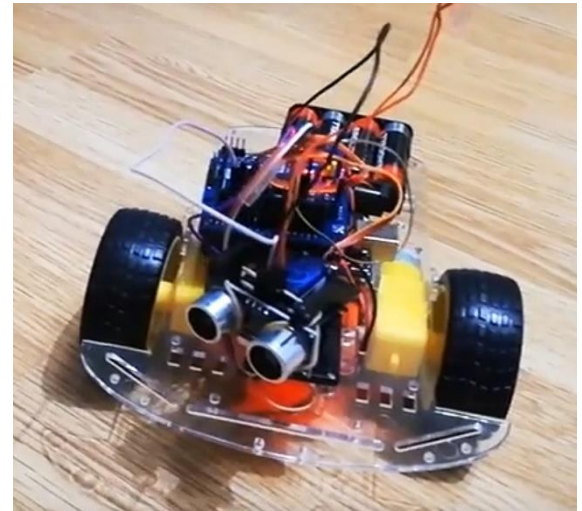


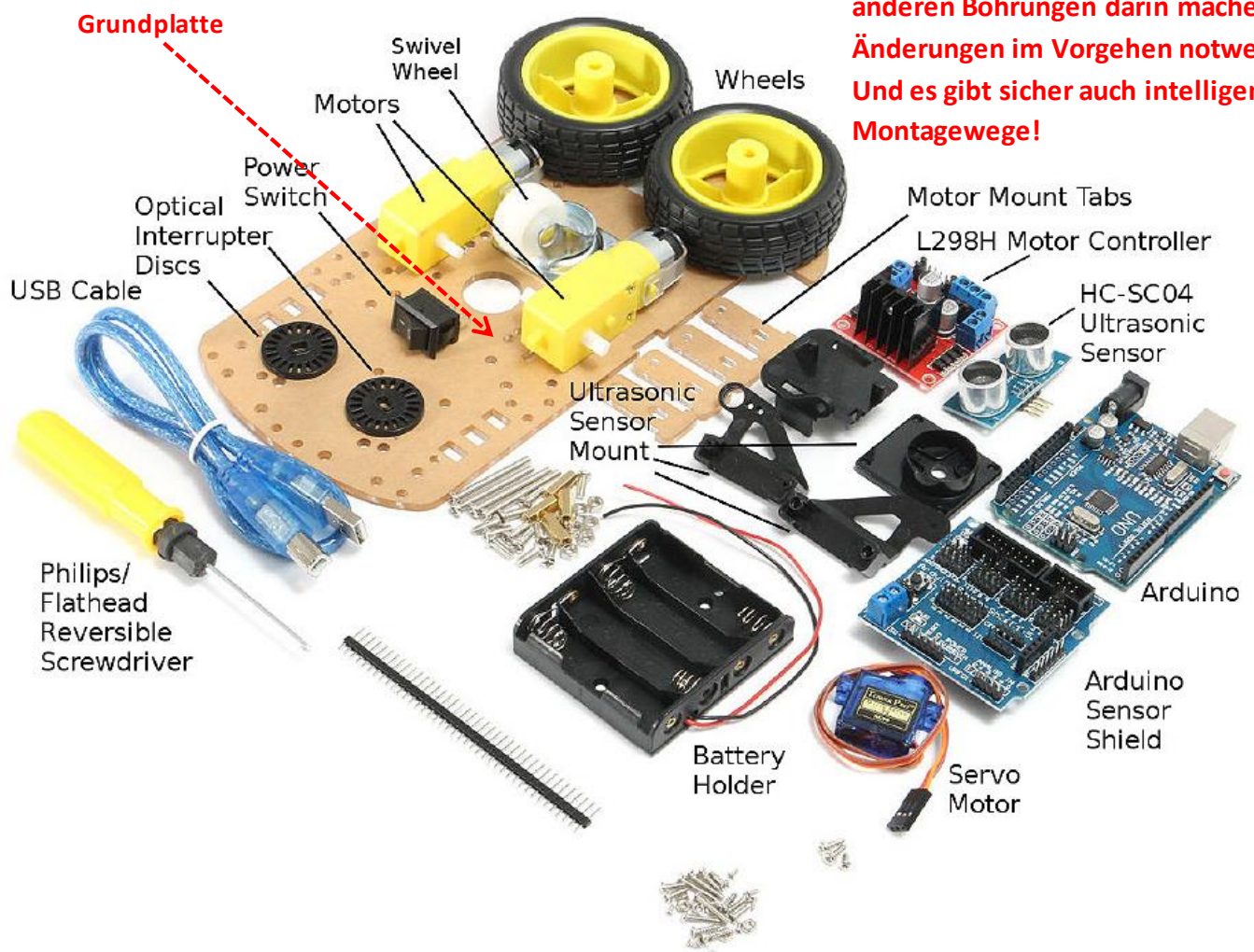
Roboter auf Arduino-Basis für weniger als 20€

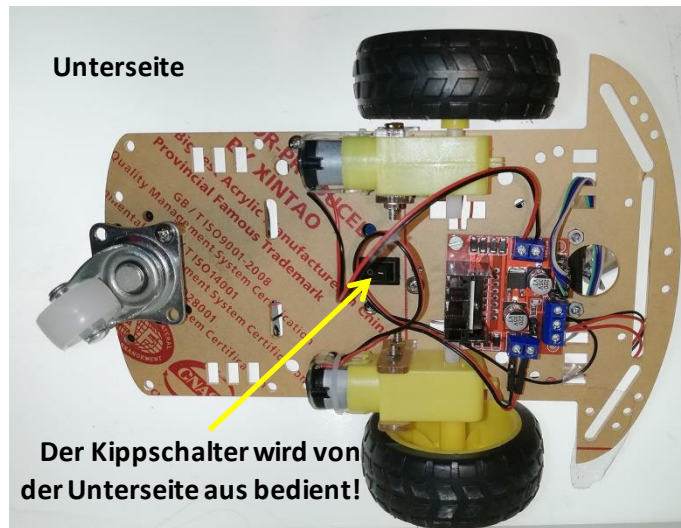
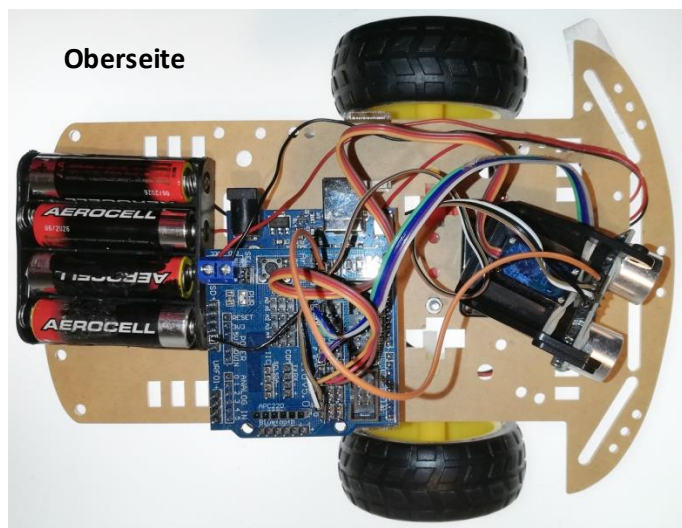
1. Einleitende Bemerkungen Aus China ließ sich zur Zeit der Vorbereitung dieser Veranstaltung ein Bausatz bestellen, der einen kompletten Roboter zusammenbaubar macht - bis auf wenige für etwa 15 €. (Die Einzelteile, die fehlen, sind 10-female-female-Pfostenstecker-Kabel und 4 Mignonzellen, schwarze und weiße Schaltlitze ca. 0,14 mm Querschnitt, 22 cm) „Herzstücke“ sind rArduino, Sensor-Shield, Motor-Controller, zwei Gleichstrom-Motoren mit Getriebe, Ultraschallsensor und Servomotor. Das unbedingt Positive an diesem Bausatz ist, dass es sich fast ausnahmslos um Standardbauteile handelt man die alle einzeln bei verschiedenen Anbietern nachgekauft werden können. Allerdings sind die Teile im Bausatz sicher keine Originalware, wie beispielweise der Arduino, was diskutiert werden sollte.



Leider sind die von mir gefundenen - natürlich englisch gehaltenen - Montageanleitungen unvollständig. Sie erklären nicht, wie alle Teile zum funktionsfähigen Roboter zusammengebaut werden können, obwohl fast alles vorhanden ist. Allerdings lassen sich die Informationen aus dem Internet zusammensuchen, sodass ein Zusammenbau gelingt. Für Unterrichtszwecke sollten Lerngruppenabhängig ggf. einige Vorbereitungen an den Bauteilen von der Lehrerin oder dem Lehrer vorgenommen werden und einige Baugruppen vormontiert werden. Die Vorschläge hier haben bei einem Exemplar funktioniert. Sie sollen als Anregung verstanden werden

Andere Grundplatten und mit anderen Bohrungen darin machen Änderungen im Vorgehen notwendig. Und es gibt sicher auch intelligentere Montagewege!





2. Vor dem eigentlichen Zusammenbau Unumgänglich ist: Die beiden Getriebe-Motoren müssen mit 18-cm-langen rotschwarzen Doppellitzen verlötet werden. Und an den Kippschalter muss ein rotes Stück Schalltitzte von 20 cm Länge angelötet werden und ein kurzes Stück von etwa 4 cm, das später mit einer Lüsterklemme mit dem gekürzten roten Batterieka- bel verbunden kann. Soll die Lüsterklemme vermieden werden, dürfen Schalter und roter Batterieanschluss erst dann verlötet werden, wenn der Schalter bereits richtig in die Grundplatte eingesetzt ist: Bedienseite nach unten und

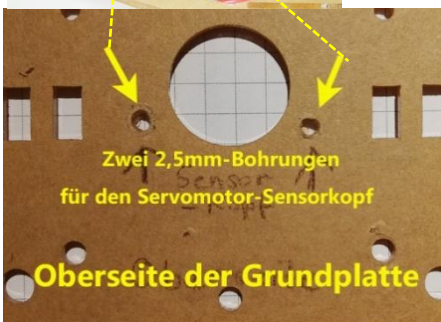
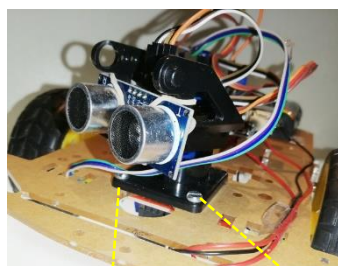


die Anschlüsse nach oben!

Umständlich ist auch, dass einige Bohrungen in der Grundplatte fehlen und dass nicht alle Schrauben in der richtigen Anzahl vorhanden sind. Unsön ist dabei besonders, dass von den Abstandshaltern aus Messing nur vier vorhanden sind. Acht wären ideal – vier für das

Hinterrad, vier für den Arduino. Aber es geht auch so ...

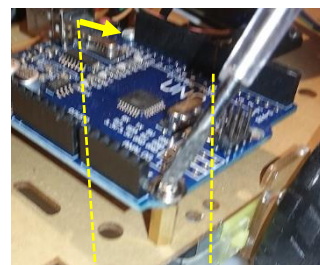
Besondere Aufmerksamkeit und Fingerspitzengefühl erfordert die Montage des Ultraschallsensors auf dem Servomotor – siehe Seite 3. Nach Fertigstellung dieser „Kopfeinheit“ muss ihre schwarze Halterung oben auf der Grundplatte verschraubt werden, wozu zwei Bohrungen von 2,5 mm links und das große runde Loch in der Grundplatte notwendig sind. Bevor mit dem Zusammenbau begonnen wird, sollten dann auch die beiden 4-mm-Bohrungen für die Abstandshalter unter dem Arduino gemacht werden.

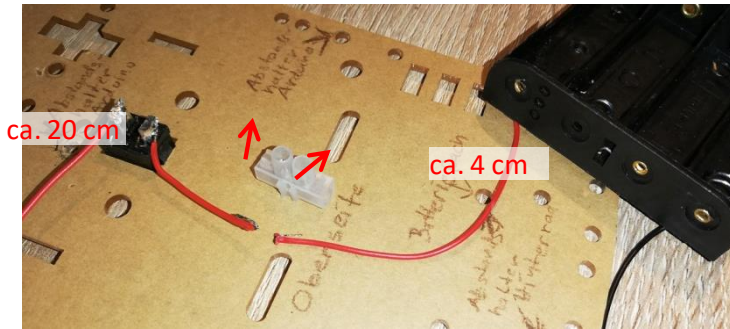


3. Zusammenbau Zunächst werden zwei Baugruppen separat fertiggestellt und geprüft, bevor sie zusammengefügt werden. Wir beginnen hier mit dem **Fahrwerk** aus Getriebemotoren, Hinterrad, Batteriehalterung, Schalter, Arduino, Sensor-Shield und Motor-Controller, im Folgenden kurz Fahrwerk genannt (3.1). Anschließend wird die „Kopfeinheit“ mit dem Ultraschallsensors auf dem Servomotor fertiggestellt. Schließlich erfolgt die Endmontage (3.3).



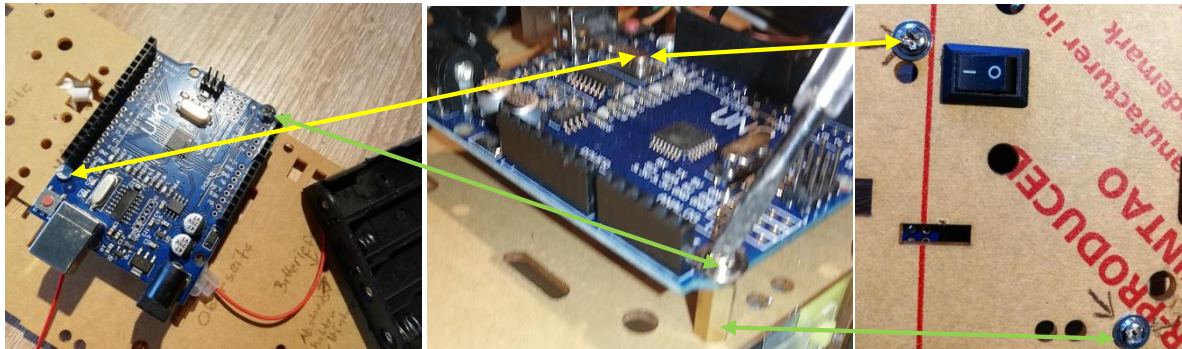
An den Getriebemotoren und dem Schalter muss gelötet werden!



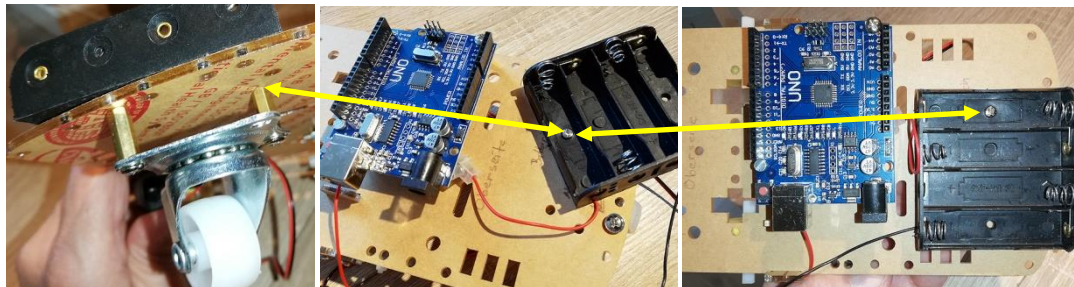


verbunden wird. (Soll die Lüsterklemme vermieden werden, dürfen Schalter und roter Batterieanschluss erst dann verlötet werden.) Die Lötflächen am Schalter müssen etwas rund gebogen werden, damit später der Arduino problemlos darüber montiert werden kann.

3.1.2 Nun wird der Arduino mit zwei Abstandshaltern und entsprechend vier Schrauben auf der Oberseite montiert.



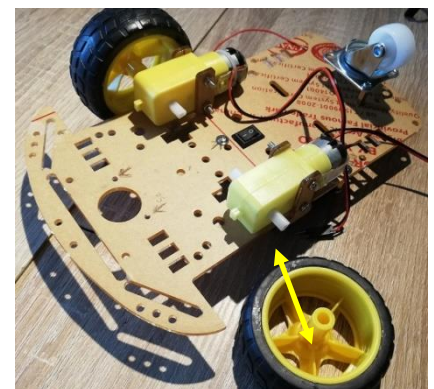
3.1.3 Die Batteriehalterung wird mit einer Schraube (gelb markiert) befestigt, wobei diese Schraube auch dazu dient,



das Hinterrad mittels der Messing-Abstandshalter zu befestigen. Der zweite Abstandhalter für die Hinterradbefestigung wird ebenfalls angeschraubt.

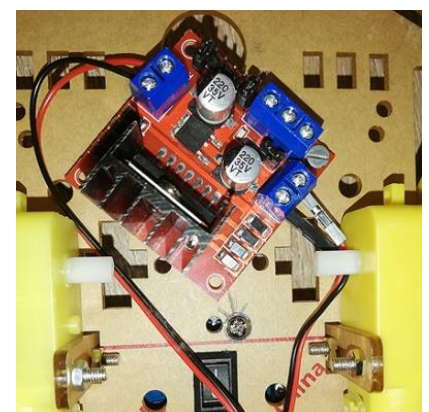
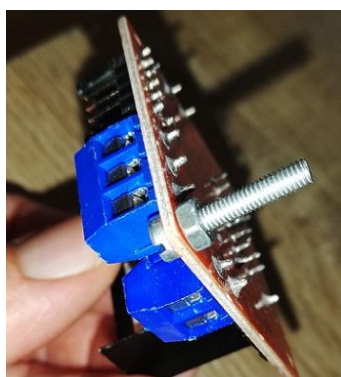


3.1.4. Beide Getriebemotoren, an welche die 20 cm langen Doppellitzen bereits angelötet sein müssen, werden so montiert, dass die Motoren in Richtung Hinterrad zeigen und die Schraubenköpfe nach außen. Denn die Gewindeüberstände würden später an den Rädern schleifen. Nun können Räder aufgesteckt werden.

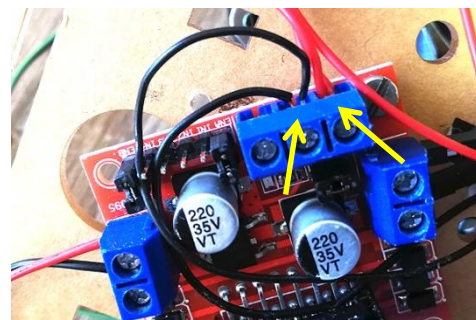
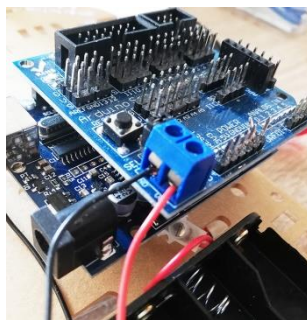


3.1.5 Der Motor-Controller wird auf der Unterseite verschraubt. Dabei muss die Schraube zunächst mit einer Extra-Mutter an der Platine abgebracht werden, die

später für einen ausreichenden Abstand Platine-Grundplatte sorgt. Nun können die Motoren angeschlossen werden, bevor die Platine mit einer zweiten Mutter an der Grundplatte festgeschraubt und fixiert wird.



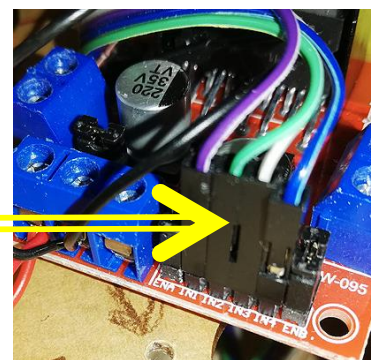
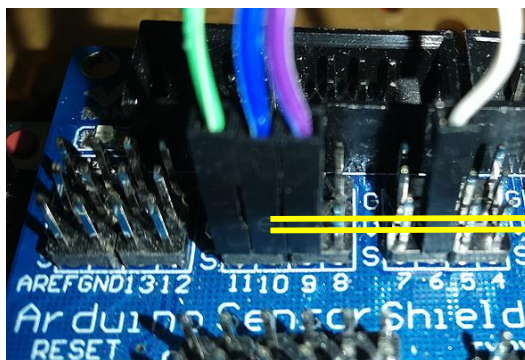
3.1.6 Der Sensor-Shield wird auf den Arduino gesteckt. Auch die beiden 20 cm langen Kabel für die Stromversorgung sollten jetzt an ihm angeschraubt werden. Diese werden dann zur Unterseite durch ein geeignetes Loch in der Grundplatte zum Motor-Controller geführt und dort verschraubt, und zwar so, dass das schwarze Kabel vom Arduino und das



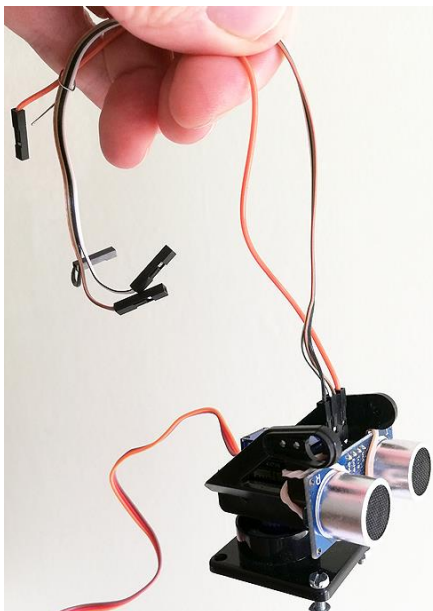
schwarze Kabel vom Arduino und das schwarze Kabel vom Batteriefach zusammen in der mittleren Buchse der Dreierleiste verschraubt werden, und das rote Kabel vom Arduino mit dem roten vom Kippschalter in der Buchse daneben – siehe Bild!

3.1.7 Die Pfostenstecker-Kabel vom Shield zum Controller werden folgendermaßen gesteckt:
Shield => Controller:

6=>IN2,
9=>IN1,
10=>IN4,
11=>IN3.



3.1.8 Erster Probelauf: Wenn die Batterien eingesetzt sind und das Programm „rob1.ino“ im Anhang auf den Arduino geladen wurde, müsste sich das Fargestell sich langsam ruckend vorwärts bewegen.



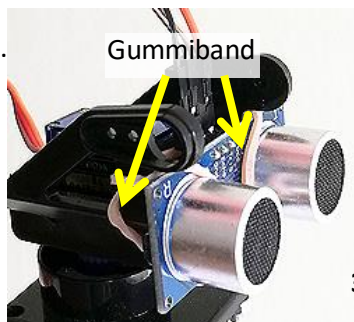
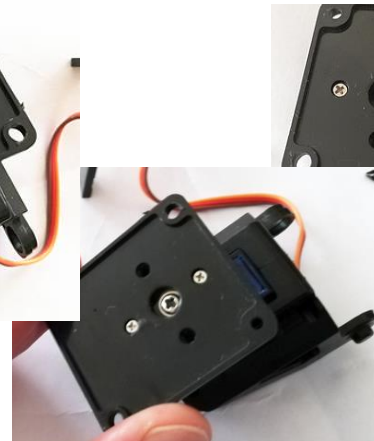
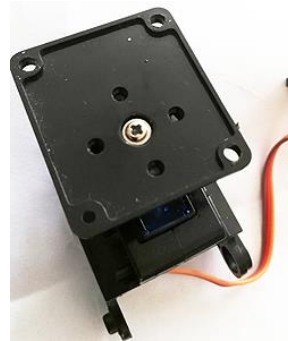
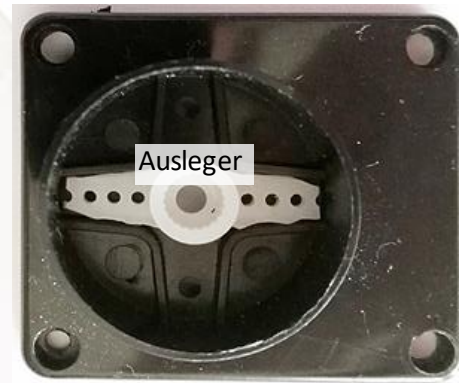
3.2 Sensorkopf

3.2.1 Der Sensorkopf besteht aus dem Ultraschallsensor, dem Servomotor und der Halterung. Halterung und Servomotor werden verschraubt, der Sensor wird mit einem Gummiband daran befestigt. Am Sensormotor sind drei Pfostenstecker-Kabel fest angebracht, der Sensor bekommt vier Leitungen.

Der Zusammenbau von Servomotor und Halterung stellt eine besondere Herausforderung dar. Hier muss mit einem Skalpell geschnitten werden, damit die Welle des Servomotors samt Ausleger fest an der schwarzen Halterungsfuß verankert werden kann. Um die feststehende Welle am Fuß dreht sich später dann die Einheit aus Servomotor, Motorfassung und Ultraschallsensor

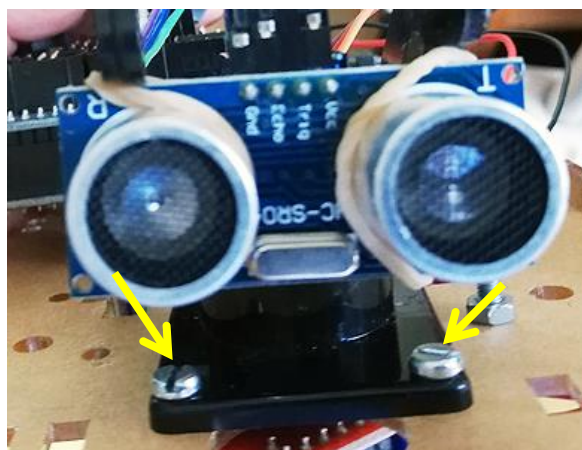


sozusagen als Kopf. Das Ganze ist sprachlich schlecht in den Griff zu bekommen. Hier sollen die Bilder weiterhelfen. Zum Drehen der winzigen Schrauben zur Befestigung des Auslegers am Halterungsfuß brauchst du einen entsprechenden Schraubenzieher.

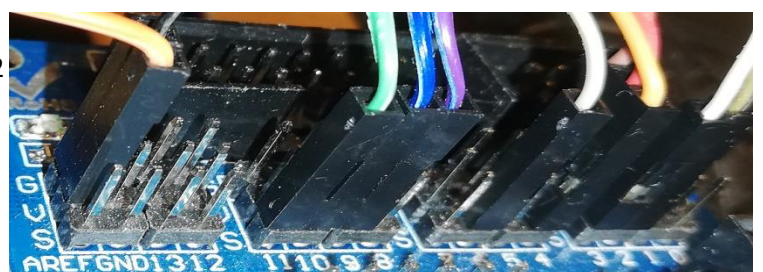


3.2.2 Zur Befestigung des Sensors an der Halterung eignet sich ein Gummiband!

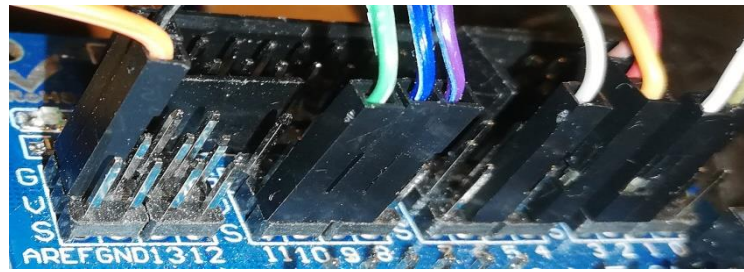
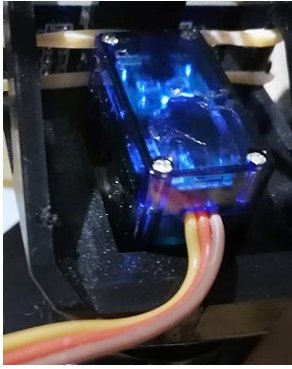
3.2.3 Der Sensorkopf wird mit zwei Schrauben mit Muttern auf der Grundplatte fixiert.



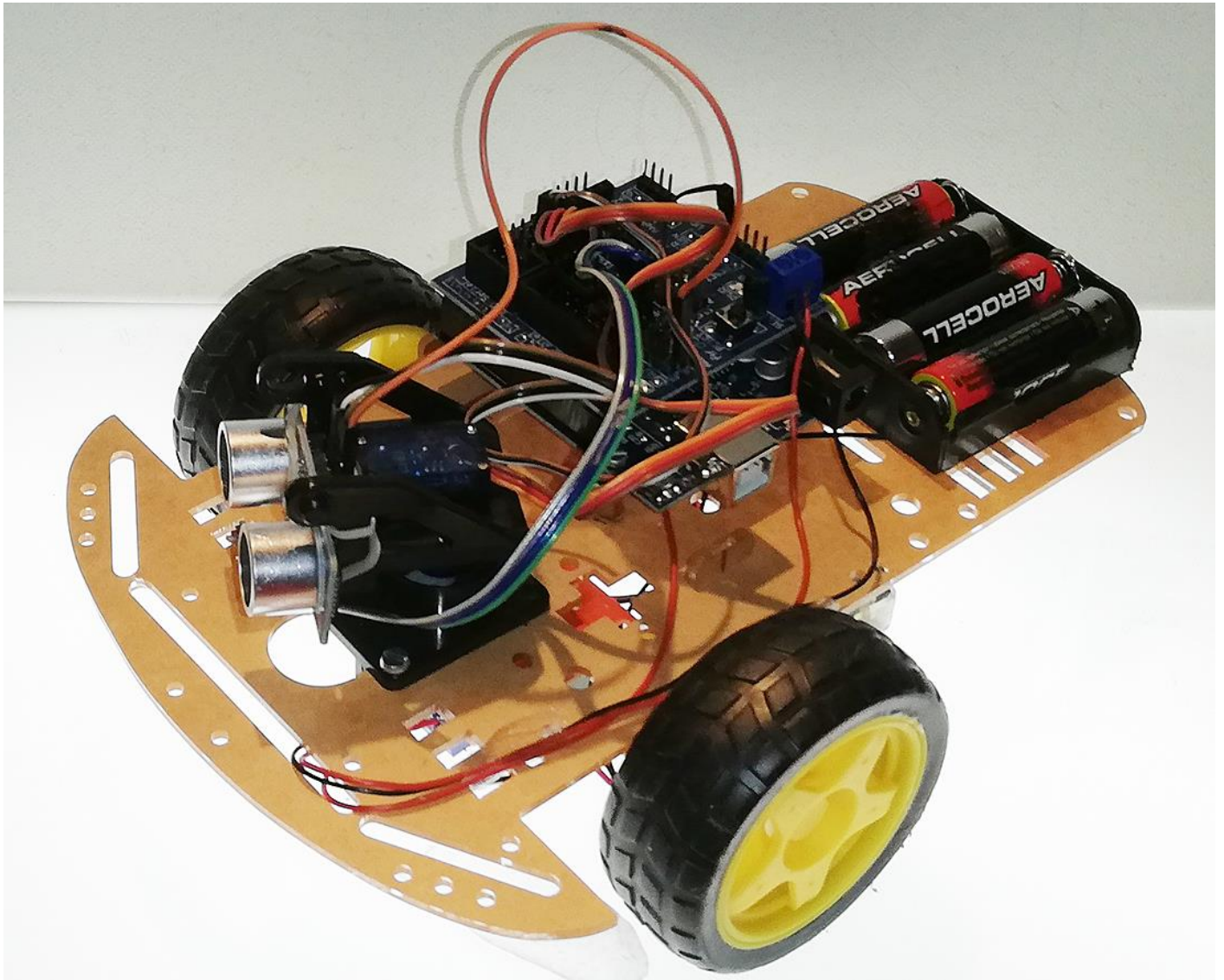
3.2.4 Verkabelung des Sensors: Sensor => Shield
GND => AREF-G, VCC => AREF-V, TRIG=>3, ECHO=>2



3.2.5 Verkabelung der Servomotors mit angebauten Kabeln: Kabelfarbe => Shield: Orange =>4-S, Rot 4-V, Brau => 4-G



3.3 Fertig – zum Testen und Fehlersuchen!



Screenshot „rob.1.ino“ in der Arduino-Entwicklungsumgebung

rob1 | Arduino 1.8.12

Datei Bearbeiten Sketch Werkzeuge Hilfe



rob1 \$

```
#include <Servo.h>
#define pinServo 4
#define trig 3
#define echo 2
Servo WinkelServo;
int untereEntfernungsGrenze=30; long echoZeit; float geradeausEntfernung;
float entfernungsListeWert[6]; int winkel;
int ScanWinkel[8] = {20, 40, 55, 70, 95, 110, 130, 70};
int DelayGeradeVorZurueckWert = 500; int DelayWertDrehung = 120;
int motor_links_a=9; int motor_links_b=6; int motor_rechts_a=11; int motor_rechts_b=10;
void setup() {
  Serial.begin(9600); WinkelServo.attach(pinServo);
  pinMode(motor_links_a, OUTPUT); pinMode(motor_links_b, OUTPUT);
  pinMode(motor_rechts_a, OUTPUT); pinMode(motor_rechts_b, OUTPUT);
  pinMode(trig, OUTPUT); pinMode(echo, INPUT);
  void stopp() {digitalWrite(motor_links_a, LOW); digitalWrite(motor_links_b, LOW);
  digitalWrite(motor_rechts_a, LOW); digitalWrite(motor_rechts_b, LOW);
  delay(DelayGeradeVorZurueckWert);}
  void vorwaerts() {digitalWrite(motor_links_a, 0); digitalWrite(motor_rechts_a, 0);
  digitalWrite(motor_links_b, 254); digitalWrite(motor_rechts_b, 254);
  delay(DelayGeradeVorZurueckWert);}
  void ruckswaerts() {digitalWrite(motor_links_b, LOW); digitalWrite(motor_links_a, 100);
  digitalWrite(motor_rechts_b, LOW); digitalWrite(motor_rechts_a, 100);
  delay(DelayGeradeVorZurueckWert);}
  void links() {digitalWrite(motor_links_a, LOW); digitalWrite(motor_links_b, 100);
  digitalWrite(motor_rechts_a, 100); digitalWrite(motor_rechts_b, LOW);
  delay(DelayWertDrehung);}
  float entfernungsMessung(int winkel) {WinkelServo.write(winkel); delay(10);
  digitalWrite(trig, LOW); delayMicroseconds(20); digitalWrite(trig, 100); delayMicroseconds(20);
  digitalWrite(trig, LOW); echoZeit = pulseIn(echo, 100); geradeausEntfernung = echoZeit/58;
  return geradeausEntfernung;}
  void scan180Grad() {for (int i = 0; i < 8; i++) {WinkelServo.write(ScanWinkel[i]); delay(1000);}}
  void loop() {scan180Grad();
  if(entfernungsMessung(70) > untereEntfernungsGrenze) {vorwaerts();}
  else {stopp(); ruckswaerts(); links();}; stopp();}
```

„Rob.ino“ zum Kopieren

```
#include <Servo.h>

#define pinServo 4

#define trig 3

#define echo 2

Servo WinkelServo;

int untereEntfernungsGrenze=30;long echoZeit;float geradeausEntfernung;

float entfernungsListeWert[6];int winkel;

int ScanWinkel[8] = {20, 40, 55, 70, 95, 110, 130,70};

int DelayGeradeVorZurueckWert = 500;int DelayWertDrehung = 120;

int motor_links_a=9; int motor_links_b=6;int motor_rechts_a=11; int motor_rechts_b=10;

void setup() {

  Serial.begin(9600);WinkelServo.attach(pinServo);

  pinMode(motor_links_a,OUTPUT);pinMode(motor_links_b,OUTPUT);

  pinMode(motor_rechts_a,OUTPUT);pinMode(motor_rechts_b,OUTPUT);

  pinMode(trig,OUTPUT);pinMode(echo,INPUT);}

void stopp(){digitalWrite(motor_links_a, LOW);digitalWrite(motor_links_b, LOW);

digitalWrite(motor_rechts_a, LOW);digitalWrite(motor_rechts_b, LOW);

delay(DelayGeradeVorZurueckWert);}

void vorwaerts(){digitalWrite(motor_links_a, 0); digitalWrite(motor_rechts_a, 0);

digitalWrite(motor_links_b, 254); digitalWrite(motor_rechts_b, 254);

delay(DelayGeradeVorZurueckWert);}

void rueckswaerts(){digitalWrite(motor_links_b, LOW);digitalWrite(motor_links_a, 100);

digitalWrite(motor_rechts_b, LOW);digitalWrite(motor_rechts_a, 100);

delay(DelayGeradeVorZurueckWert);}

void links(){digitalWrite(motor_links_a, LOW);digitalWrite(motor_links_b, 100);

digitalWrite(motor_rechts_a, 100);digitalWrite(motor_rechts_b, LOW);

delay(DelayWertDrehung);}

float entfernungsMessung(int winkel){WinkelServo.write(winkel);delay(10);

digitalWrite(trig, LOW);delayMicroseconds(20);digitalWrite(trig, 100);delayMicroseconds(20);

digitalWrite(trig, LOW);echoZeit = pulseIn(echo,100);geradeausEntfernung = echoZeit/58;

return geradeausEntfernung;}

void scan180Grad(){for (int i = 0; i < 8; i++){WinkelServo.write(ScanWinkel[i]);delay(1000);}}

void loop(){scan180Grad();

if(entfernungsMessung(70)> untereEntfernungsGrenze){vorwaerts();}

else{stopp();rueckswaerts();links();};stopp();}
```