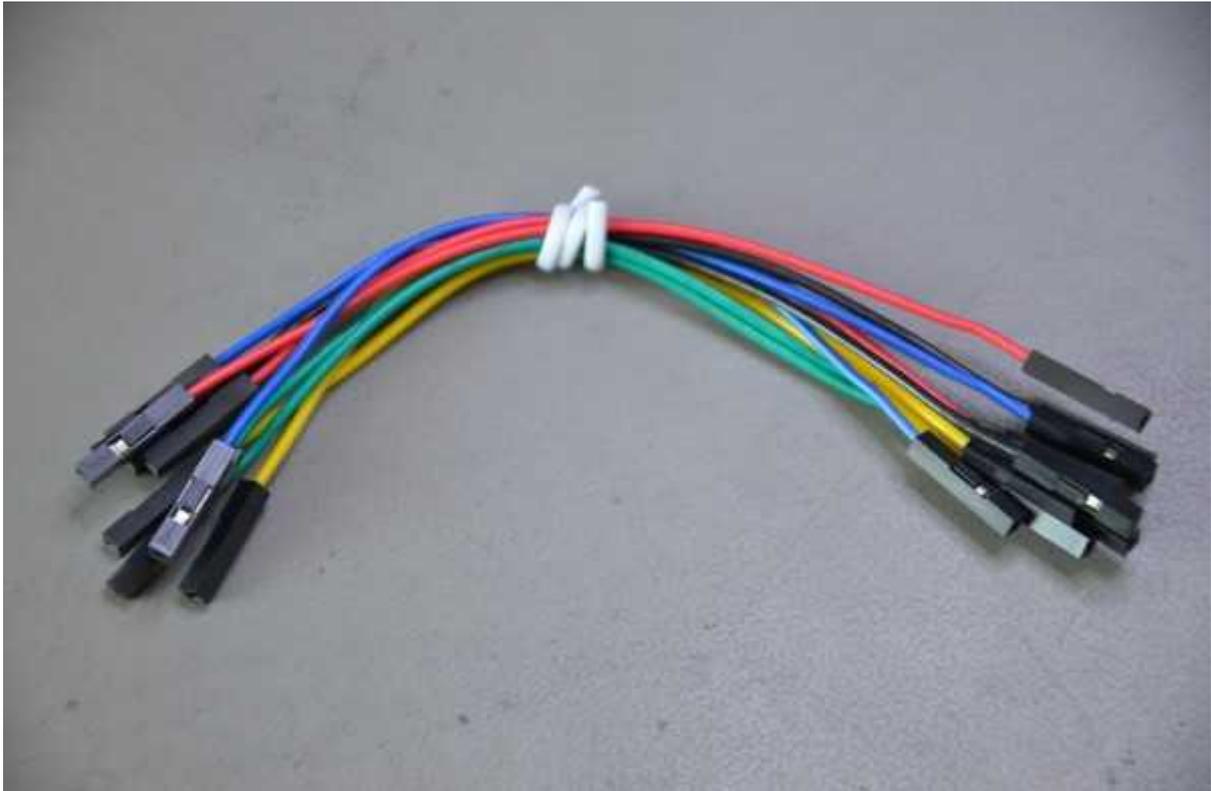
1x Arduino Uno



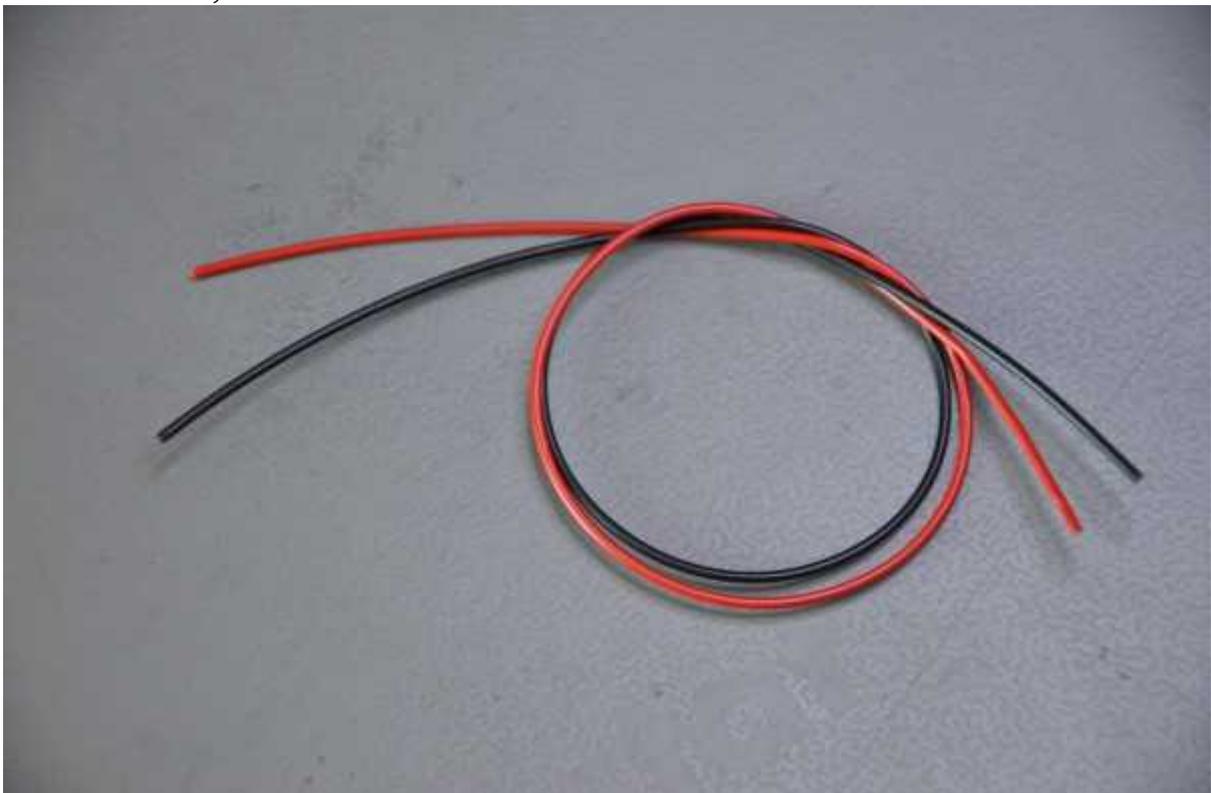
1x Arduino Motor Shield



1x Jumper Wires F/F (10 Stk. Päckchen)



Stückchen Draht, rot und schwarz



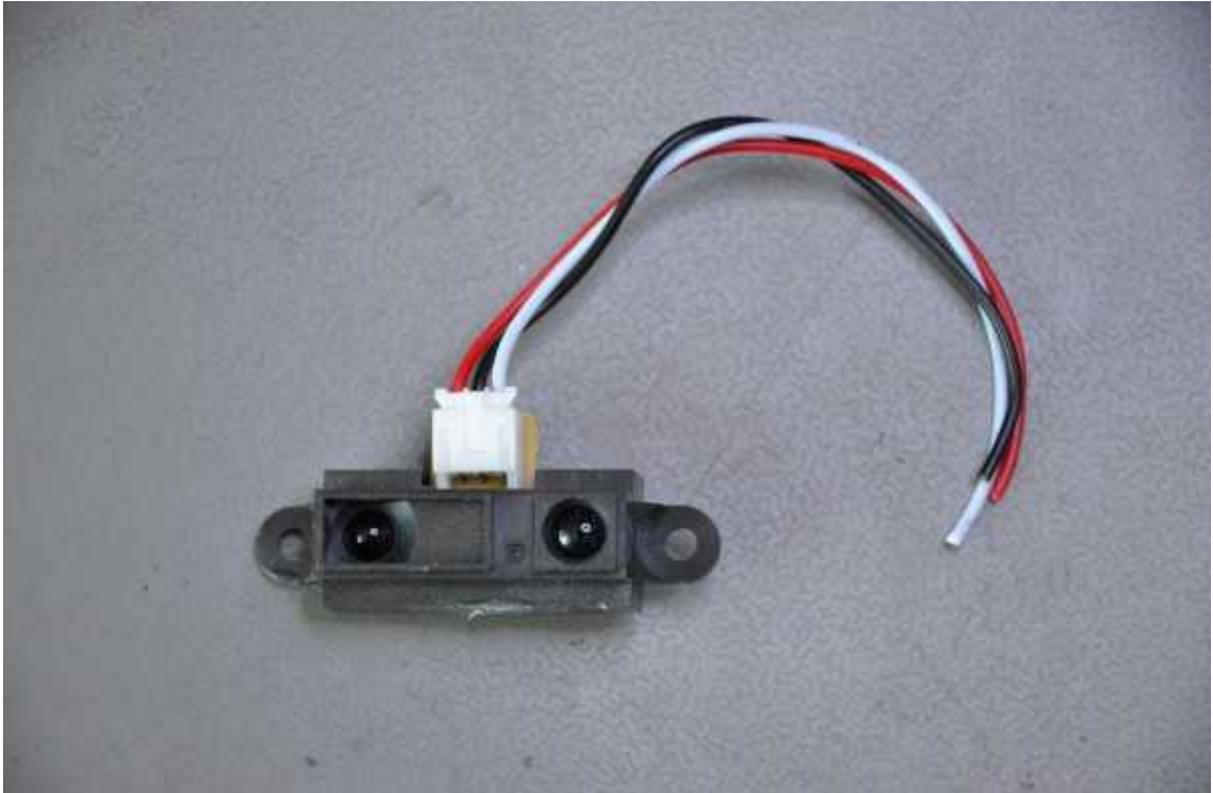
Stückchen Draht irgendeiner Farbe, außer rot oder schwarz



1x HiTEC HS-311 Standard Servo



1x Sharp IR-Distanzmesser (GP2Y0A21YK0F). Denkt dran, dass die Sensoren ohne Anschlusskabel verkauft werden. Also immer ein passendes Anschlusskabel mitbestellen.



1x Batteriehalter für 4x 1,5V AA Batterien



2x Solarbotics GM9 Motoren



2x Solarbotics Räder für GM9 Motoren



Etwas Schrumpfschlauch



1x Set mit Acryl-Teilen



1x Set Schrauben und Muttern



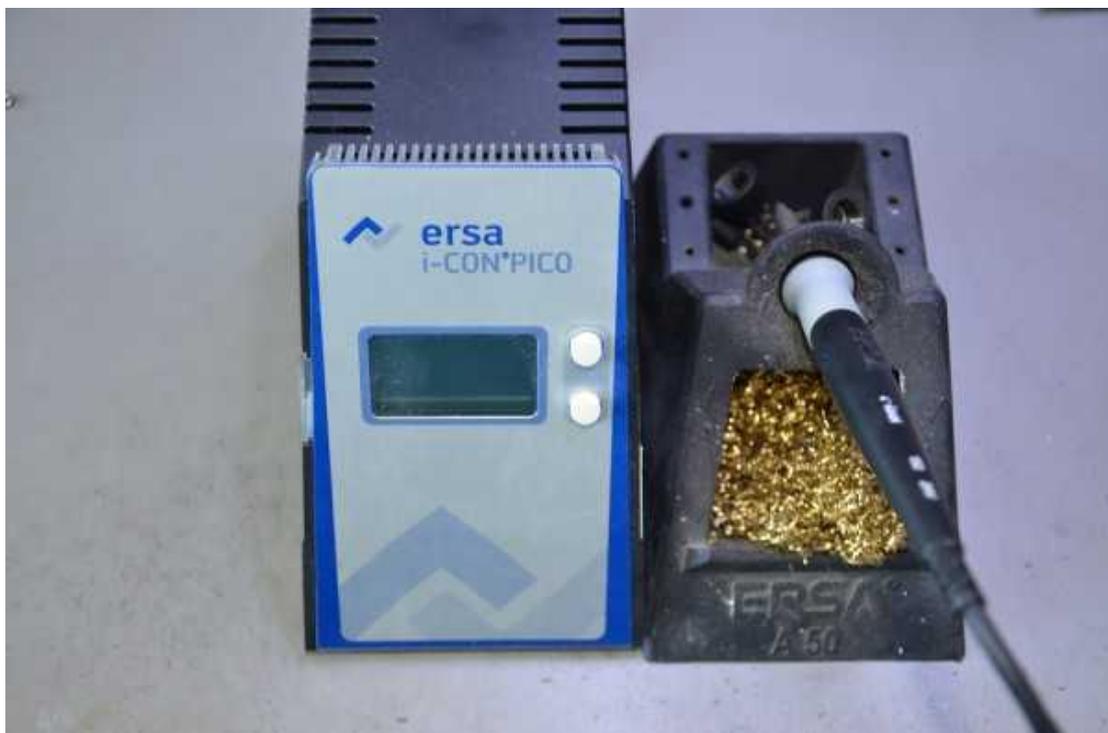
Menge	Typ
4	Schraube M2,5x30
13	Schraube M2,5x10
2	Schraube M2,5x15
2	Schraube M2x10
2	Schraube M2x18
2	Blechschaube (6mm lang)
22	Mutter M2,5
4	Mutter M2
2	Unterlegscheibe M2,5

Feuerzeug, Seitenschneider, Schraubendreher (Kreuz und Schlitz), Abisolierzange und evtl. ein kleines Messer (nicht abgebildet).

Evtl. kann bei der Montage, eine Pinzette sehr nützlich sein



Und mein Lieblingswerkzeug – eine Lötstation (oder Lötkolben) und etwas Lötzinn.
Löttemperatur ca. 350°C



Bist du bereit und hast alles? Dann wollen wir uns mal in die faszinierende Welt der selbstgebauten Roboter hineinstürzen :-)

Schritte

Ein wichtiger Hinweis: **BITTE KEINE GEWALT BEIM AUFBAU DES ROBOTERS ANWENDEN!**

Alle Bauteile sind relativ robust und halten Einiges aus. Jedoch können bei falscher Handhabung, die Acryl-Glas- oder andere Teile kaputt gehen.

1. Auf dem Bild ist die obere Platte des Bots zu sehen.



Schutzfolie nicht vergessen abzuziehen.



2. Jetzt werden die vier M2,5 x 10 Schrauben und M2,5 Muttern geschraubt. Achte dabei auf die Ausrichtung der Platte.

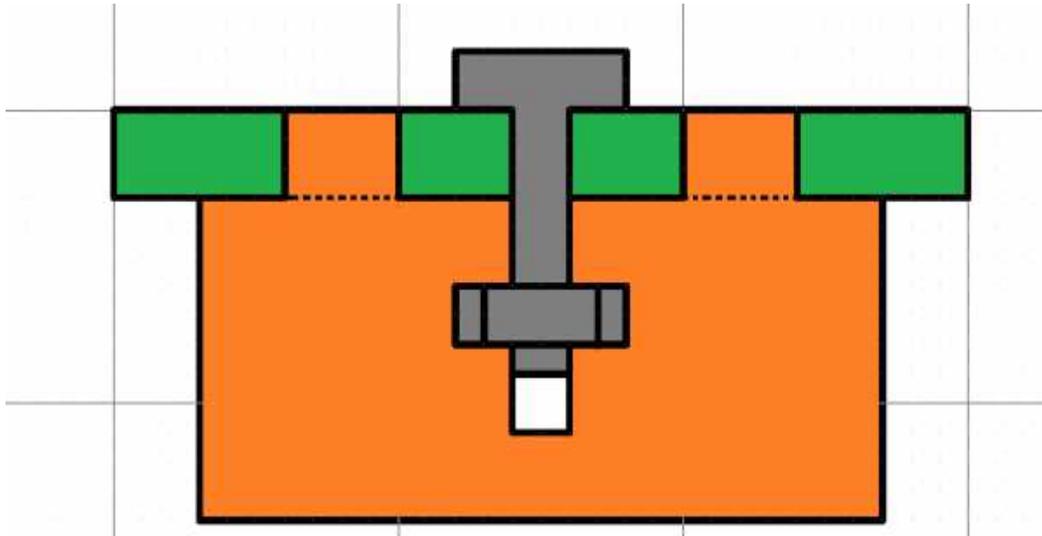


3. Nun werden zwei Acryl-Teile, im 90° Winkel miteinander verschraubt. Suche dir die Teile aus, die man auf dem Bild sieht. Die Schraube ist wieder eine M2,5 x 10 und M2,5 Mutter.



Im Prinzip ist so eine 90° Verschraubung gar nicht so kompliziert. Siehe Bild unten.

Stelle dir vor, das Grüne (oben) ist die große Platte (horizontal) und das orangene (vertikal, unten) die kleine Platte. Die kleine Platte wird in die passenden Fassungen der großen Platte eingesteckt. In die untere Platte wird (horizontal) eine passende Mutter eingeschoben und durch das Loch der oberen Platte, wird die Schraube eingeschraubt. Durch die Verschraubung werden die beiden Teile zusammengehalten. Merke dir diesen Schritt, denn in allen weiteren Schritten mit Acryl-Glas werden die Teile im 90° nach genau diesem Prinzip verschraubt.



Und so sieht es dann aus



4. Nun wird die vordere Platte mit zwei M2,5×10 Schrauben und M2,5 Muttern verschraubt.



5. In das hintere Plättchen wird eine M2,5 Mutter eingesetzt.



6. Die zweite große Platte vorsichtig oben drauf legen und die Schraube einschrauben. Achte darauf, dass die rechteckige Löcher (grüner Pfeil) der beiden Platten, übereinander angeordnet sind.



An den Markierten Stellen kommen erst mal keine Schrauben rein. Diese werden im Schritt 13 und 14 zusammen mit dem Batteriehalter verbaut.



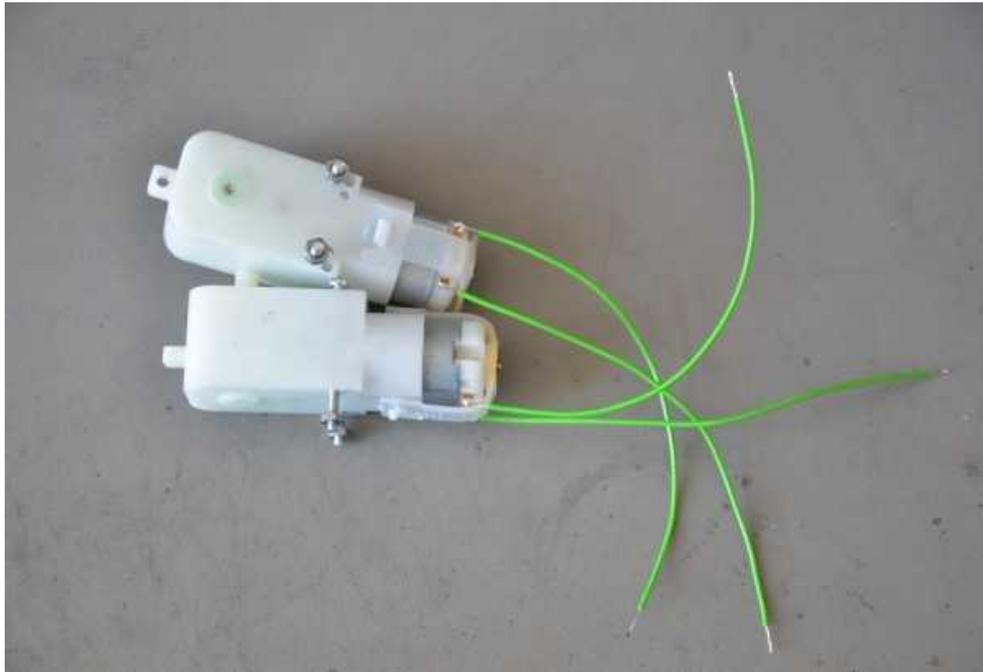
7. Jetzt werden die GM9 Motoren vorbereitet. Dazu benötigen wir noch 4 Stück M2,5×30 Schrauben, 2 Stück M2,5×10 Schrauben und 7 Stück M2,5 Muttern. Wobei die kleineren Schrauben ein paar Schritte später eingesetzt werden.



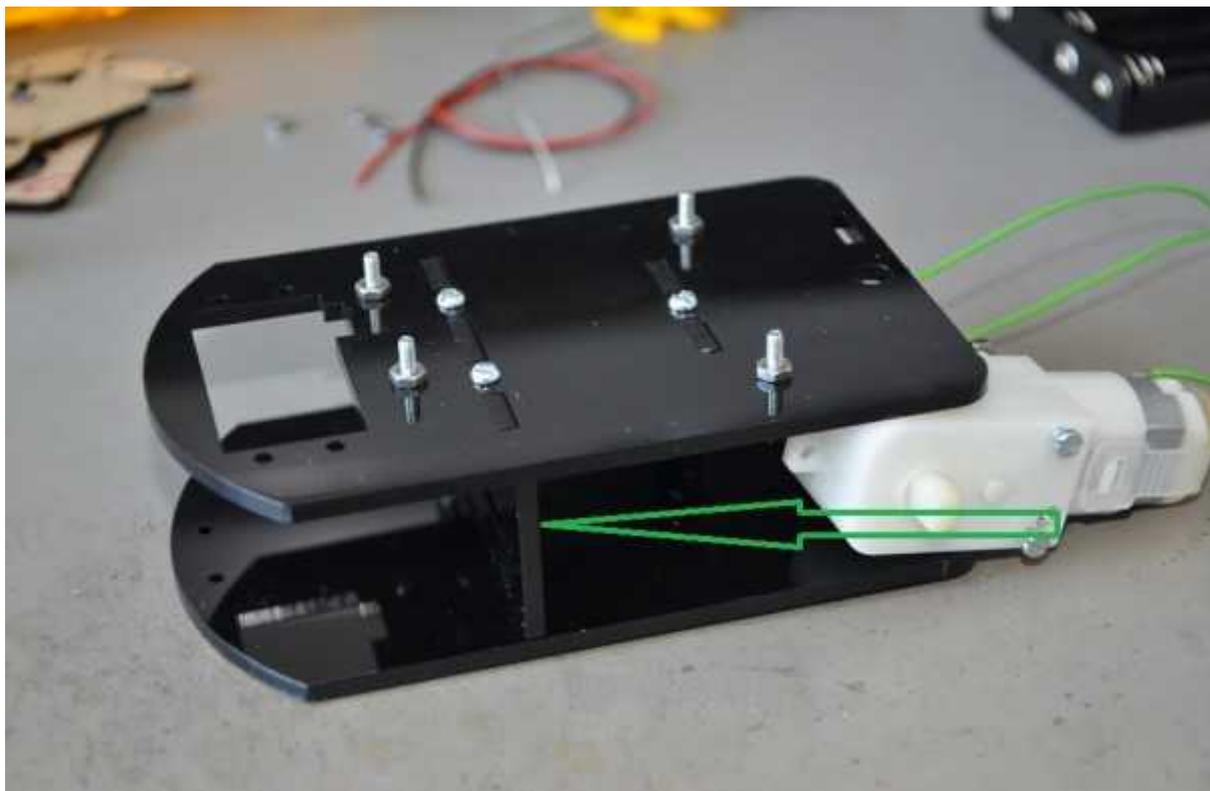
8. Setze die Schrauben von der Seite ein, wo sich die Antriebsachse befindet und auf der anderen Seite werden die Muttern aufgeschraubt. Ziehe die Muttern aber nicht fest. Nur ca. 2 Umdrehung, dass sie nur auf der Schraube drauf sind.



9. Jetzt machen wir mal ein bisschen Rauch :-)
Schneide den Draht in 4 Stücke, entferne die Isolierung an den Enden und löte die Drähte an die Motoranschlüsse.



10. Jetzt werden die Motoren eingebaut. Drücke die Schrauben vorher bis zum Anschlag an, und dann vorsichtig die Motoren von hinten einschieben. Wenn das etwas schwer geht, dann wackle mit den Motoren etwas hin und her, damit die Muttern in den Fassungen einrasten. Auch hier kommt das 90° Prinzip zum Einsatz, mit dem einzigen Unterschied, dass statt einer zweiten Platte die Motoren verschraubt werden. Wenn die Motoren eingeschoben worden sind, ziehe die Schrauben vorsichtig fest. Bitte nicht zu viel Kraft anwenden - ganz leicht anziehen und es hält.

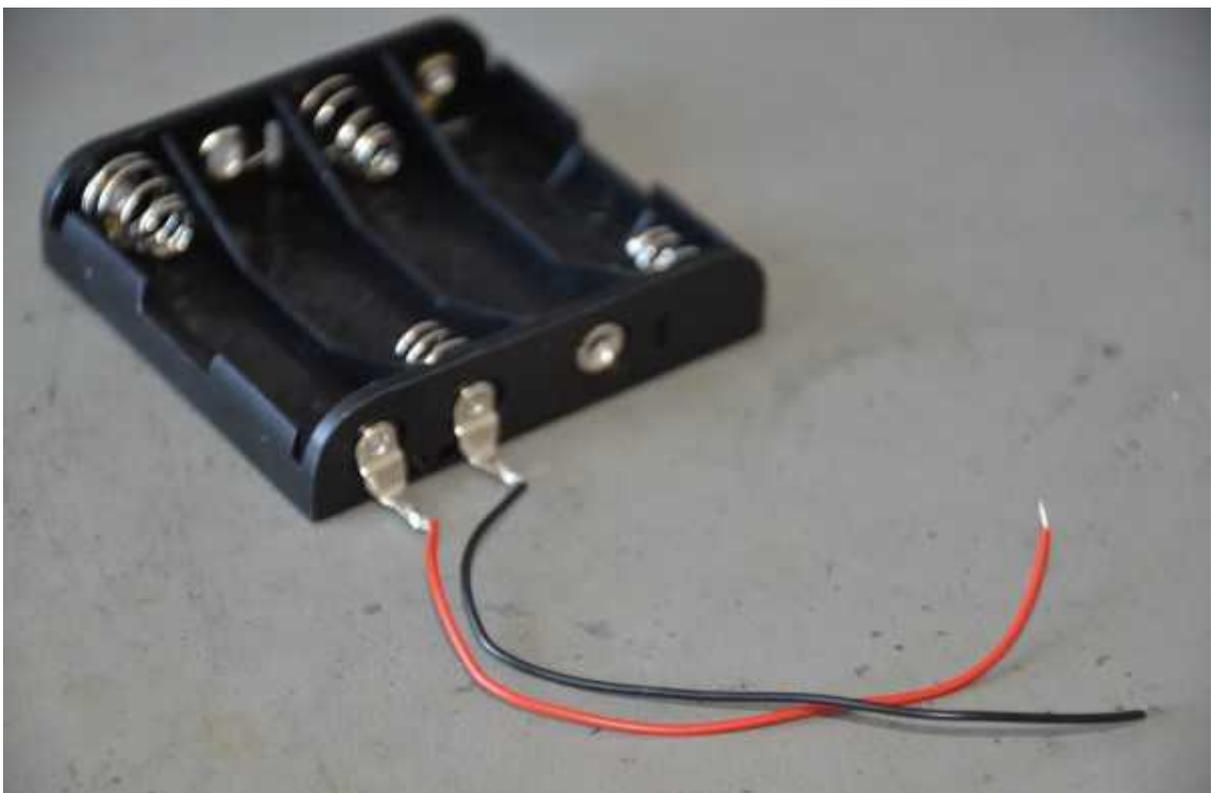


11. Im nächsten Schritt werden etwas kleinere Schrauben und Muttern verwendet. 2 Stück M2x10 Schrauben und M2 Muttern.

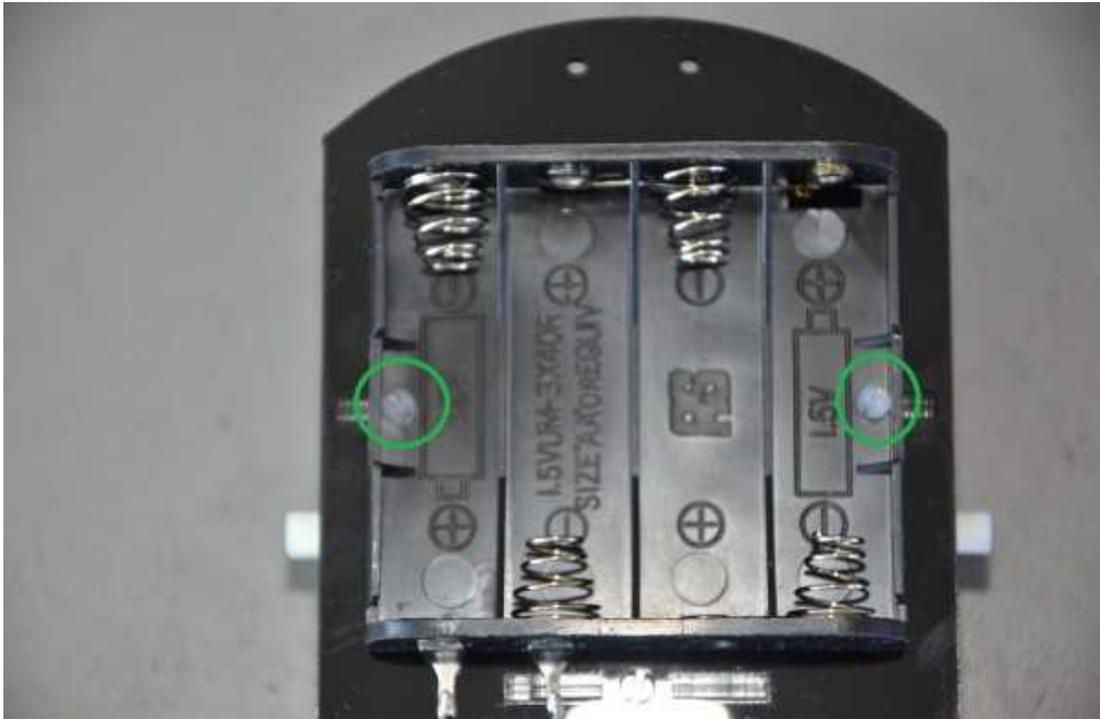
Das Einsetzen der Muttern könnte evtl. aufwendig werden. Hier ist eine kleine Pinzette von Vorteil.



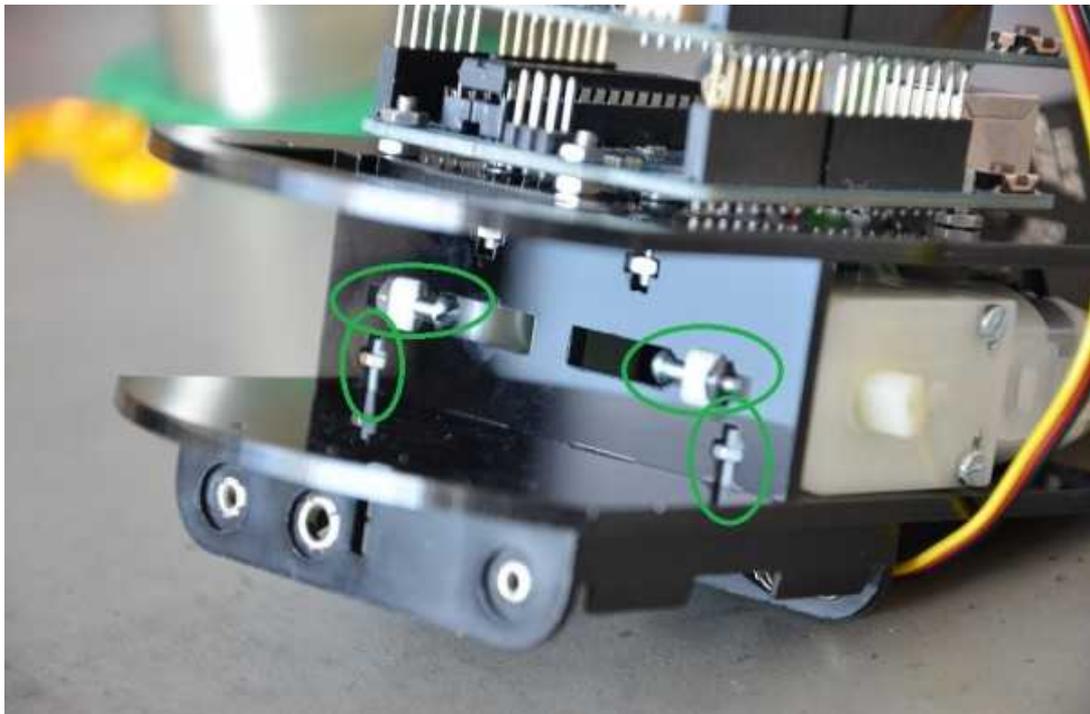
12. Wenn ein Batteriehälter, mit abstehen Lötfähnchen verwendet wird, müssen die Fähnchen erst mal um 90° gebogen werden. Dann die zwei Drähte wie auf dem Bild abisolieren und anlöten.



13. Die M2x10 Schrauben werden wie im Bild eingesetzt, und der Batteriehalter wird unten am Chassis befestigt.



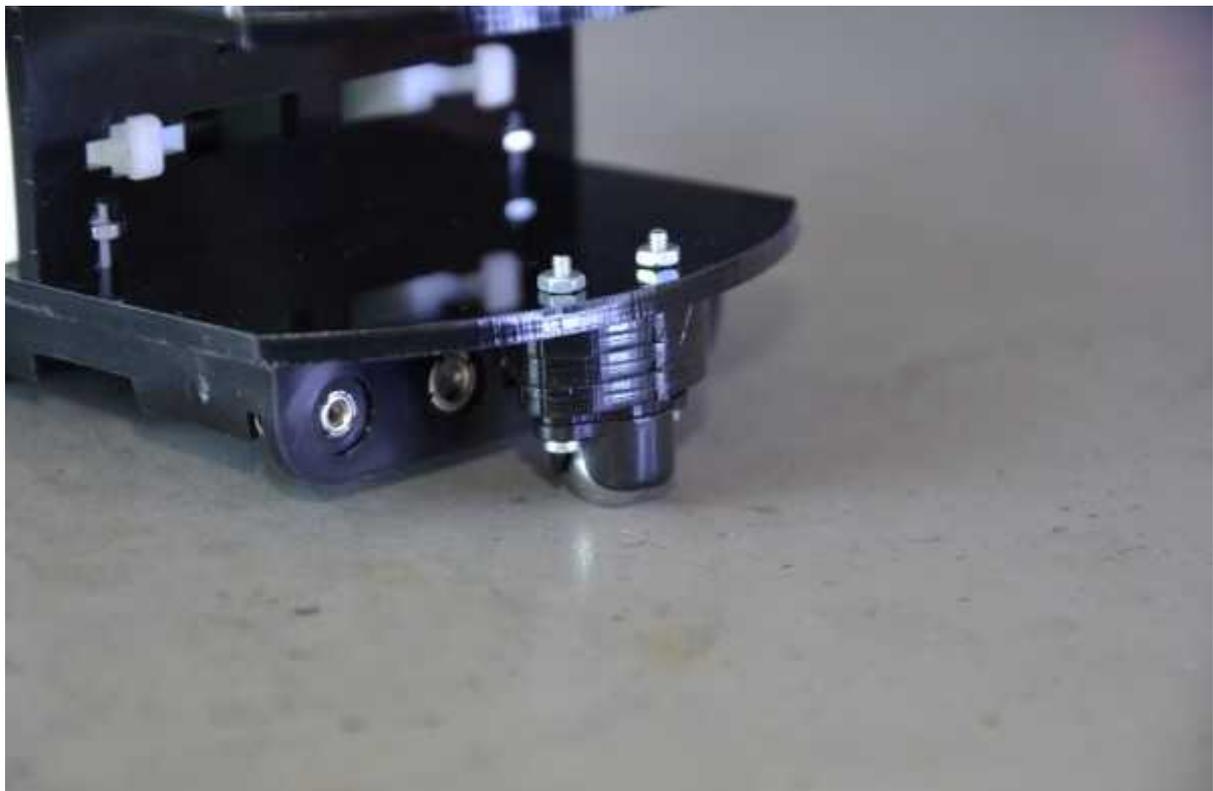
14. So sieht es dann aus. Und da wir schon bei der Ansicht sind, können wir die Motoren auch vorne befestigen. Ok, ok – erwischt! :-)) Hier wurde ein bisschen geschummelt und ist erst beim Schreiben des Artikels aufgefallen, dass der Schritt mit den Muttern nicht fotografiert wurde. :-))



15. Jetzt benötigen wir den Ball Caster bzw. das Stützrad, die Abstandshalter und die Schrauben dafür. Bei dem Ball Caster selbst sind schon zwei Abstandshalter dabei – ein dünner und ein dicker. Wir nehmen davon den dünneren. Schrauben sind M2x18 und Muttern sind M2



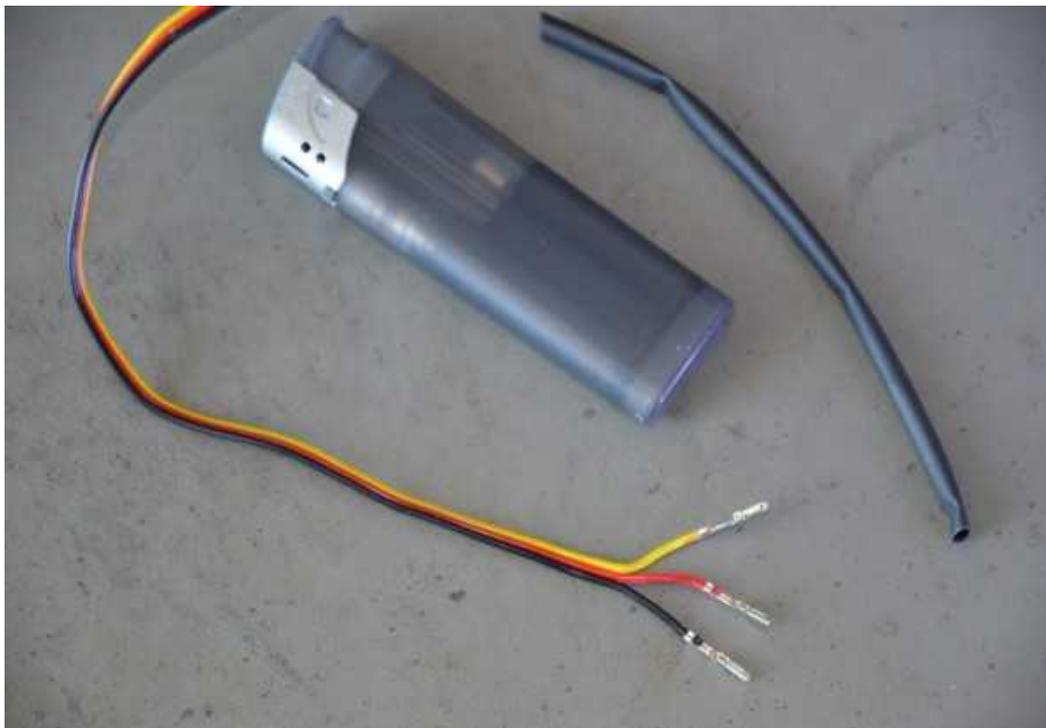
16. Jetzt wird der ganze "Sandwich" wie im Bild verschraubt ;-)



17. Jetzt kommen wir zum Servo. Am Stecker sind drei kleine Näschen, die die Kontakthülsen im Steckergehäuse festhalten. Mit einem kleinen Messer oder besser ganz kleinem Schraubendreher, kann man diese Näschen leicht anheben und die Drähte herausziehen. Bitte ganz vorsichtig mit dem Messer umgehen.



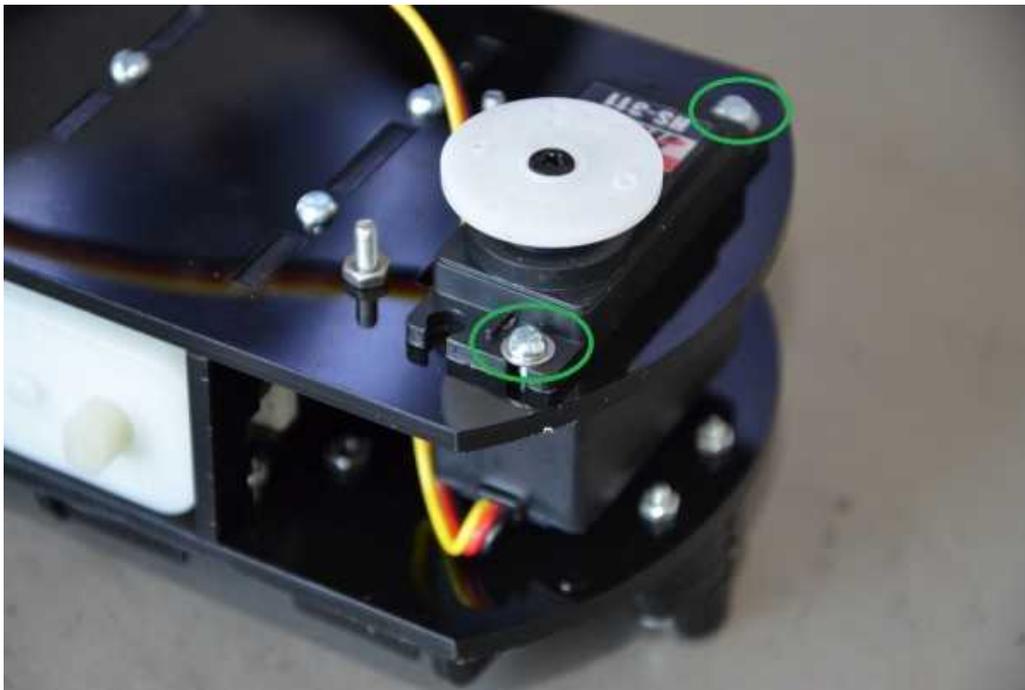
18. Nach der Aktion mit dem Steckergehäuse, müssen wir auf die Hülsen etwas Schrumpfschlauch drüber ziehen, damit kein Kurzschluss entsteht, wenn wir sie nebeneinander stecken.



19. Schneide drei kleine Stückchen Schrumpfschlauch, setze sie auf die Hülsen auf und halte es kurz über einem Feuerzeug. Der Schlauch schrumpft dann zusammen und rutscht nicht mehr so leicht ab. Daher auch der Name – Schrumpfschlauch schrumpft eben ;-)



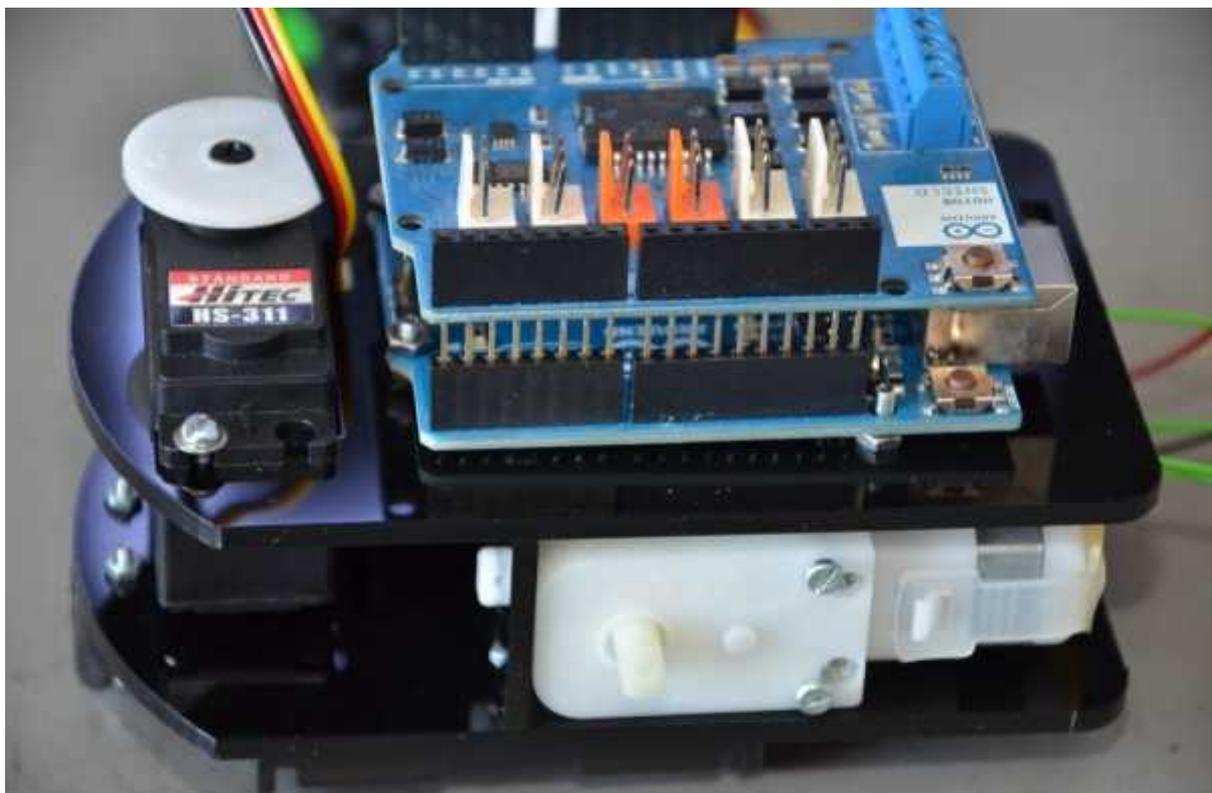
20. Nun wird der Servo auf dem Chassis befestigt. Außer 2 Stück M2,5×10 Schrauben und zwei M2,5 Muttern, brauchen wir hier noch zwei Unterlegscheiben (M2,5). Die Leitung zum Servo soll wieder oben rauskommen. Dort ist extra ein kleiner Schlitz. (Siehe kurz Schritt 21.)



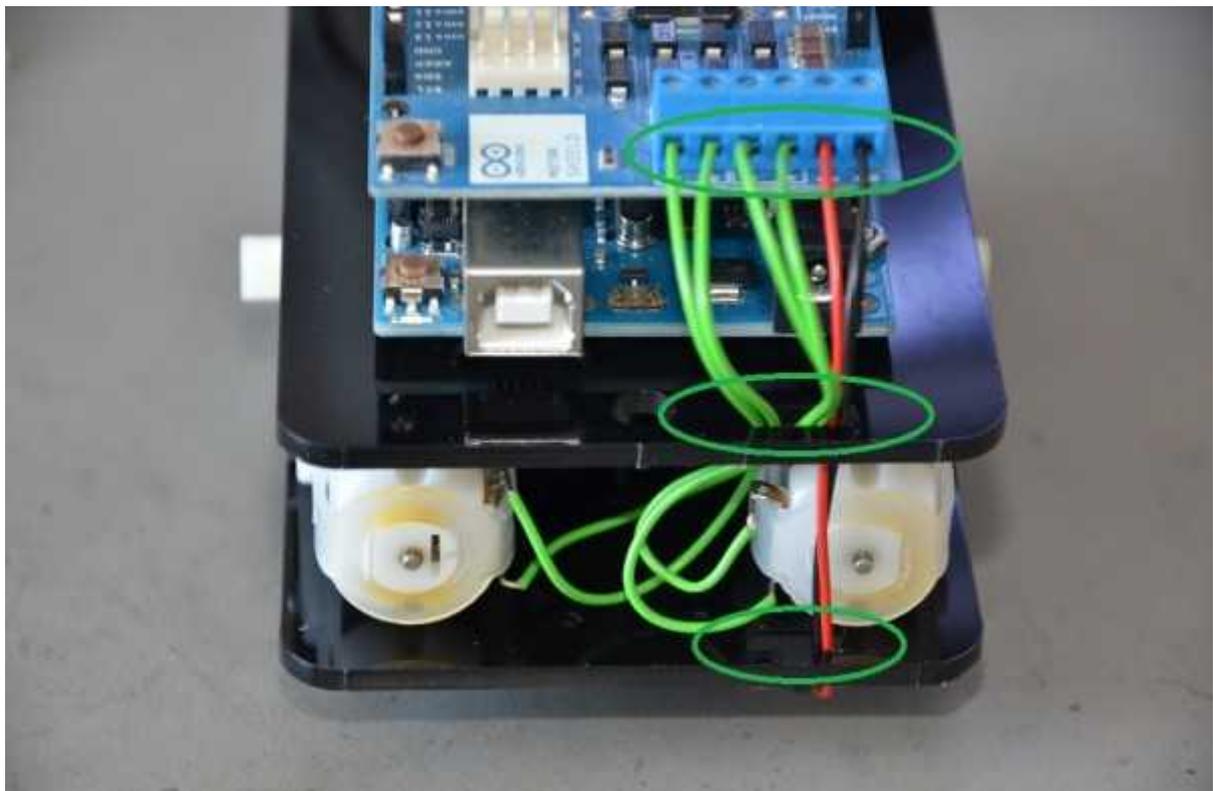
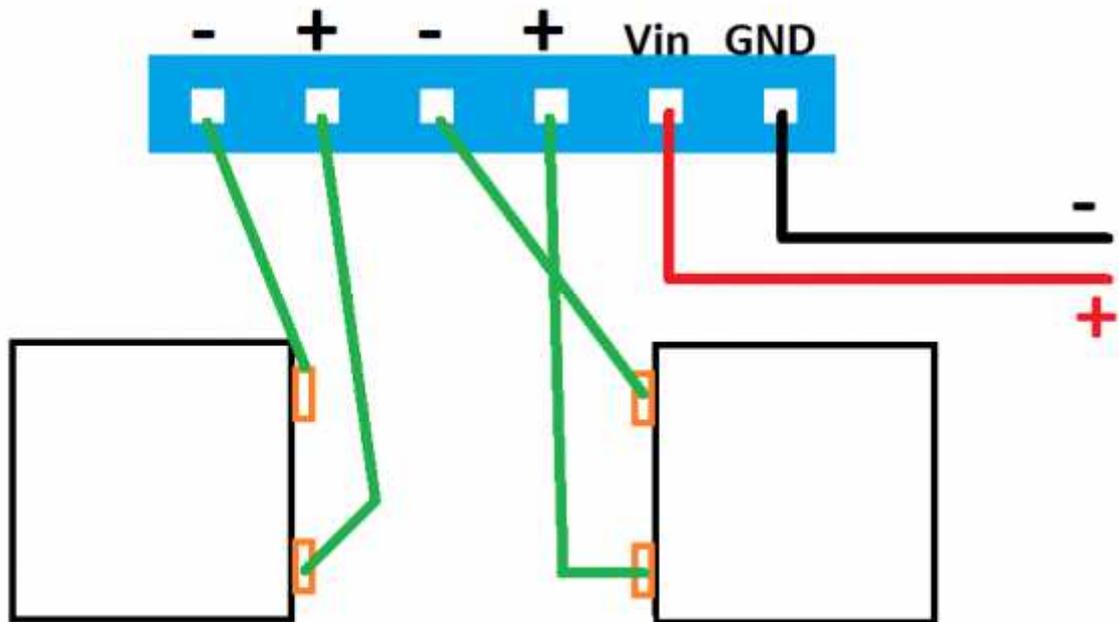
21. Im nächsten Schritt befestigen wir den Arduino mit Hilfe von drei Muttern. Die Stellen sind mit grünen Kreisen markiert, Eine 4. Mutter passt leider nicht drauf, weswegen die Schraube dort einfach nur als Stütze dient (mit Kreuz markiert)



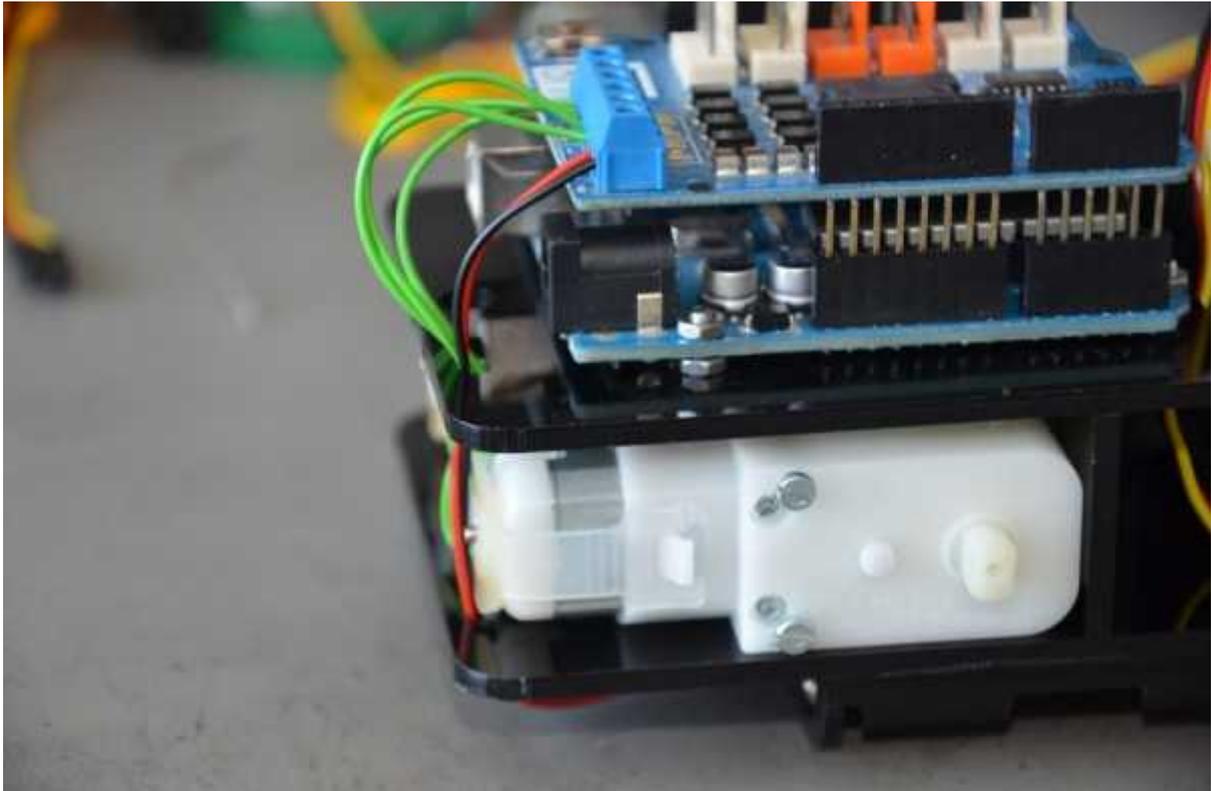
22. Motor Shield aufstecken. Hier sollte man vorsichtig sein, damit die Beinchen nicht verbogen werden und auch nicht daneben gesteckt werden.



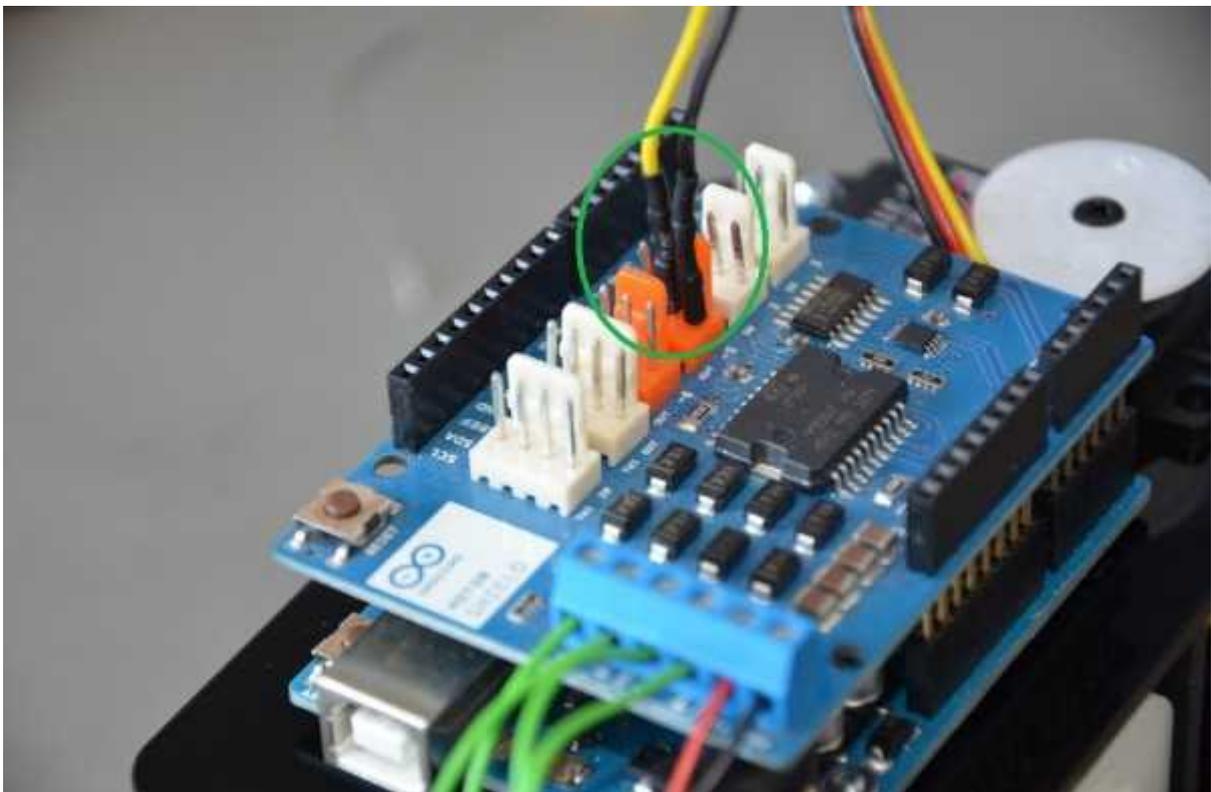
23. Jetzt verbinden wir die Motoren und den Batteriehalter mit dem Motor Shield. Die Leitungen werden durch die Öffnungen nacheinander durchgeschleift und angeschlossen. Um die Verdrahtung etwas klarer darzustellen, habe ich eine kleine Skizze erstellt.



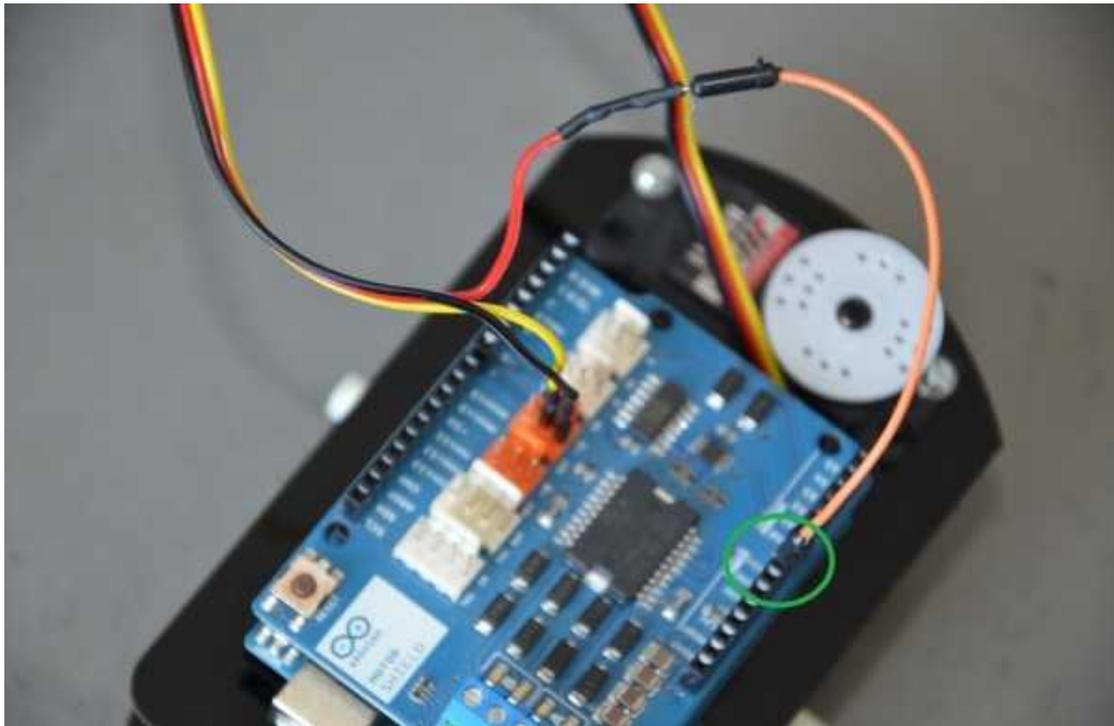
Die Ansicht von der Seite.



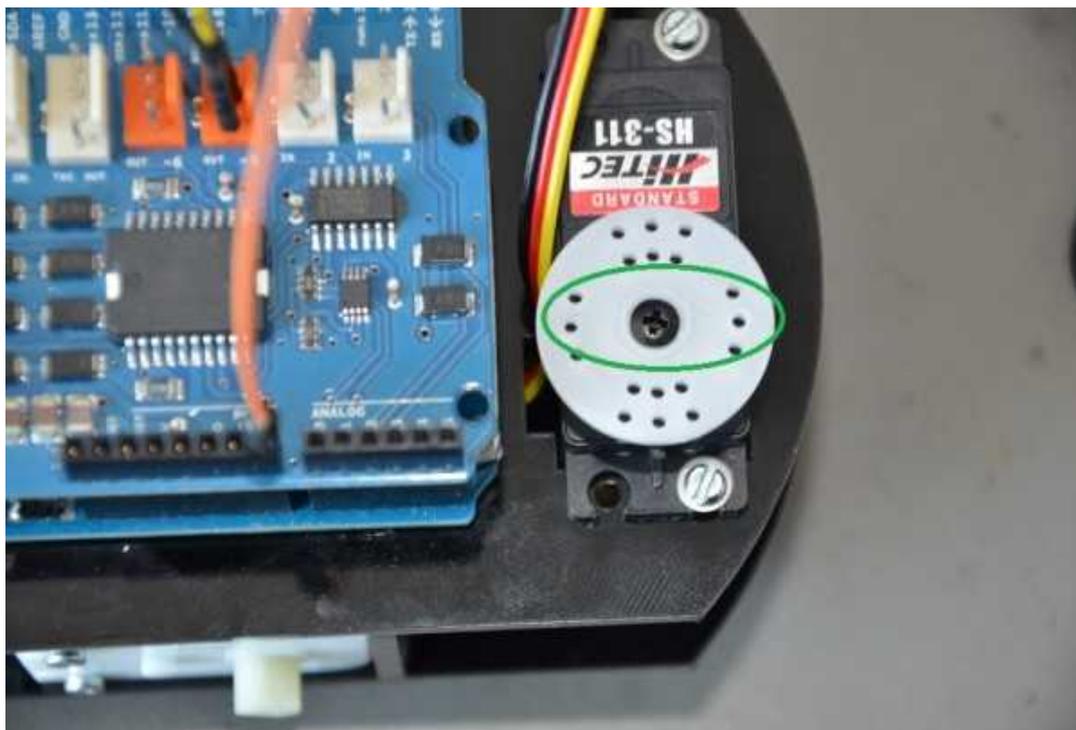
24. Die gelbe (Signal) und schwarze (minus/GND) Leitung des Servos, an OUT 5 vom Motor Shield anschließen (siehe Bild)



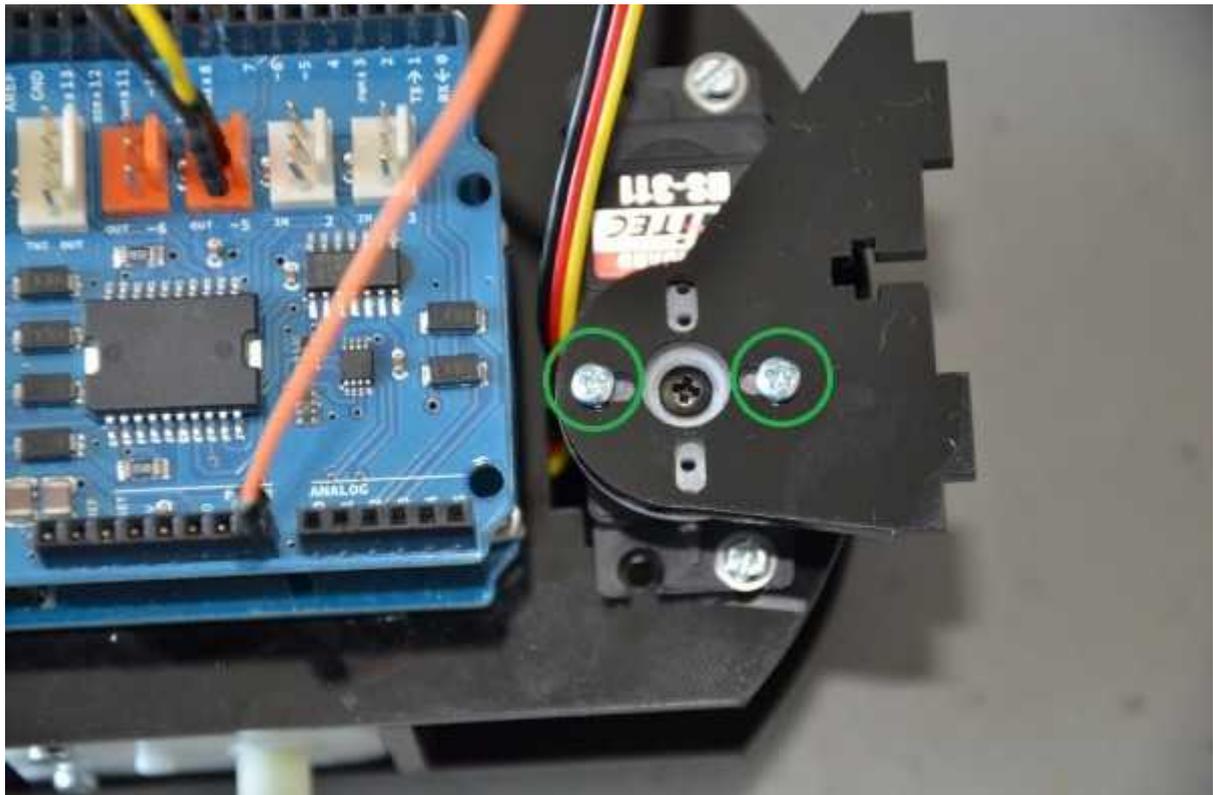
25. Die rote Leitung (plus/Vcc) wird mit Hilfe eines anderen Jumper Wires mit Vin verbunden.



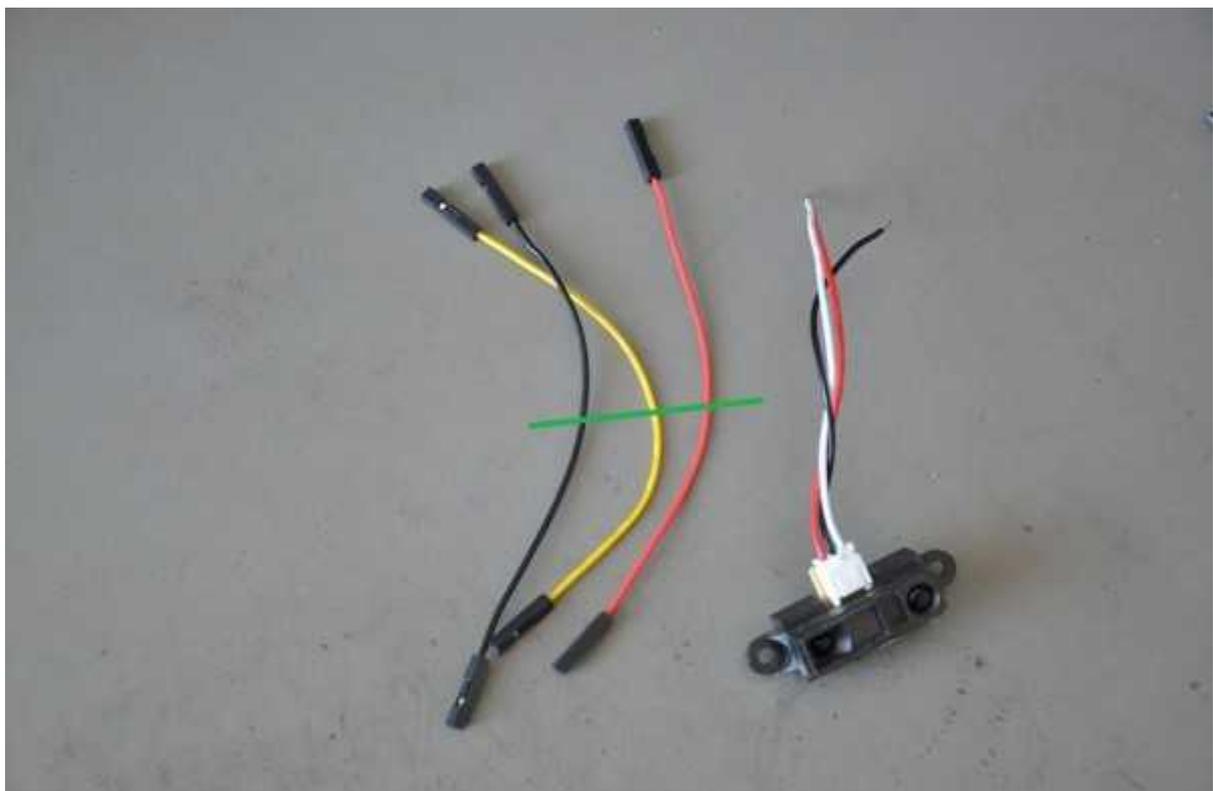
26. Jetzt nähern wir uns so langsam dem Kopf des Roboters. Spätestens jetzt, empfehle ich euch die schwarze Schraube am Servo zu entfernen. Der Grund ist einfach der, dass wir den Servo erst mal mittig stellen müssen, damit der Kopf in Mittelstellung des Servos auch geradeaus guckt. Um den Servo mittig zu stellen, müssen wir zuerst das Programm auf den Roboter übertragen. Soweit sind wir aber noch nicht (kommt noch). Ohne diese Schraube können wir den Kopf abziehen, den Servo in den richtigen Winkel bringen und den Kopf wieder aufsetzen. Erst dann kommt die Schraube wieder rein.

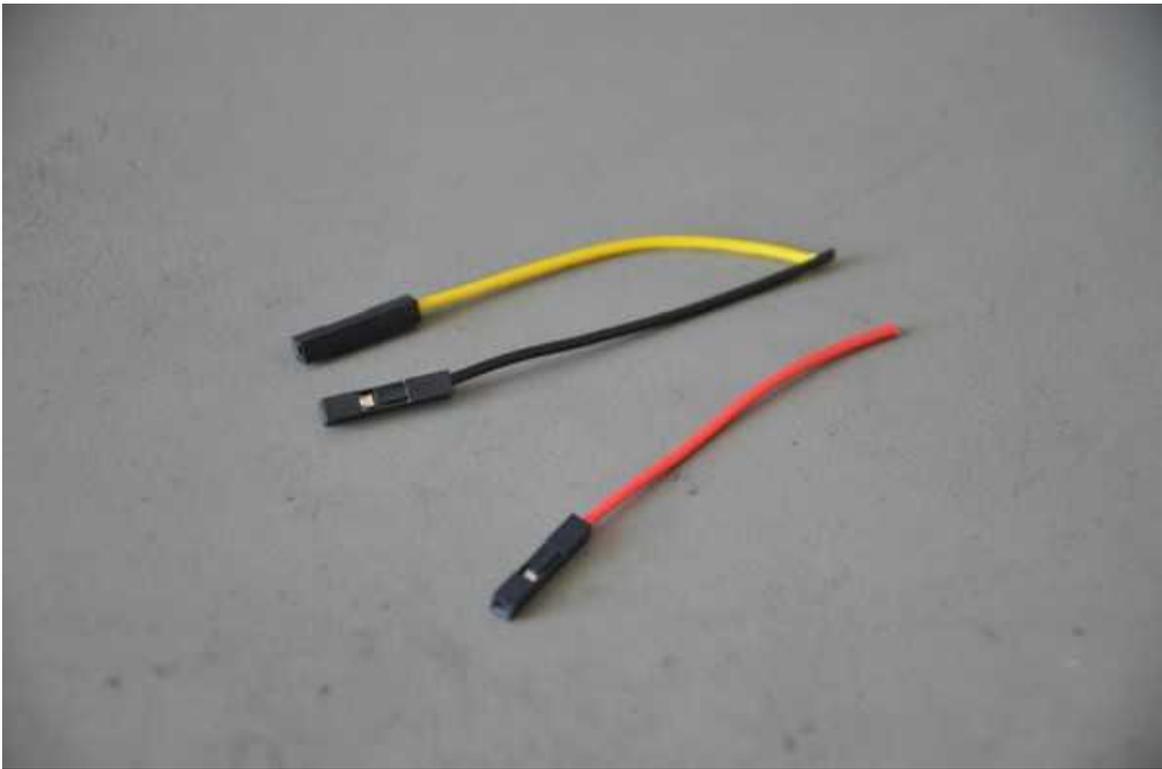


27. Das untere Teil des Kopfes wird mit zwei kleinen Blechschrauben befestigt. Diese Schrauben sind spitz an den Enden.

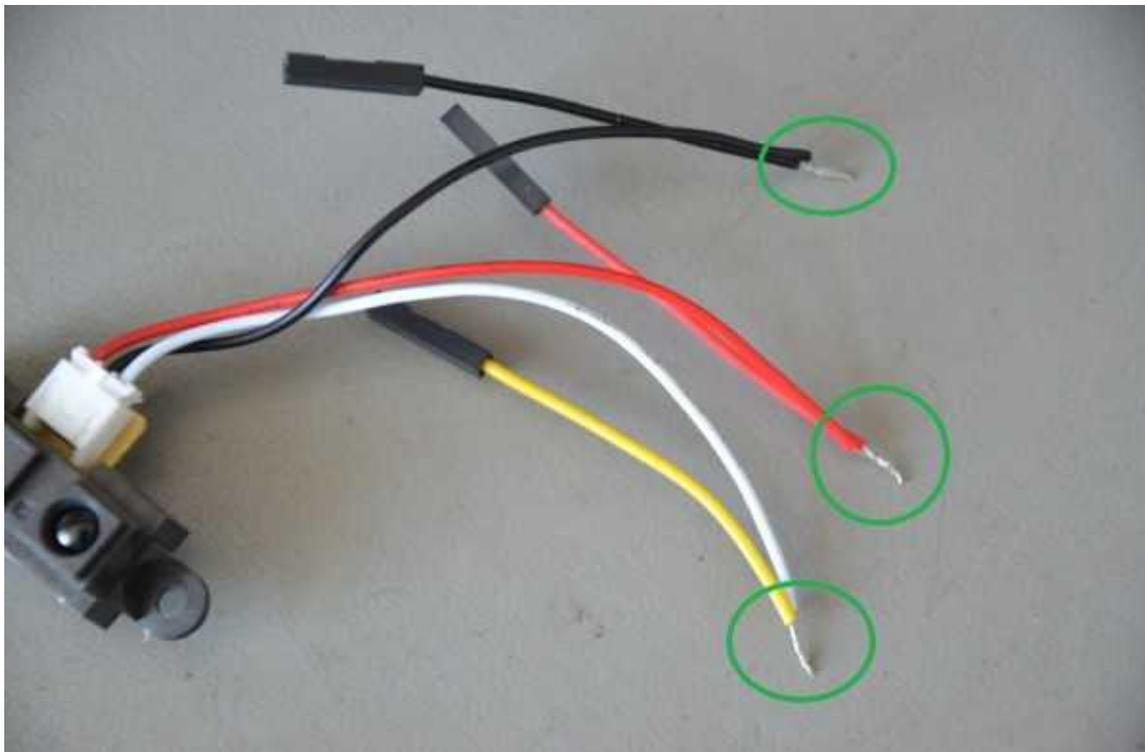


28. Und jetzt bauen wir dem Roboter die Augen zusammen. Nimm drei Jumper Wires und schneide sie in der Mitte durch. Ich hatte hier keine weißen Jumper Wires, also habe ich ein gelbes statt weiß genommen.

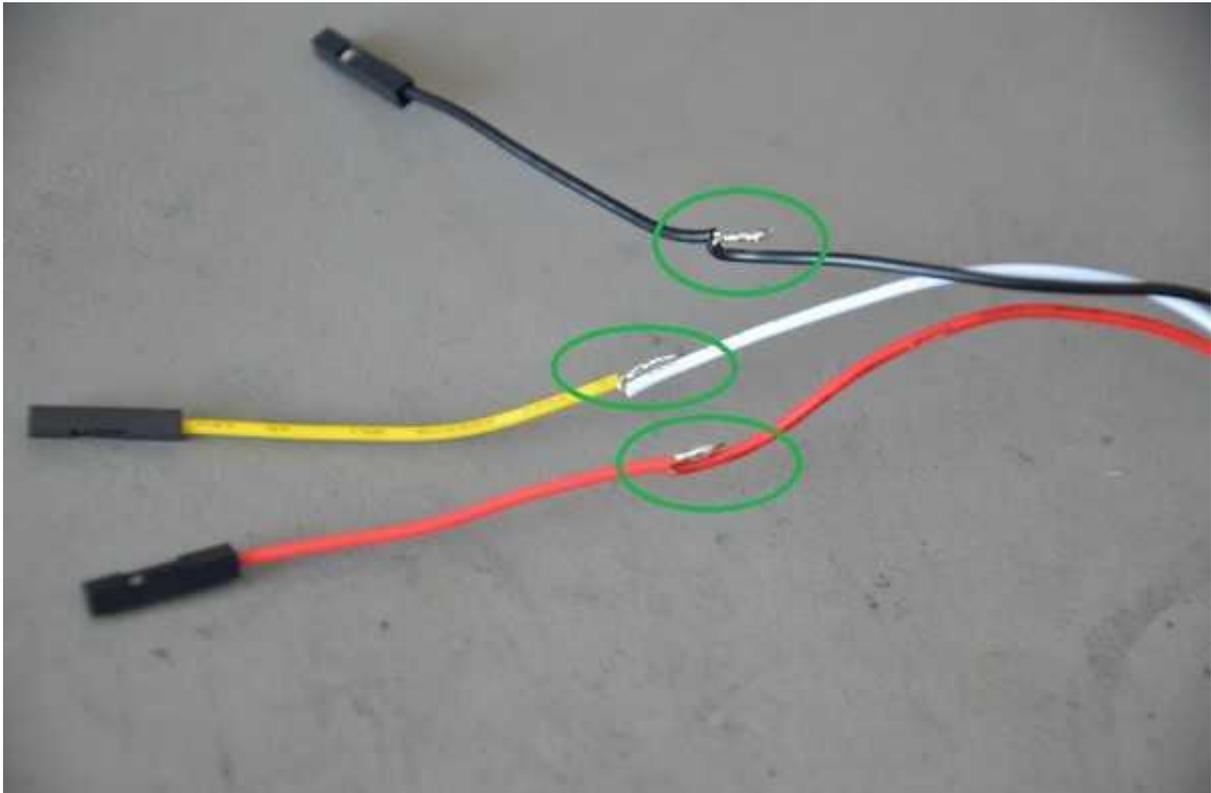




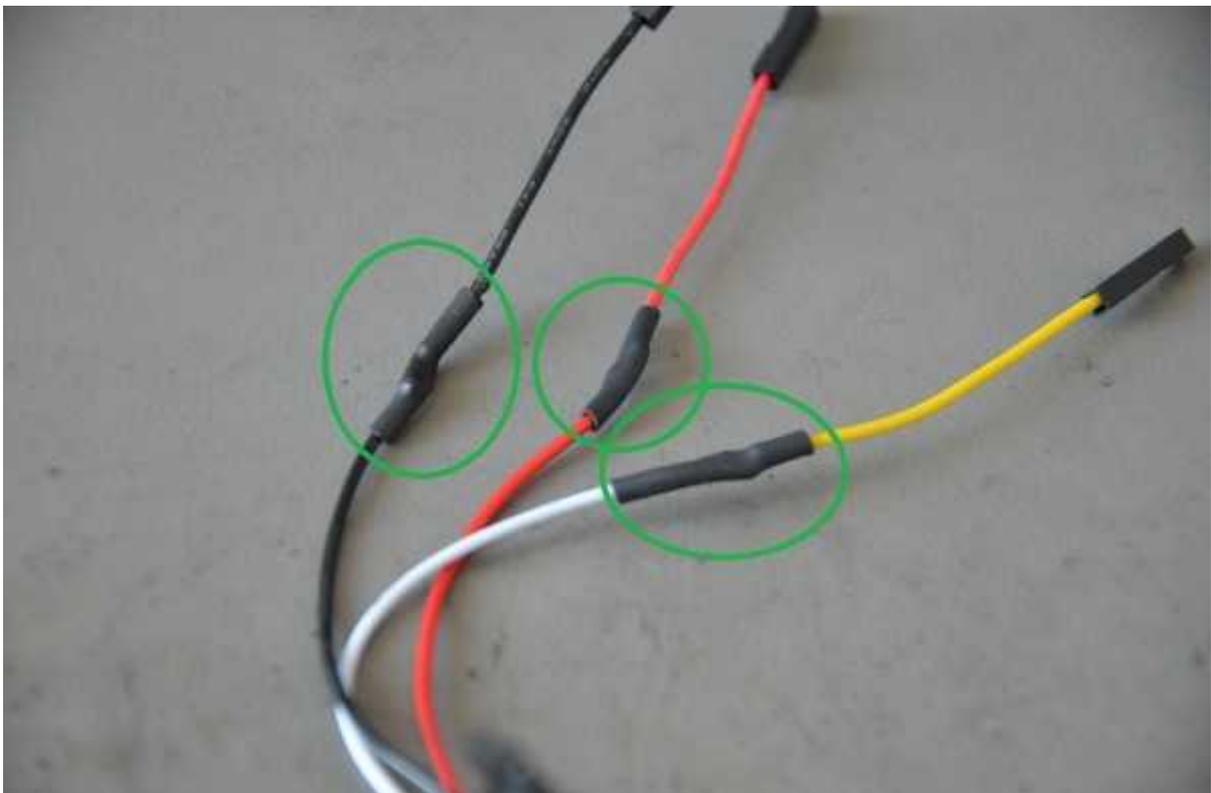
29. Die Jumper Wires werden an den abgeschnittenen Enden abisoliert und mit den Anschlussdrähten des Sensors verdrillt. Und damit das Ganze auch bombenfest sitzt, werden die drei Stellen noch gelötet.



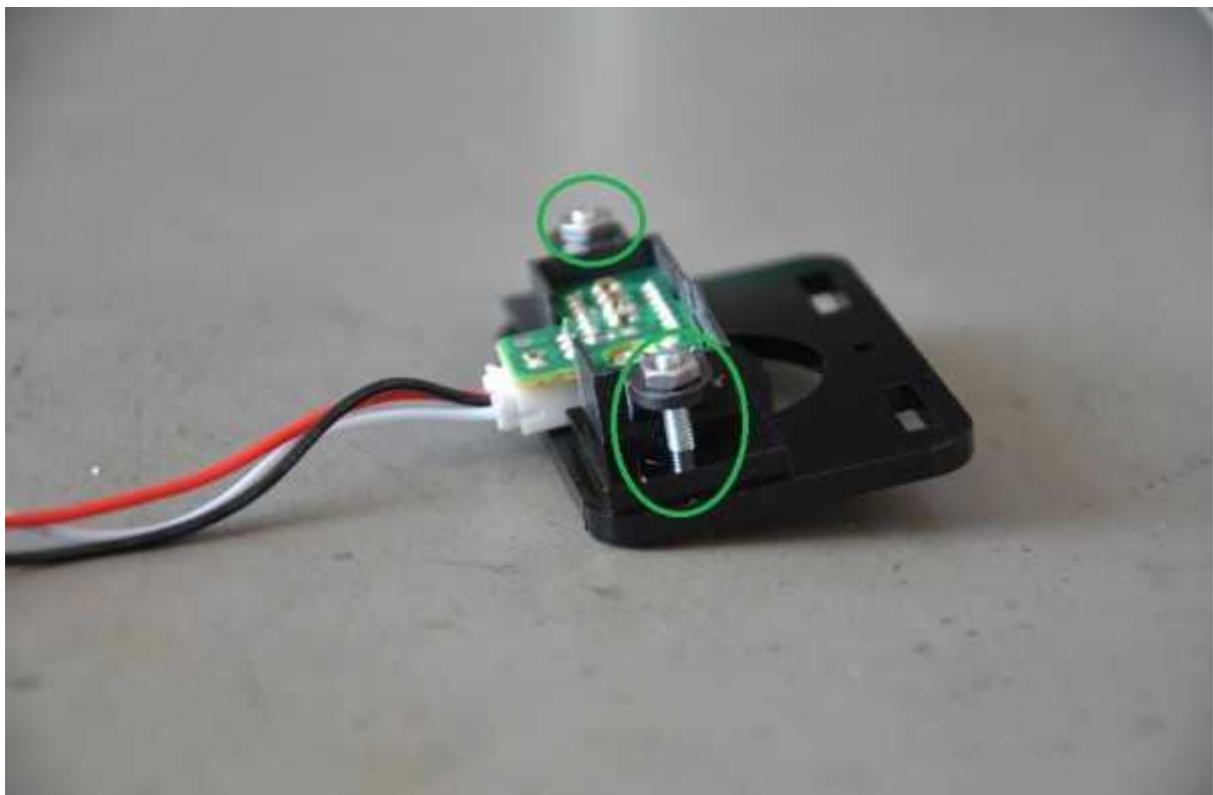
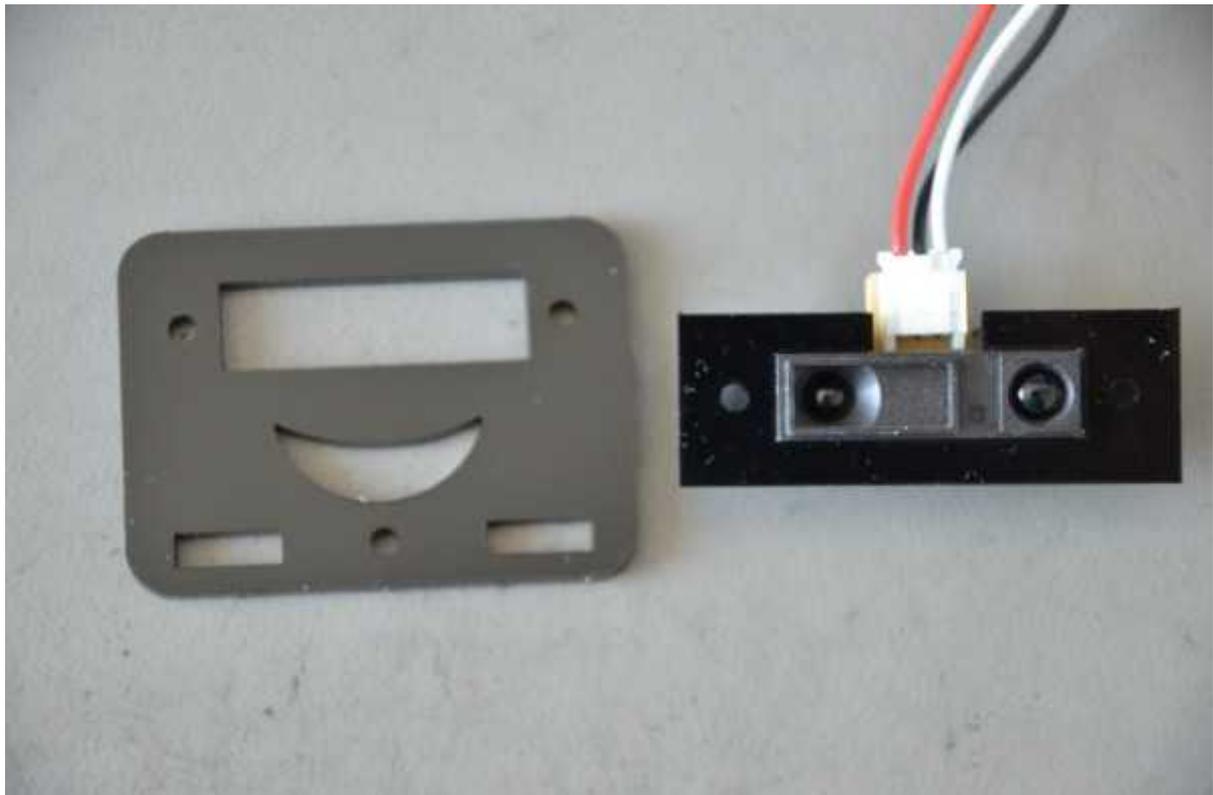
30. Puuuhhh. Schritt Nummer 30. Bald haben wir es. Biege die überstehenden nackten Spitzen parallel zu den Leitungen...



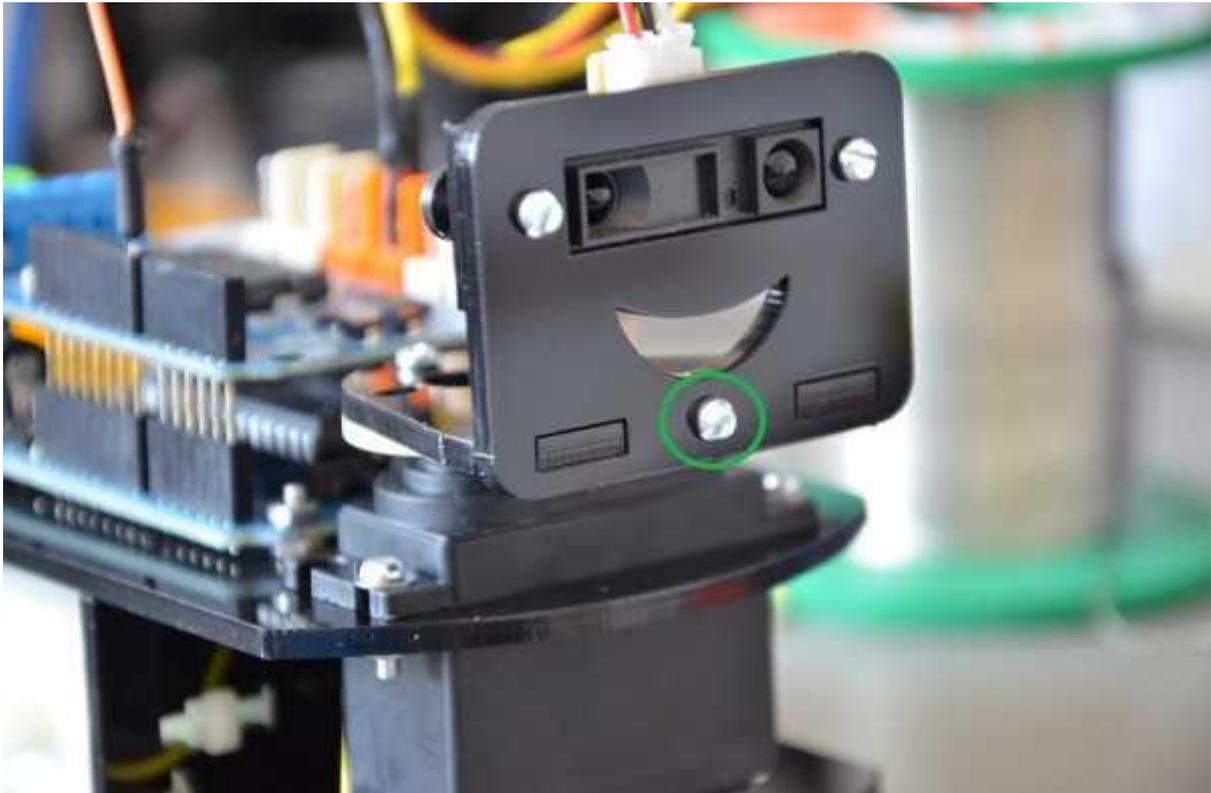
... und ziehe Schrumpfschlauch drüber. Die Schrumpfschläuche dann wieder kurz über einem Feuerzeug halten.



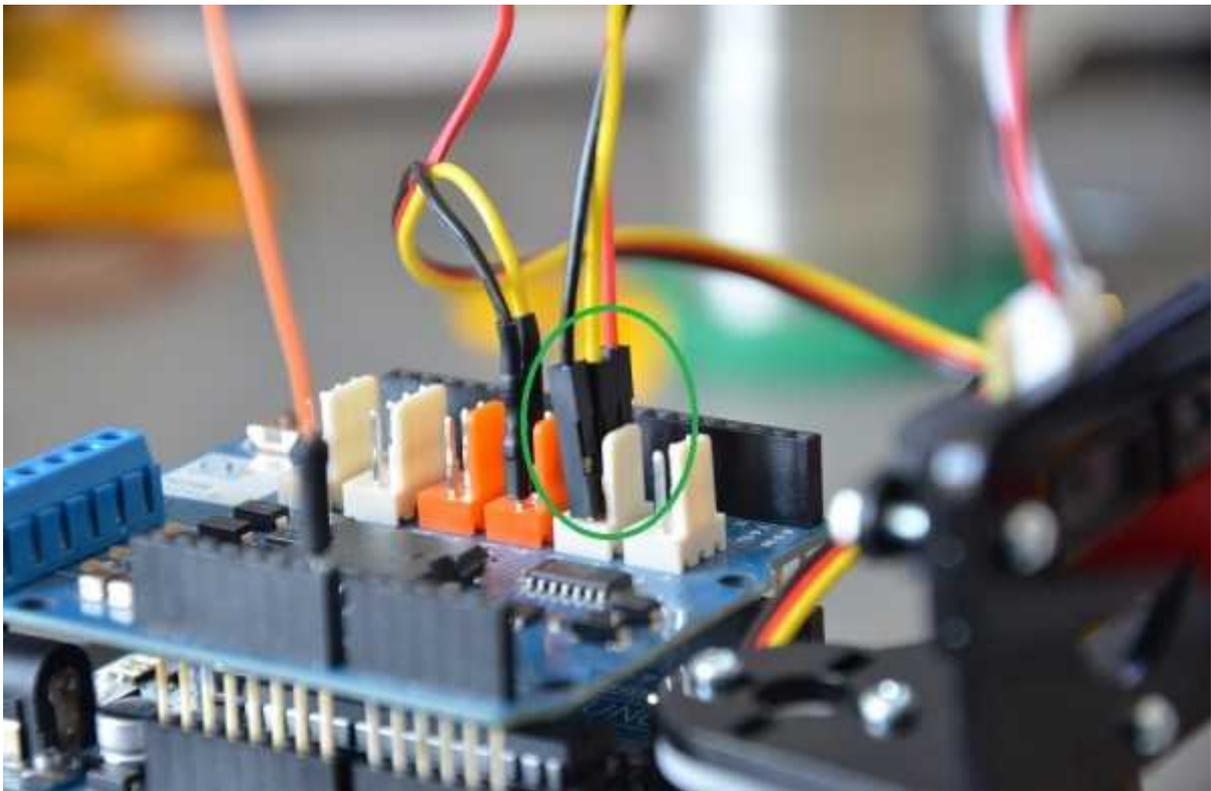
31. Jetzt nehmen wir das "Gesicht" des Roboters und einen Abstandshalter für den IR-Distanzmesser und verschrauben das Ganze.



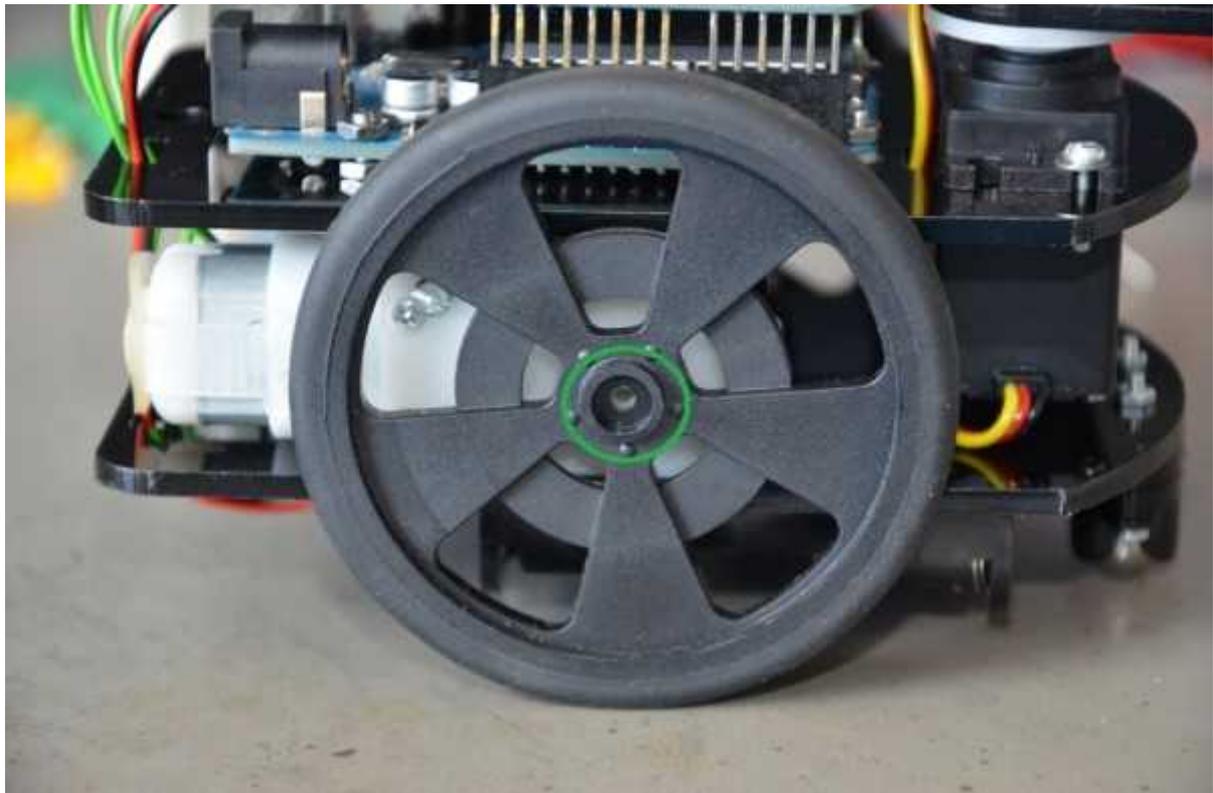
32. Jetzt wird das “Gesicht” am Servo befestigt. Wieder die Schraube M2,5×10 und M2,5 Mutter und wieder das übliche Verfahren, mit der 90° Verschraubung



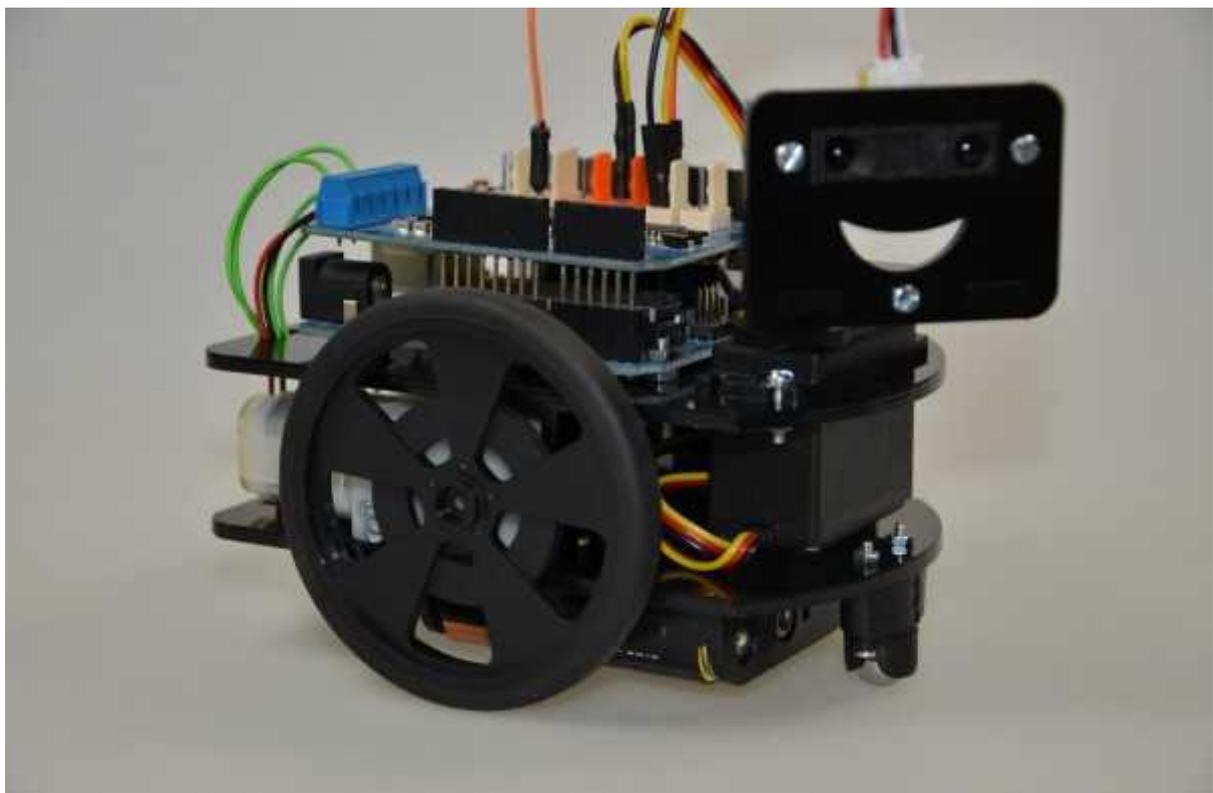
33. Nun werden die Leitungen des Sensors wie im Bild, an IN 2 des Motor Shields angeschlossen.



34. Und damit der Roboter auch fährt und nicht nur steht, machen wir die Räder dran. Man kann diese noch zusätzlich mit Schrauben befestigen. Meistens sitzen die Räder fest genug, sodass man diese Schrauben gar nicht braucht.



35. TAAADAAAAA! Die Mechanik und die Elektronik hast du jetzt aufgebaut. Als nächstes kommt die Programmierung.



Das Programm

```
/*
**
** Version: 1.0
** Datum: 15.02.2013
**
** Dies ist der Programmcode für das Starter Kit von Florian H.
**
** Weiter-Entwicklung erwünscht! Aber bitte erzählt uns davon... ;-)
**
*/

// Nützliche Funktionen für die Servo Ansteuerung kommen von hier:
#include <Servo.h>

// Welcher Pin hat welche Funktion???
//      Funktion  Pin  Beschreibung
#define IR_SENSOR 2    // Der Sharp IR Sesor ist an Pin 2
#define DIR_B     13   // Richtung B
#define DIR_A     12   // Richtung A
#define PWM_B     11   // Geschwindigkeit B
#define BRAKE_A   9    // Bremse A
#define BRAKE_B   8    // Bremse B
#define PWM_A     3    // Geschwindigkeit A
#define SERVO     5    // Servo wird an Pin 5 angesteuert

// Nützliche Einstellungen:
#define FULL_SPEED      128          // Vollgas ist 255, höher geht nicht.
#define TURN_SPEED      255          // schnell Drehen
#define LEFT             LOW         // Links drehen bei LOW
#define RIGHT            HIGH        // ... und Rechts bei HIGH
// Bei diesem Wert am IR Sensor soll der Roboter anhalten
```

```

// --> Ist der Wert KLEINER hält der Roboter FRÜHER an
#define CLOSEST_DISTANCE 350
// Mit welchem Winkel stößt der Sensor auf dem Servo rechts und links an?
#define SERVO_RIGHT      55           // rechts
#define SERVO_LEFT      125          // links

Servo SensorServo; // Mit diesem Element wird der Servo gesteuert

// An welcher Servo Position ist die Distanz zur Gefahr am geringsten?
int SeekingPositionWithClosestDanger()
{
    // Erst mal anhalten
    digitalWrite(BRAKE_A, HIGH);
    digitalWrite(BRAKE_B, HIGH);
    int ServoPosition;
    int MinimumDistance = 0;
    int MinimumServoPosition = 0;

    // Von Rechts (SERVO_RIGHT) nach Links (SERVO_LEFT) alle Servo-Positionen anfahren.
    // mit ServoPosition++ wird Wert immer um 1 erhöht
    for(ServoPosition = SERVO_RIGHT; ServoPosition <= SERVO_LEFT; ServoPosition++)
    {
        // Servo Wert einstellen
        SensorServo.write(ServoPosition);
        delay(10);
        // ist der aktuelle Wert näher, als der minimale Wert?
        if(analogRead( IR_SENSOR ) > MinimumDistance )
        {
            // Ja: aktueller Wert ist neues Minimum
            MinimumDistance = analogRead( IR_SENSOR );
            // Außerdem merken wir uns die Servo Position
            MinimumServoPosition = ServoPosition;
        }
    }
}
// Die gefundene Position wieder einstellen und Wert zurückgeben

```

```

    SensorServo.write(MinimumServoPosition);
    return MinimumServoPosition;
}

byte ServoPosition = 90;
boolean TurnServo = RIGHT;
// Vorwärts fahren und dabei den Sensor hin und her schwenken
void DriveForward()
{
    SensorServo.write( ServoPosition );
    //Beide Motoren auf Geradeaus stellen, ...
    digitalWrite( DIR_A, HIGH );
    digitalWrite( DIR_B, HIGH );

    // ... Vollgas und ...
    analogWrite( PWM_A, FULL_SPEED );
    analogWrite( PWM_B, FULL_SPEED );

    // ..Bremsen lösen!
    digitalWrite( BRAKE_A, LOW );
    digitalWrite( BRAKE_B, LOW );

    // Dreht sich der Servo nach Rechts?
    if( TurnServo == LEFT )
        ServoPosition = ServoPosition+1; // Weiter nach Rechts, Wert um 1 erhöhen
    if( TurnServo == RIGHT )
        ServoPosition = ServoPosition-1; // Sonst nach Links, 1 vom Wert abziehen

    // Hat der Servo das Linke Ende erreicht?
    if( ServoPosition > SERVO_LEFT )
        TurnServo = RIGHT; // Jetzt nach Rechts drehen
    if( ServoPosition < SERVO_RIGHT )
        TurnServo = LEFT; // Sonst nach Links drehen...
}

```

```

// Drehen! Aber in welche Richtung???
// LEFT für Links (gegen den Uhrzeiger) RIGHT für Rechts (im Uhrzeigersinn)
void Turn( boolean Direction )
{
    // Bremsen
    digitalWrite( BRAKE_A, HIGH );
    digitalWrite( BRAKE_B, HIGH );
    delay( 500 );

    digitalWrite( DIR_A, Direction );    // Motor A in die "RICHTUNG" drehen
    digitalWrite( DIR_B, !Direction );  // Motor B in entgegen der "RICHTUNG" drehen

    // Geschwindigkeit für das Drehen einstellen
    analogWrite( PWM_A, TURN_SPEED );
    analogWrite( PWM_B, TURN_SPEED );

    // Bremsen lösen
    digitalWrite( BRAKE_A, LOW );
    digitalWrite( BRAKE_B, LOW );

    // Solange drehen, bis der Sensor eine 10% weitere Entfernung misst
    while( ( CLOSEST_DISTANCE * 1 ) < analogRead( IR_SENSOR ) )
    {
        delay( 50 );
    }
    // Halt!
    digitalWrite( BRAKE_A, HIGH );
    digitalWrite( BRAKE_B, HIGH );
    delay(1000);
}

// Bevor es los geht... Diese Funktion wird am Anfang genau einmal ausgeführt
void setup( )
{
    // Motor A (rechts) initialisieren

```

```

pinMode( DIR_A, OUTPUT );    // Pin für Richtung Motor A als Ausgang definieren
pinMode( BRAKE_A, OUTPUT ); // Pin für Bremse Motor A als Ausgang definieren
// Motor B (links) initialisieren
pinMode( DIR_B, OUTPUT );    // Pin für Richtung Motor B als Ausgang definieren
pinMode( BRAKE_B, OUTPUT ); // Pin für Bremse Motor B als Ausgang definieren

// Beide Bremsen anziehen, HIGH = Bremsen!
digitalWrite( BRAKE_A, HIGH );
digitalWrite( BRAKE_B, HIGH );

// Servo initialisieren und auf 90° stellen
SensorServo.attach( SERVO );
SensorServo.write( 90 );
delay( 500 );

// Warten bis etwas vor dem Sensor ist
while( CLOSEST_DISTANCE > analogRead( IR_SENSOR ) )
{
    // solange warten...
    delay( 100 );
}

// Los geht's!!!
}

// Der eigentliche Programmablauf.
// Nachdem setup() fertig ist wird die Funktion loop() endlos nacheinander ausgeführt.
void loop( )
{
    int DangerPosition;
    int Distance;

    Distance = analogRead( IR_SENSOR );    // was misst der Sensor?
    if( Distance > CLOSEST_DISTANCE )      // Hindernis = größerer Wert
    {

```

```
// Nochmal alles scannen um zu wissen, wo genau die Gefahr am nächsten ist...
DangerPosition = SeekingPositionWithClosestDanger();

// Gefahr Links?
if( DangerPosition <= 90 )
{
    Turn( RIGHT ); // Rechts drehen
}
// Oder doch Rechts?
if( DangerPosition > 90 )
{
    Turn( LEFT ); // dann Links drehen
}
}
DriveForward(); // Immer gerade aus!!!
delay( 10 );
}
```

FAQ

Frage: Mein Roboter hält immer wieder ohne Grund an, bewegt den Kopf hin und her, versucht wieder anzufahren und bleibt wieder stehen. Woran liegt das?

Antwort: Es liegt entweder an den zu schwachen Batterien oder an dem Servo. Manche Servos ziehen sehr viel Strom, sodass die Spannung kurzzeitig sehr niedrig wird. Dadurch spinnt dann die Elektronik.

Frage: Wieso fährt der Roboter, bei einem Hindernis auf der linken Seite, auch nach Links, obwohl er nach rechts fahren sollte?

Antwort: Das hängt so bisschen mit dem Servo und dem Programmcode zusammen. Die üblichen Servos haben einen Stellwinkel zwischen 0° und 180° . Bei manchen Servos liegen die 180° ganz links und 0° ganz rechts. Und bei manchen Servos genau umgekehrt. Im Programmcode wird abgefragt, wo der Servo sich gerade befindet, also in welche Richtung es gerade guckt. Entsprechend werden dann die Motoren so angesteuert, dass der Roboter sich vom Hindernis weg bewegt.

So wird der Controller von dem Servo manchmal reingelegt und der "Controller" denkt, dass das Hindernis sich links befindet, obwohl das nicht stimmt. Als Lösung sollte man die Drahtpaare der Motoren gegeneinander tauschen. D.h. dort wo der rechte Motor angeschlossen war, wird der linke Motor angeschlossen und dort wo der linke Motor war, wird der rechte Motor angeschlossen.