

Arduino-Roboter

Für den Bau eines Arduino-Roboters gehst du auf die Vivibot.ch Seite:

<http://vivibot.ch/>

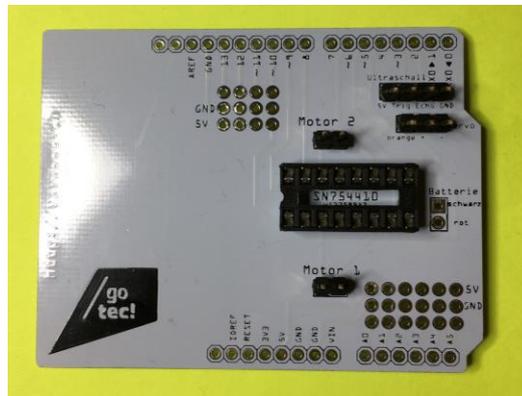
Das Material kannst du dir mit der Bestellliste (siehe unsere *Downloads*) auf AliExpress bestellen. Beachte aber, dass das chinesische Arduino UNO Board einen eigenen Treiber braucht. Das Vivi-Shield kannst du auf SchulArena (ViviShield Zusatz-Platine) herunterladen und die Platine bestellen bei pcbway.com (Gerber-Files als zip hochladen):

<https://www.pcbway.com/>

Arduino Uno:

Vivi-Shield*:

Chin. Modell



Treiber: <http://vivibot.ch/china-arduino/>

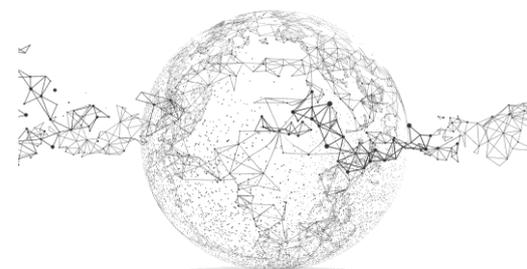
Es folgen nun einige Bilder zum Modell R2-D2 Büchsenroboter:



*Das ViviShield (Platine) kannst du auch bei vivibot.ch als Fritzing-Datei holen (.fzz) und bei Fritzing hochladen. Fritzing Programm: <http://fritzing.org/home/>

Fritzing ist ein Platinen-Programm, das dir von deinen Bread-Board Konstruktionen Platinenpläne erstellt.

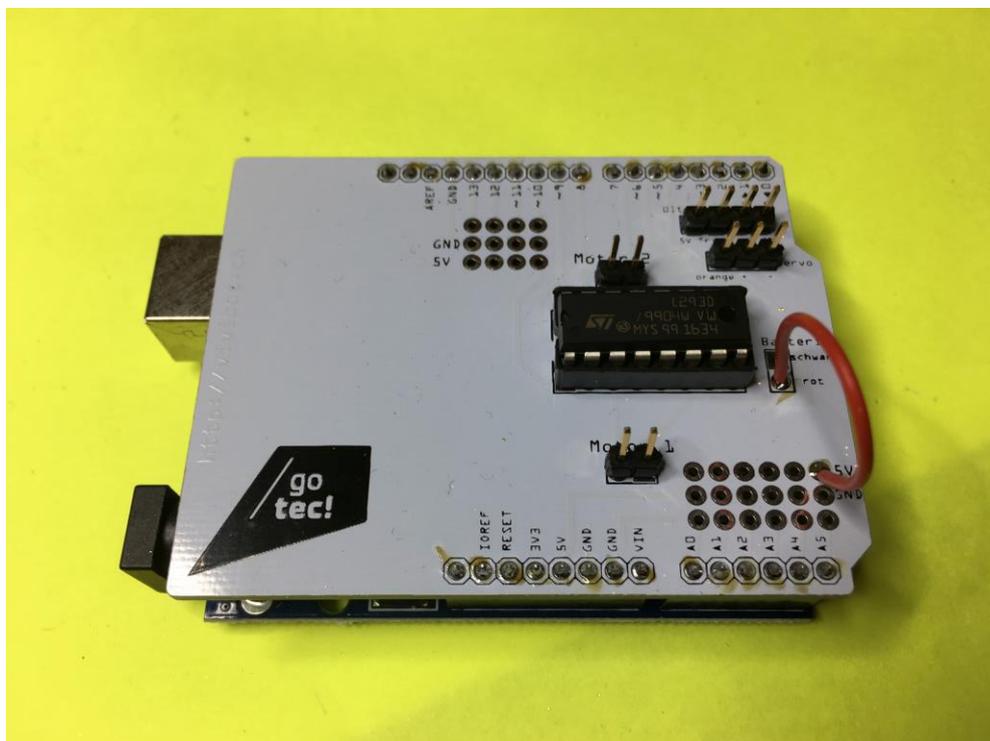
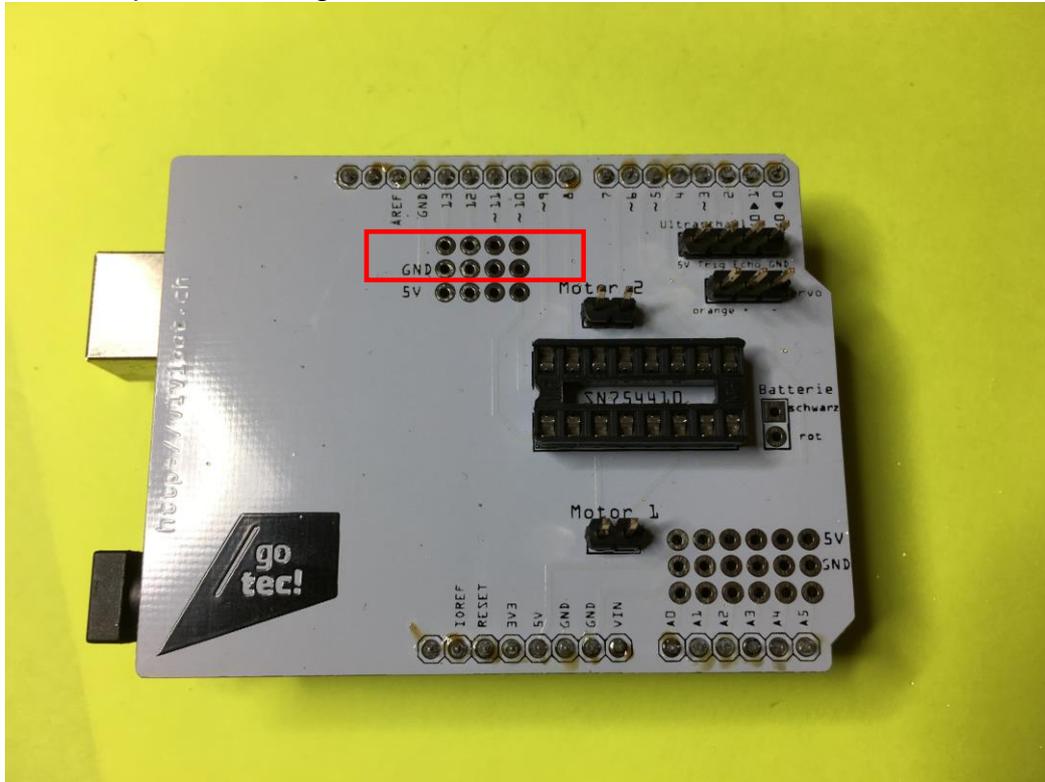
In Auftrag geben solltest du aber bei pcbway.com, da es viel billiger ist. Dort werden aber keine .fzz Dateien unterstützt. Du musst sie zuerst als Gerber-Files speichern.



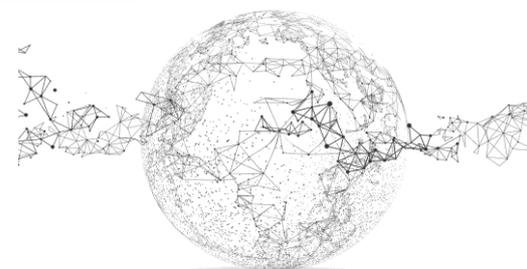
Tipp: Löte überall Stecker: Motor 1 und 2, Ultraschall (4er), Servo (3er)

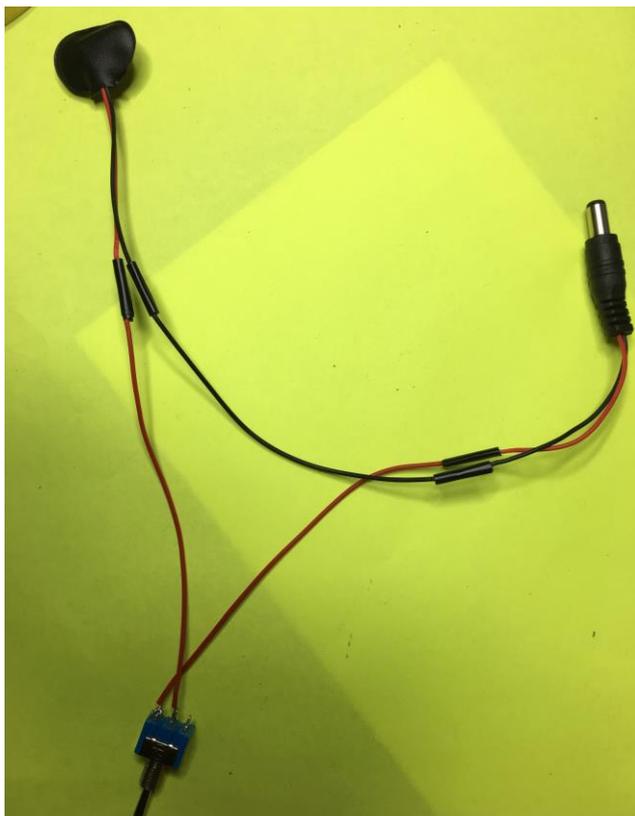
Ausserdem auch an den PINS 13, 12, 11, 10 (siehe roter Rahmen) / Dort kannst du LEDs oder eine Hupe anlöten und im Code programmieren.

Ein Beispiel-Code folgt am Ende des Dossiers.



Batterie-Brücke löten!



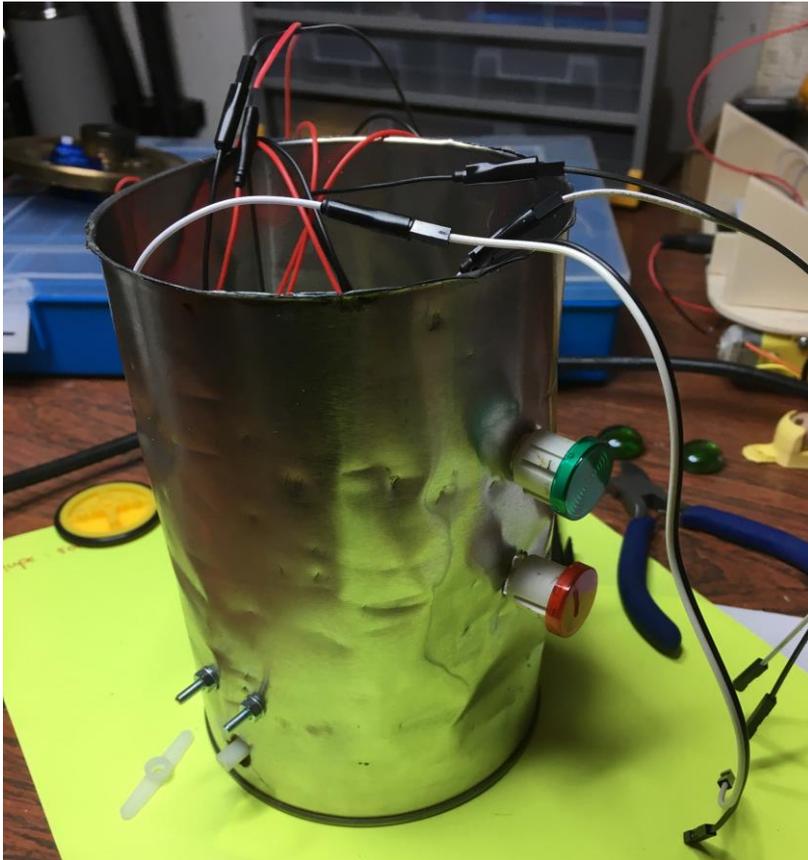


Schalter für 9V-Batterie-Unterbruch



Deckel mit Hupe und Servo

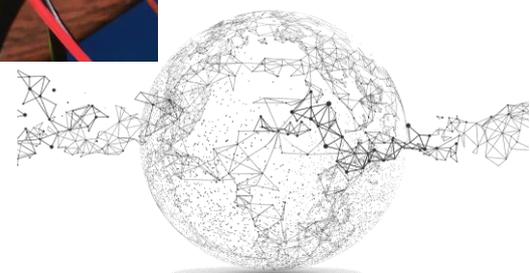
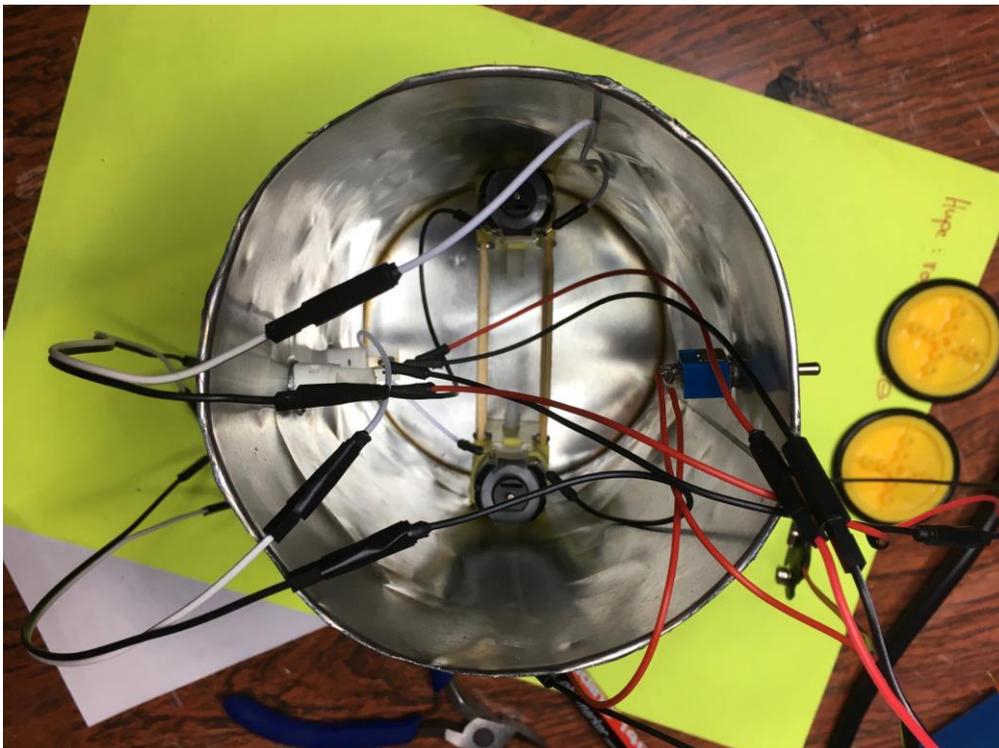


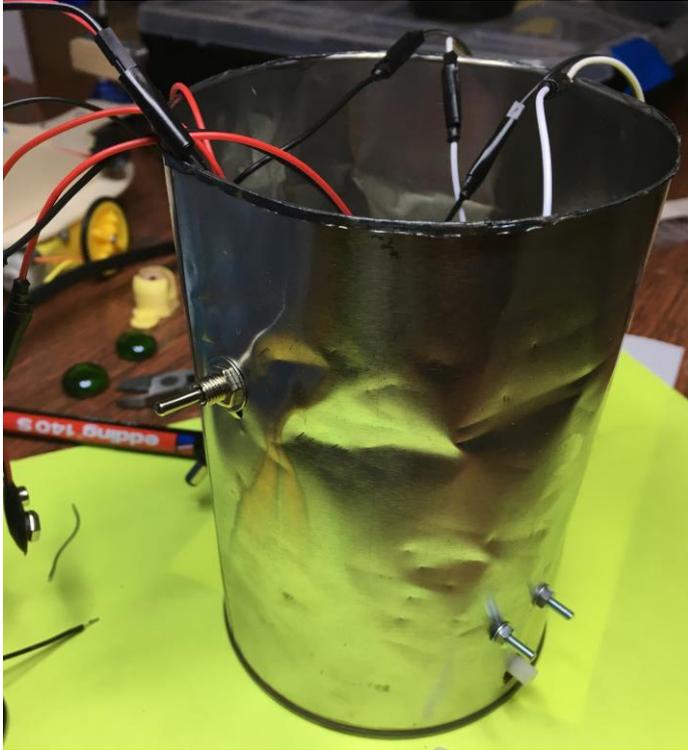


Buchse mit Vergabelungen

Bohrungen für Räder, Lampen und Schalter

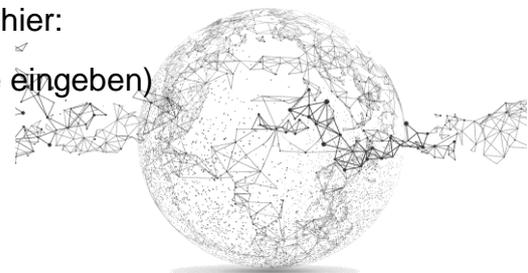
Tipp: mit Steckern und Jumpers ist alles einfacher zum Ein- und Ausstecken





Die Maske für den Ultraschallsensor wurde mit einem 3D-Drucker (Ultimaker) gedruckt. Vorlagen zu solchen und ähnlichen Teilen gibt es hier:

<https://www.thingiverse.com/> (einfach *Arduino* in der Suche eingeben)



Damit du programmieren kannst, brauchst du die **Arduino-Software**.

Download:

<https://www.arduino.cc/en/Main/Software>

Falls, du einen billigeren China *Arduino Uno* gekauft hast, musst du bei der Installation das Häkchen für den USB-Treiber rausnehmen und danach den Treiber von Hand installieren. Achte auch, dass der COM-Anschluss stimmt!

Wenn du einen originalen *Arduino Uno* hast (sieht man am *Made in Italy* oder an *arduino.cc*), kannst du normal installieren und brauchst dich nicht um Treiber zu kümmern.

Das Original (unten abgebildet) kostet ca. 20 Euro, der falsche Chinese bekommst für etwa 3.30 Dollar.

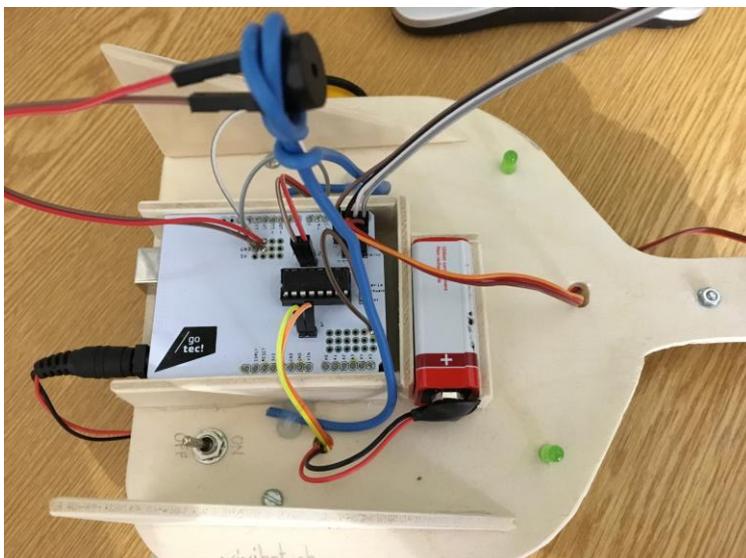


Die Programmiersprache ist CC / C+

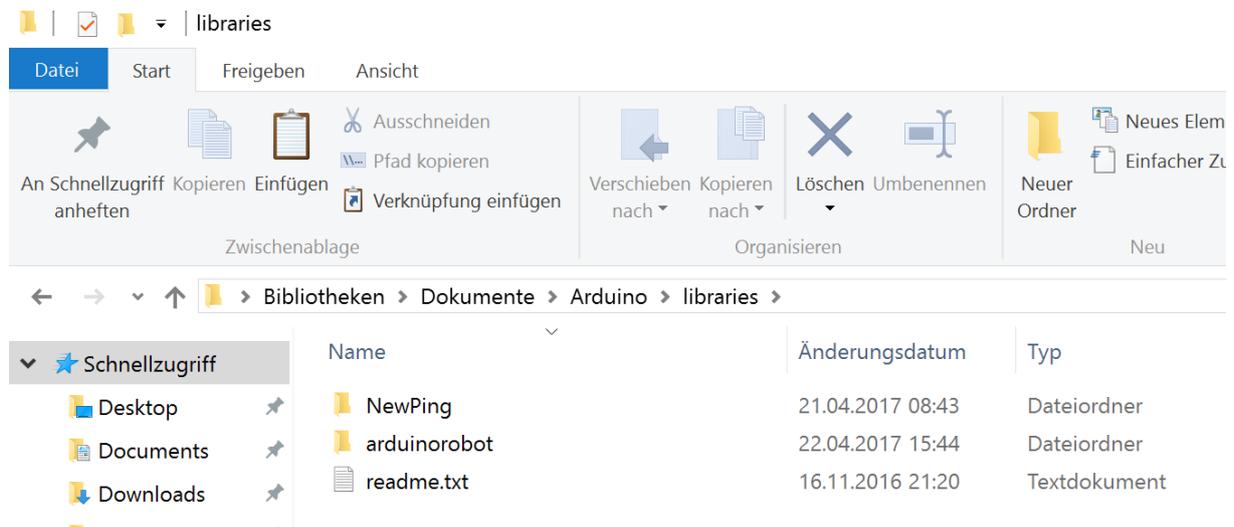
Du kannst mit einem sogenannten *Breadboard* auch LED-Vorübungen am Arduino Board machen. Anleitungen dazu gibt es hier:

<http://vivibot.ch/arduino-programmieren/>

Hier ein Bild eines anderen Modells (nicht Büchsenroboter):



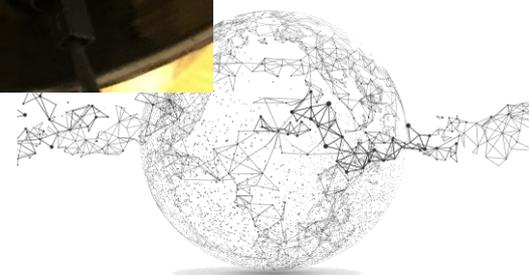
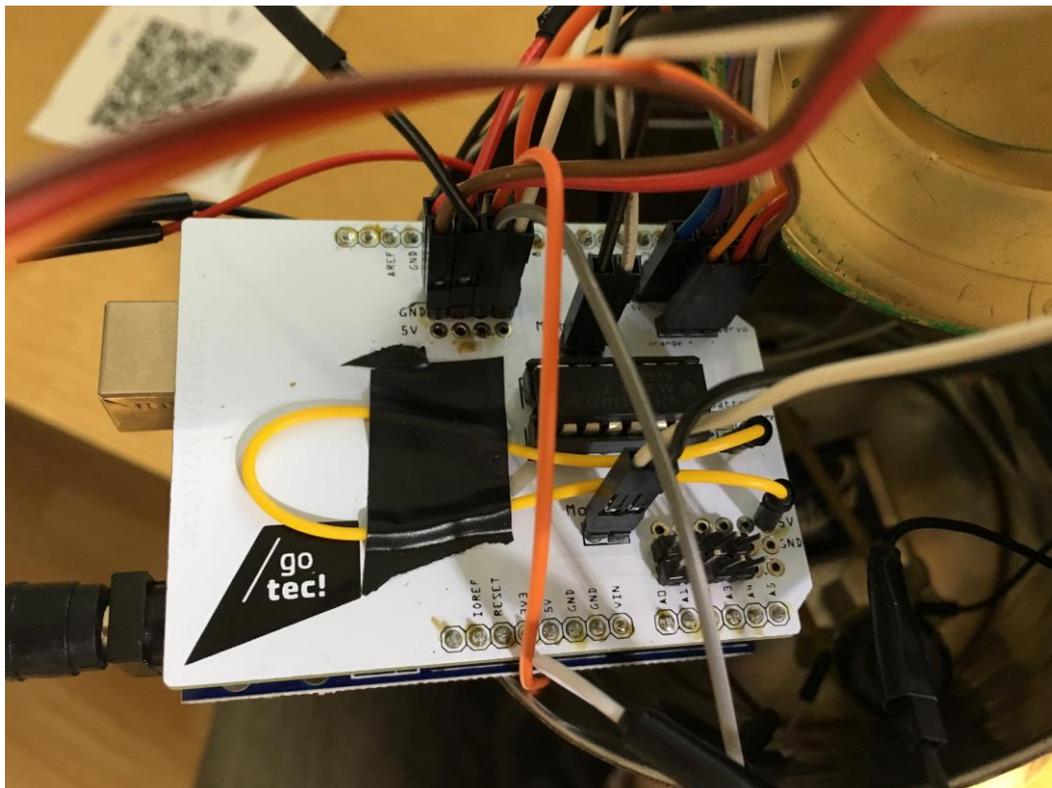
Willst du den Roboter programmieren, musst du in den Ordner (siehe eigene Dateien/Dokumente) „Arduino/libraries“ folgende Library (NewPing) kopieren: <http://playground.arduino.cc/Code/NewPing#Download>



Im folgenden Code kannst du dann Veränderungen anbringen und ihn auf das Board hochladen. Dieses Beispiel hat zusätzlich an den Pins 13, 10, 11 LEDs und am Pin 12 eine Hupe. Du kannst den Code auch bei unseren *Downloads* herunterladen (arduinorobot.zip) und gleich in den Arduino Editor einlesen.

Code: auf Seiten 8 und 9

Bild R2-D2 Büchsenroboter mit allen Anschlüssen:



```

arduinorobot
//Weil wir einen Servo und einen Ultraschall-Sensor benutzen, fügen wir zwei Bibliotheken ein, die deren Benutzung vereinfachen.
#include <Servo.h>
#include <NewPing.h>

//damit wir uns nicht die Nummern der Pin's merken müssen, geben wir ihnen Namen, an denen wir die Pinfunktion erkennen.
#define LeftMotorForward 6
#define LeftMotorBackward 5
#define RightMotorForward 8
#define RightMotorBackward 9
#define USTrigger 3
#define USEcho 2
#define MaxDistance 100
#define LED 13
#define LED 10
#define LED 11
#define Horn 12

//Hier erstellen wir zwei 'objekte', eine für den Servo und eine für den Ultraschall-Sensor
Servo servo;
NewPing sonar(USTrigger, USEcho, MaxDistance);

//Hier erstellen wir Variablen für Zahlen ohne Vorzeichen, welche wir weiter unten im Code brauchen für Messwerte, Geschwindigkeit etc.
unsigned int duration;
unsigned int distance;
unsigned int FrontDistance;
unsigned int LeftDistance;
unsigned int RightDistance;
unsigned int Time;

void setup() //Dieser Abschnitt läuft nur einmahl beim Programmstart
{
  //Serial.begin(115200); //damit wir zum Fehler suchen eine serielle Verbindung haben

  //Hier definieren wir die Pin Modus. Weil wir Signale ausgeben werden, stellen wir sie als Ausgänge
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorForward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);
  pinMode(LED, OUTPUT);
  pinMode(Horn, OUTPUT);
  servo.attach(4); //Der Servo hängt an pin 4
}

void loop() //Dieser Abschnitt wiederholt sich selber während der Arduino eingeschaltet ist.
{
  servo.write(90); //Der Servo schaut vorwärts
  delay(500);
  scan(); //ruft die Scan Funktion auf
  FrontDistance = distance; //Setzt die Variable FrontDistance auf den Wert, den wir von der Scan Funktion bekommen haben
  if(FrontDistance > 40 || FrontDistance == 0) //falls 40cm vor dem Roboter nichts ist, dann..
  {
    moveForward();
    digitalWrite(13, HIGH);
    digitalWrite(10, HIGH);
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    //rufe die moveForward Funktion auf
  }
  else //andernfalls (also falls etwas kommt innert 40cm ) dann...
  {
    moveStop();
    digitalWrite(13, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(12, HIGH);
    //rufe die moveStop Funktion
    servo.write(167); //Drehe den Servo nach links (möglicherweise kann der kleine Servo nicht die ganzen 180Grad)
    delay(500); //Warte eine halbe Sekunde, bis der Servo links ankommt
    scan(); //Rufe die scan Funktion
    LeftDistance = distance; //Setze die Variable LeftDistance auf die Distanz von links
    servo.write(0); //Drehe den Servo nach rechts
    delay(500); //Warte eine halbe Sekunde, bis der Servo rechts ankommt
    scan(); //Rufe die scan Funktion auf
    RightDistance = distance; //Setze die Variable RightDistance auf die Distanz von rechts
    if(RightDistance < LeftDistance) //Ist die Distanz rechts kleiner als der linke Abstand..
    {
      moveLeft(); //Rufe die moveLeft Funktion auf
      delay(500); //Pausiere das Programm eine halbe Sekunde, bis der Roboter gedreht hat
    }
    else if(LeftDistance < RightDistance) //Andernfalls, wenn die Distanz links kleiner ist als rechts dann..
    {

```



```
    moveRight(); //Rufe die moveRight Funktion auf
    delay(500); //Pausiere das Programm eine halbe Sekunde, bis der Roboter gedreht hat
  }
  else //falls die Distanz links und rechts gleich ist (das passiert sehr unwarscheinlich) dann..
  {
    moveBackward(); //Rufe die moveBackward Funktion auf
    delay(200); //Pausiere das Programm für 200 Millisekunden um den Roboter rückwärts fahren zu lassen
    moveRight(); //rufe die moveRight Funktion
    delay(200); //Pausiere das Programm für 200 Millisekunden um den Roboter nach rechts zu drehen
  }
  moveStop();
}

void moveForward() //Diese Funktion macht das der Roboter vorwärts fährt
{
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(RightMotorForward, HIGH);
}

void moveBackward() //Diese Funktion macht das der Roboter rückwärts fährt
{
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorForward, LOW);
  digitalWrite(RightMotorBackward, HIGH);
}

void moveLeft() //Diese Funktion macht das der Roboter nach links dreht
{
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(RightMotorForward, HIGH);
}

}

void moveRight() //Diese Funktion macht das der Roboter nach rechts dreht
{
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, LOW);
  digitalWrite(RightMotorBackward, HIGH);
}

}

void moveStop() //Diese Funktion macht das der Roboter still steht
{
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);
}

}

void scan() //Diese Funktion misst die Distanz vor dem Ultraschallsensor
{
  delay(50);
  Time = sonar.ping();
  distance = Time / US_ROUNDTRIP_CM;
}
}
```

Viel Spass ☺



