

# Roboter-Fahrzeug für Arduino

Ellmitron-Best.Nr.: 10-654-1, 10-654-2, 10-654-G1, 10-654-G2 und 10-654-SE

10-654-PL, 10-654-PLE



## Thema

Es wird ein Roboterfahrzeug hergestellt, das je nach Programmierung verschiedenste Aufgaben erfüllen kann. Der Arduino-UNO wird dabei auf eine Trägerplatine gesteckt, die alle nötigen Bauteile und das Treiber-IC L-293 enthält. Eine lose Verdrahtung entfällt. In dieser Anleitung werden der Aufbau, die Möglichkeiten des Fahrzeugs und grundlegende Programmierschritte besprochen.

## Möglichkeiten

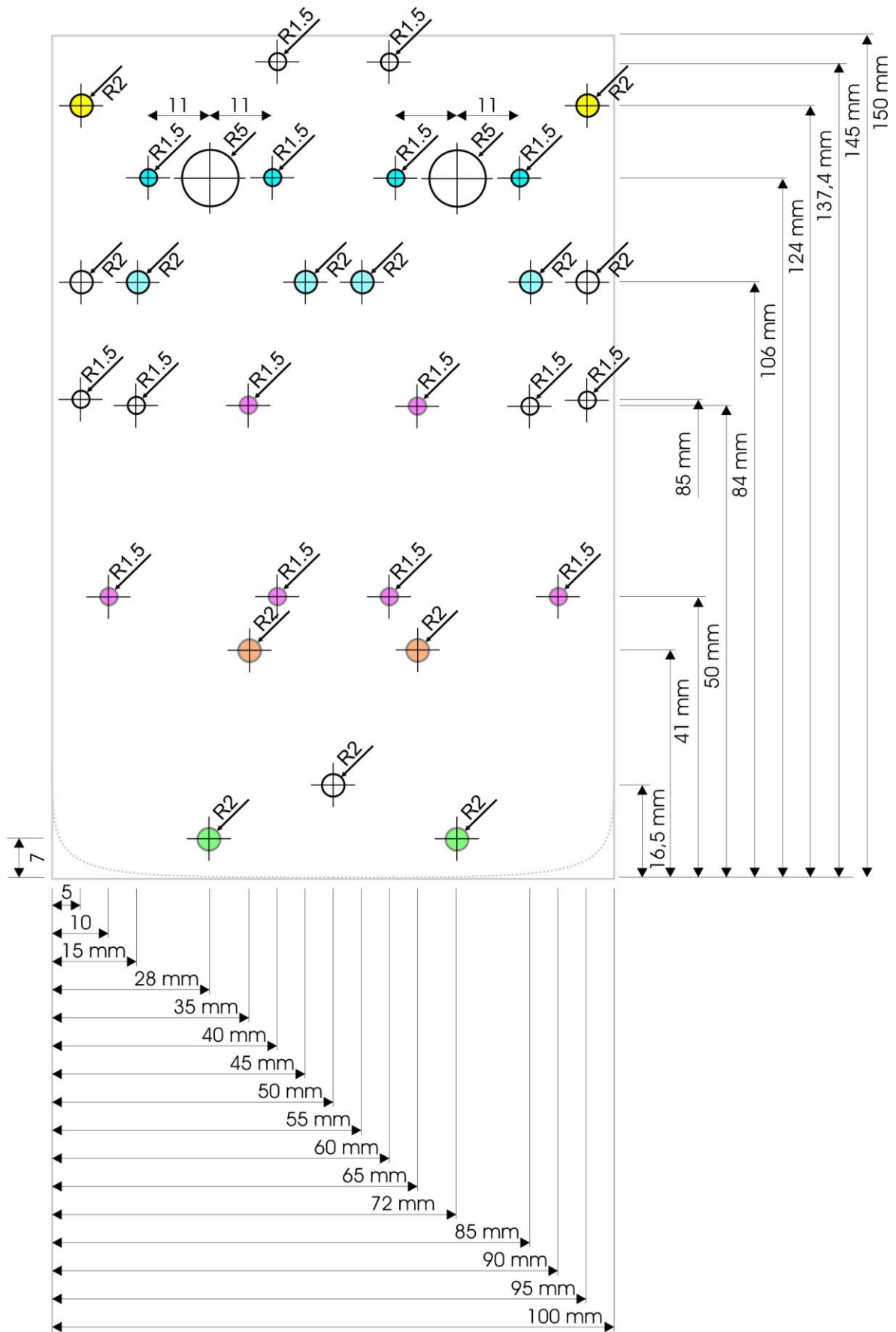
Ansteuerung von neun fahrzeugtypischen LEDs (Bremslicht, Blinklicht etc.)  
 Zwei unabhängige Motoren für viele verschiedene Fahrmanöver (vorwärts, rückwärts, auf der Stelle drehen, Kurven fahren, beschleunigen, bremsen)  
 Hindernis-/Abstandserkennung (mit Ultraschall-Entfernungssensor)  
 Fernbedienung aller Funktionen (mit Infrarot Fernbedienung)

## Arbeitsschritte

Folgende Arbeiten sind zur Herstellung und zum Betrieb des kompletten Fahrzeugs nötig und werden in dieser Anleitung beschrieben (alternative Hinweise sind *kursiv* gedruckt):

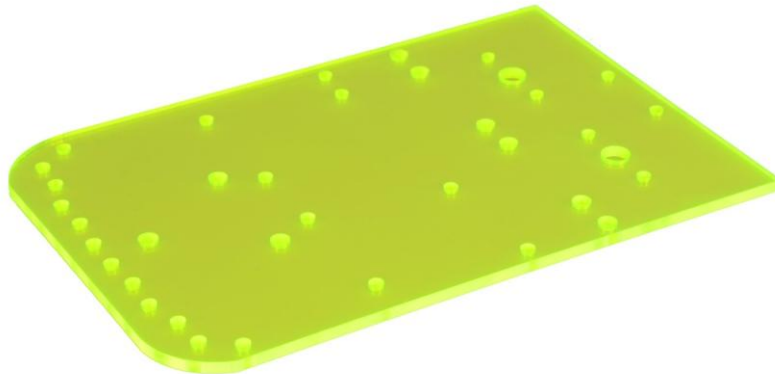
- Fräsen der Grundplatte mit dem KOSY - alternativ: Bohren, Sägen, Feilen
- Montage der mechanischen Komponenten
- Bestückung der Platine und Verlöten der Bauteile
- Funktionsprüfung der Platine mit aufgestecktem Arduino
- Montage der Platine auf dem Fahrzeug
- Programmierung des Arduino

Herstellen der Grundplatte



Alle mechanischen Komponenten sowie die Stromversorgung und die Trägerplatine werden an der Grundplatte 150mm X 100mm befestigt.

Ein Fräsprogramm in NCCAD für die Grundplatte können sie auf unserer Homepage in der Detailansicht des Artikels 10-654 downloaden.



Wird die Grundplatte manuell hergestellt, ist es besonders wichtig, die Bohrungen zur Befestigung der Achse und der beiden Motoren genau zu positionieren. Die Verwendung eines Anreiß- und Höhenmessgerätes ist ideal. Alternativ können die Bohrungen für die Achslager etwas größer ausgeführt werden, damit man später den Abstand zwischen Motorschnecke und Achszahnrad optimieren kann.

*Werden fertige Getriebemotoren verwendet (Zusammenstellung GT-6 oder GT-180), befestigt man sie wie auf Seite 4 unten mit je 2 Winkeln und Schrauben M4 X 6.*

*Das Roboterfahrzeug kann sowohl mit einer 9V Batterie/Akku als auch mit 6 X 1,5V Batterie/Akku AA betrieben werden. Beide Stromquellen können oben oder unten angebracht werden sollen. Bei Verwendung der Solarergänzung ist die Anbringung unten sinnvoll.*

*Alle schwarz gezeichneten Bohrungen in der Zeichnung auf Seite 2 sind unbedingt nötig. Alle anderen Bohrungen sind farblich gekennzeichnet (abhängig von Antrieb und Stromversorgung). Werden alle Bohrungen hergestellt, stehen Ihnen auch alle Möglichkeiten offen.*

- Nur für Roboterfahrzeug mit Solar-Getriebemotor
- Nur für Roboterfahrzeug mit Schneckenantrieb
- Nur für Roboterfahrzeug mit 9V-Block
- Nur für Roboterfahrzeug mit 2 X Batteriehalter für je 3 X Batterien oder Akkus AA
- Nur für Solar-Ergänzung

## Trägerplatine

Die Trägerplatine übernimmt eine entscheidende Rolle. Sie kann sehr gut für eigene Projekte (aufrechter Roboter, Propellerboot, Spezialmaschine etc.) verwendet werden.

Durch die Platine sind alle Sensoren, 9 LEDs sowie die Stromversorgung und die beiden Motoren nach dem Aufstecken des Arduino fest verbunden und mit den passenden Vorwiderständen bzw. Treiber-IC versehen.

"Wackelkontakte" sind dadurch ausgeschlossen und ein Steckboard ist nicht nötig.

Die beiden Motoren werden über den IC L-293-D angesteuert, der fest auf der Platine verbaut und angeschlossen ist. Gleichzeitig können alle digitalen und 5 analoge Ausgänge über Stiftleisten bereitgestellt werden, sodass anstatt der eingebauten Verbraucher auch andere Verbraucher angeschlossen werden können.

*Die LEDs können direkt in die Platine eingelötet werden oder über Stiftleisten mit Verbindungsleitungen verbunden werden, sodass die LEDs in eigen konstruierten Robotern frei platziert werden können.*

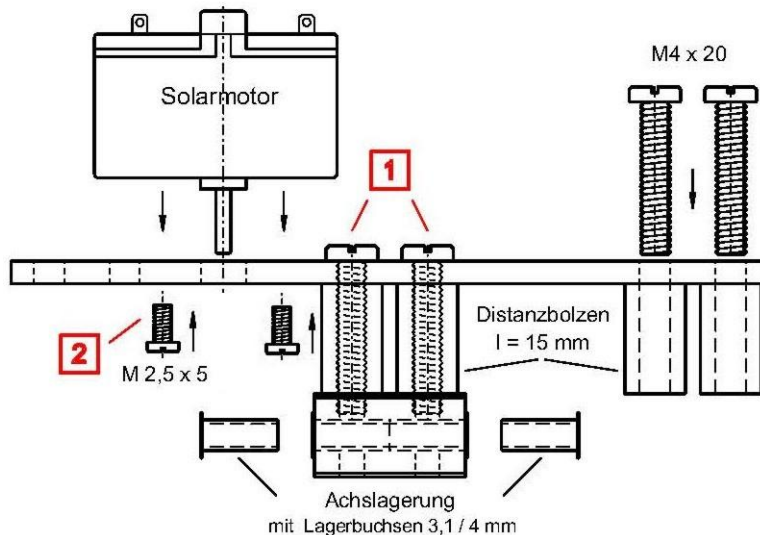
*Das Gleiche gilt auch für den IR-Empfänger und den Distanzmesser, sofern eingesetzt.*

Nähere Info finden Sie im Kapitel "Aufbau der Platine".

## Montage der mechanischen Komponenten

- Befestigung des mittleren Achslagers und der Solarmotoren  
(Befestigung der Getriebemotoren siehe unten)

Die Achslager werden mit je 2 Schrauben M4 X 20mm und Distanzrollen M4 X 15mm befestigt. Zunächst wird das mittlere Lager **1** befestigt. Die beiden Motoren werden mit je 2 Schrauben M2,5 X 5mm befestigt.



- Befestigung des Vorderrads (Holzkugel)

Schraube M4 X 40, Distanzrolle M4 X 12, Mutter M4  
Die Holzkugel D=30mm wird mit 3,3mm mittig ca. 25mm tief vorgebohrt. Mit einem Gewindebohrer M4 wird ein Gewinde in die Bohrung geschnitten, damit die Kugel wie in der Abb. aufgeschraubt werden kann.

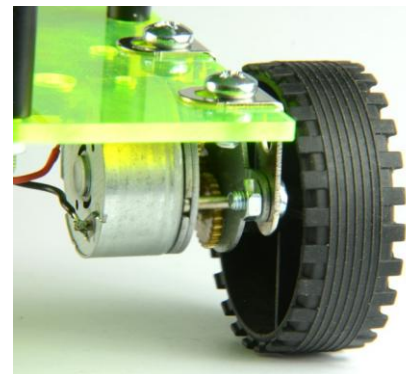
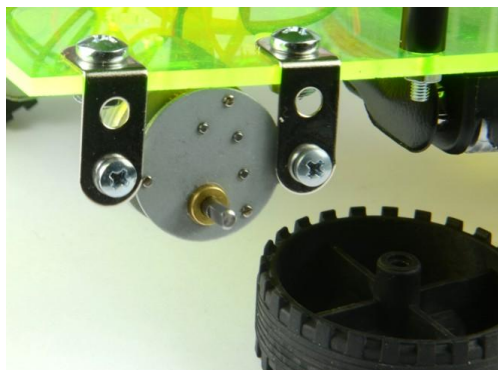
*Alternativ kann die Holzkugel 30mm mit 3,9mm ca. 25mm tief vorgebohrt und aufgesteckt werden.*



Die Holzkugel hat sich sehr bewährt. eine Drehung auf der Stelle ist hiermit sehr viel genauer als mit drehbaren Vorderrädern.

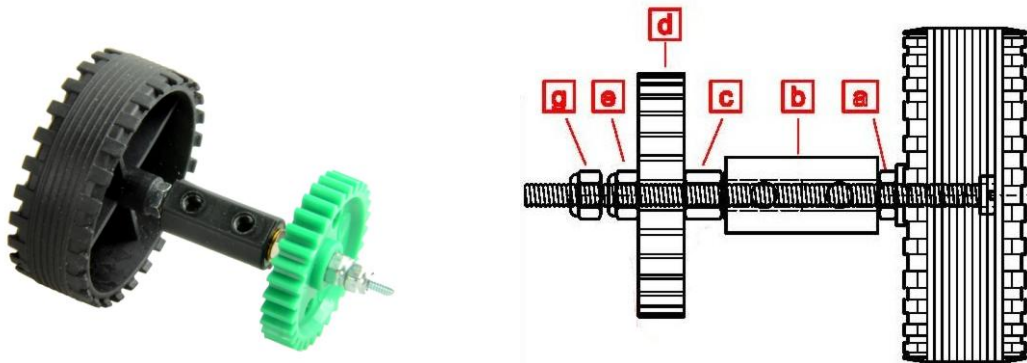
- Befestigung der Getriebemotoren (alternativer Präzisions-Flüsterantrieb)

Die Getriebemotoren werden mit je 2 Winkeln befestigt. Hierfür werden die Schrauben M4 X 8 und M 3X 8 verwendet. Die Reifen werden direkt aufgesteckt.



- Montage der Antriebsachsen

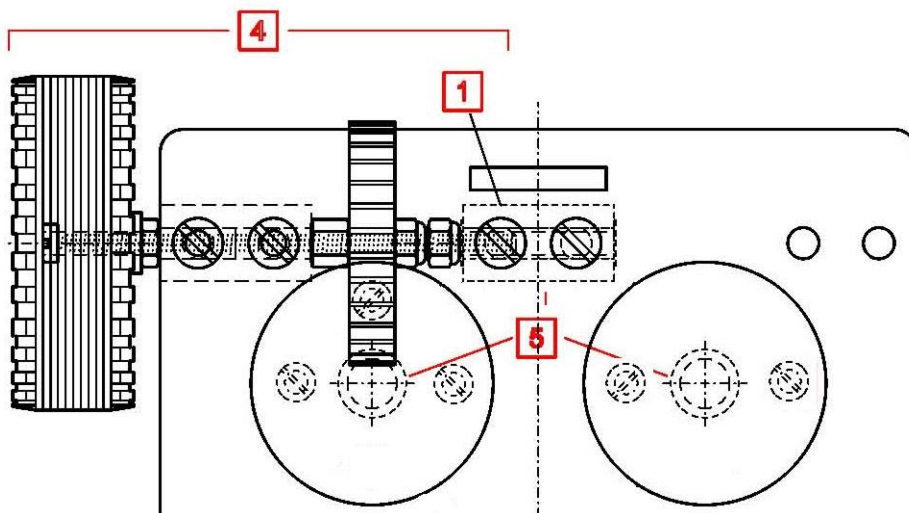
Antriebsrad auf die Schraube M3 X 60 stecken und mit der Mutter M4 **a** festschrauben. Von beiden Seiten Lagerbuchsen in das Achslager **b** stecken und ebenfalls auf die Schraube stecken. Abstandsbolzen 5mm **c** auf die Schraube drehen bis sich die Achse gerade noch gut drehen lässt. Zahnrad **d** aufstecken und mit Sicherungsmutter **e** fixieren. Dabei den Abstandsbolzen mit einem Gabelschlüssel halten. Wenn das Zahnrad fest sitzt, die Achse sich leicht, aber nicht zu lose drehen lässt, die zweite Sicherungsmutter **g** eindrehen.



- Befestigen der Antriebsachsen

Montierte Antriebsachsen in das mittlere Achslage stecken und mit den Schrauben M4 X 20 leicht befestigen. Dabei wieder Distanzrollen M4 X 15mm verwenden. Beide Antriebsachsen müssen sich jetzt leicht drehen lassen. Gegebenenfalls die Achslager korrigieren und erst dann festdrehen.

**Achtung!** Die Schrauben dürfen nicht zu fest angezogen werden, da die Achslager aus Kunststoff sind und die Gewinde sonst Schaden nehmen könnten.



- Befestigen der Antriebsschnecken

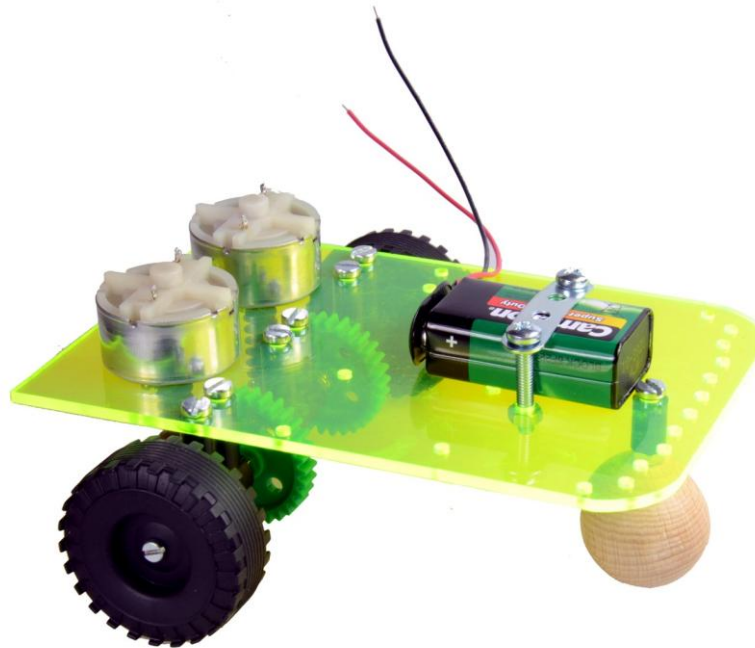
Die Antriebsschnecken werden auf die Motorachse so aufgesteckt, dass sie noch einen Abstand von ca. 2mm zum Motorgehäuse haben. Die Schnecken müssen sich leicht drehen lassen. Evtl. Die Motorschrauben etwas lösen und korrigieren.

- Stromversorgung befestigen

Das Roboterfahrzeug kann sowohl mit einer 9V Batterie (Standard) als auch mit 6 X 1,5V Batterie AA betrieben werden.

*Werden AA-Batterien oder Akkus verwendet ist es wichtig, vor der Montage des Gesamtfahrzeugs zu entscheiden, ob die Batteriehalter oben oder unten angebracht werden sollen. Die Maßzeichnung auf Seite 2 sieht die Anbringung von unten vor. Andernfalls wird die Grundplatte zur Montage gewendet.*

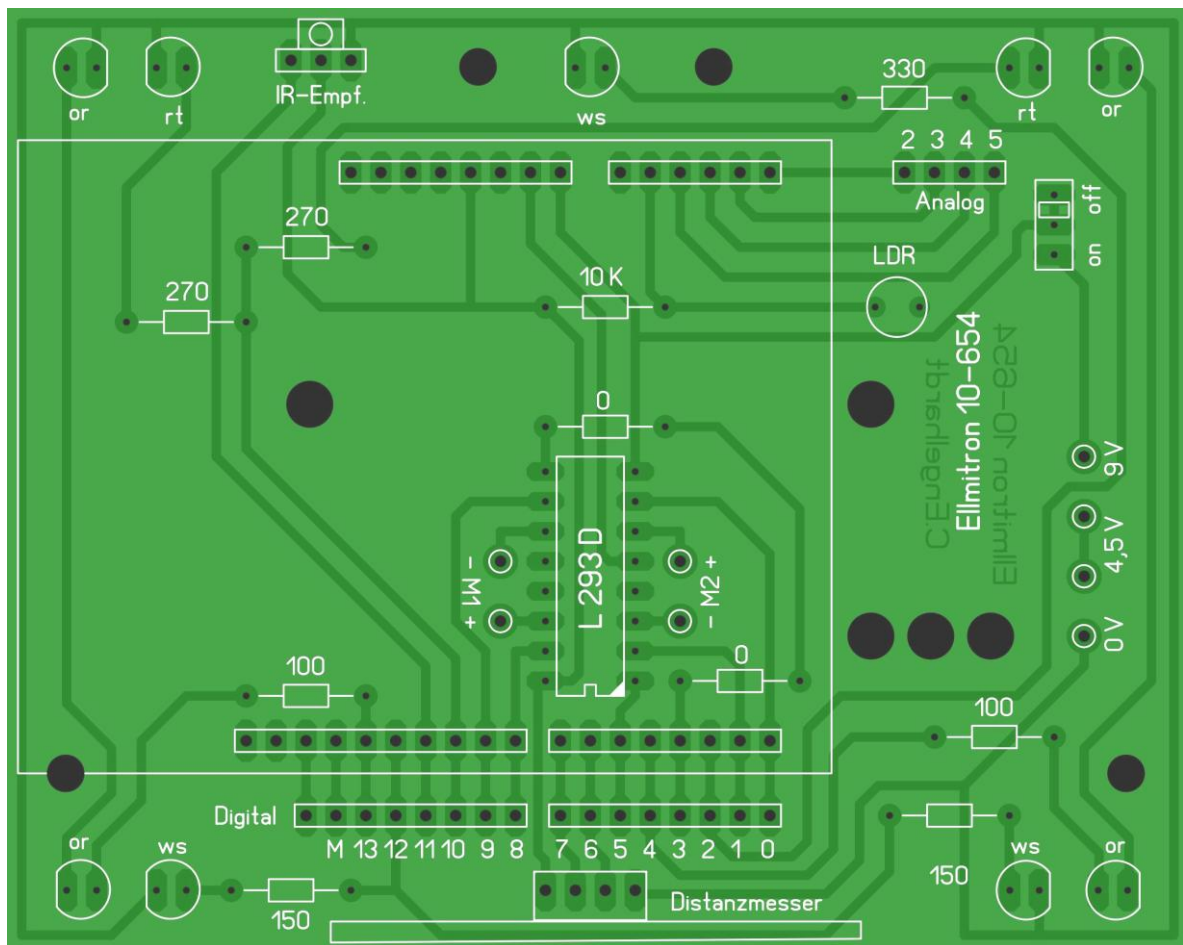
Die 9 Volt Batterie (Standard) wird mit einem Flachstab 4-Loch und 2 Schrauben M4 X 25 befestigt. Die Schrauben werden ohne Distanzrolle verwendet, damit die Batterie geklemmt werden kann. Wenn man das Fahrzeug von vorne betrachtet, sollte der Minusanschluss der Batterie nach rechts schauen, damit der Batterieclip durch das entsprechende Loch der Platine geführt werden kann.



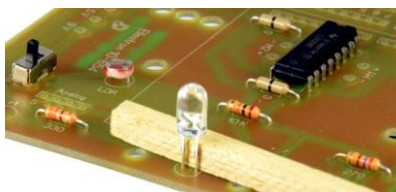
*Die beiden Batteriehalter für je 3 X AA werden, (optional, wenn mehr Leistung gewünscht ist) wie in der Abb. mit je 2 Senkkopfschrauben M3 X 8 und Muttern M3 befestigt. Die Anschlüsse werden auf der Platine in Reihe geschaltet.*



## Bestückung der Platine (für das auf der Titelseite abgebildete Fahrzeug)



Widerstände, LDR und Schiebeschalter einlöten. Dann den L-293-D. Für den L-293-D wird kein Sockel verwendet um die Wärmeabfuhr auf die Platine zu verbessern.



LEDs so einlöten, dass sie einen Abstand von 6mm zur Platine haben. Anschließend nach außen umbiegen (siehe Abb. unten). Beachten Sie die LED-Farbe und die Polung. Die Abgeflachte Seite im Bestückungsdruck entspricht dem kurzen Anschluss der LED (-).

Aus den Stiftleisten werden mit einem Seitenschneider folgende Teilstücke hergestellt:  
1 X 4pol. 1 X 6pol. 4 X 8pol. und 1 X 10pol.

*Die Reste der Stiftleisten kann man verwenden, wenn man die LEDs an anderer Stelle anbringen möchte. Dann 2pol.-Stücke herstellen, anstelle der LEDs einlöten und diese mit Verbindungsleitungen verbinden. Besonders eignen sich hierfür die leicht trennbaren Busleitungen 24-213 St./Bu. und 24-214 Bu./Bu.*

Die Stiftleisten für den Arduino kann man mit aufgestecktem Arduino einlöten, damit sie später genau ausgerichtet sind. Anschließend Arduino wieder entfernen.

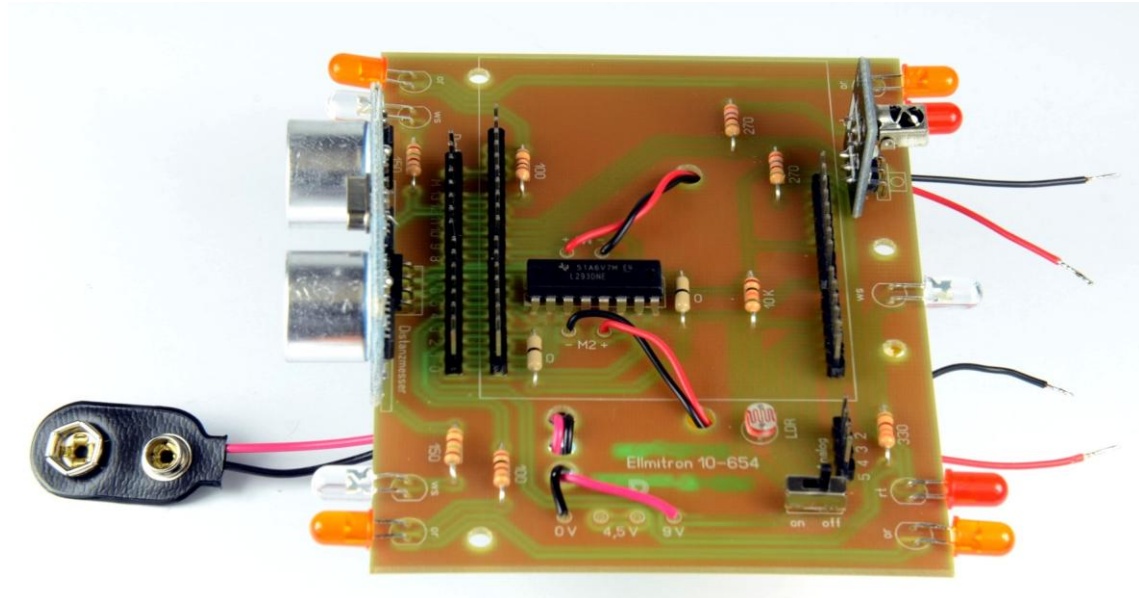
IR-Fernbedienung und Ultraschall-Distanzmesser einlöten.

Für den Anschluss der Motoren (M1 und M2) werden je 2 Leitungen 10cm in rot und schwarz abisoliert, verdreht und verzinkt. Wie in der Abb. verlöten und die Leitungsenden über die beiden großen Bohrungen nach unten führen.

Batterieclip von unten durch ein Loch der Platine nach oben führen (Zur besseren Zugentlastung kann man hierzu auch drei Löcher wie in der Abb. verwenden). Der schwarze Anschluss wird an 0 V, der rote an 9 V gelötet.

*Werden AA-Batterien und 2 Batteriekästen verwendet, lötet man den einen Batterieclip an 0V und 4,5V, den anderen an 4,5V und 9V. So sind beide Batteriekästen in Reihe geschaltet.*

## Fertig bestückte Platine



So sollte die Platine aussehen, wenn das Roboterfahrzeug auf der Titelseite gebaut werden soll.

Motorleitungen an die Motoren löten. Dabei jeweils die rote Leitung an den + Anschluss.

*Wenn Sie ein Fahrzeug ohne Schneckenantrieb bauen (z.B. GT-60 oder GT-180), muss die rote Leitung bei Motor 2 an den - Anschluss gelötet werden.*

## Montage der Trägerplatine auf dem Fahrzeug

Die Trägerplatine wird mit den Schrauben M3 X 40 und den Distanzrollen 3 X 30mm mit Muttern M3 befestigt.

## Aufstecken des Arduino und Funktionsprüfung

Nachdem der Arduino vorsichtig auf die Trägerplatine gesteckt wurde, ist er mit allen Fahrzeugkomponenten verbunden und Sie können die Testprogramme ausprobieren. Dazu am besten das Fahrzeug auf eine "Hebebühne" stellen (z.B. 2 Klötzchen 5cm), damit die Räder sich frei drehen können. Dann den Arduino mit dem Computer verbinden und das gewünschte Testprogramm aufspielen.

Die Testprogramme können per Copy-Paste in die Arduino-Oberfläche kopiert und dort auf den Arduino übertragen werden.

Wenn Sie länger mit dem Fahrzeug am Computer arbeiten und Programme mit Motoren testen und optimieren, empfiehlt es sich, ein Netzgerät (9 V) zu verwenden um die Batterie zu schonen. Wenn ein Netzgerät verwendet wird, bitte den Schalter am Fahrzeug auf OFF schalten oder die Batterie abklemmen.

Für Programme, die keine Motoren verwenden, genügt die USB-Spannung.



Wenn auf der Hebebühne alles korrekt funktioniert hat, wird das USB-Kabel und das Netzteil entfernt. Jetzt kann man das Roboter-Fahrzeug mit genügend Bewegungsfreiheit auf den Boden setzen und arbeiten lassen. Die Testprogramme wiederholen sich, bis der Schalter in Stellung "OFF" gebracht wird.

Alle Testprogramme können beliebig abgeändert werden. Dazu am besten das Fahrzeug wieder auf die Hebebühne stellen, den Arduino mit dem Computer verbinden und loslegen. Nach dem Hochladen sind die Änderungen sofort sichtbar.

## Programmieren des Roboter-Fahrzeugs mit dem Arduino

Für den Einstieg und die grundlegende Programmierung des Arduino gibt es gute Literatur (z.B. Arduino-Handbuch für Einsteiger Best.Nr.05-030 oder Roboter bauen mit Arduino Best.Nr.05-031) sowie die umfangreiche Referenz, die man in der Hilfe der Programmieroberfläche aufrufen, und die Elegoo-Handreichung, die man sich kostenlos bei Elegoo herunterladen kann. In letzterer werden alle im Sensorkit / Starterkit enthaltenen Bauteile besprochen (Deutsch).

Hier sollen nur die Programmelemente angesprochen werden, die sich auf das Roboterfahrzeug beziehen. Zunächst ist es wichtig, im Einzelnen zu wissen, welches Bauteil mit welchem Pin verbunden ist und welche Möglichkeiten es hat. Das Zeichen ~ bedeutet, dass es sich bei dem betreffenden Pin um einen PWM-Pin handelt.

- Frontscheinwerfer, weiß (D-Pin 12)  
Beide Scheinwerfer sind parallel geschaltet. Funktion: Ein/Aus
- Rückfahrscheinwerfer, rot (D-Pin 10~)  
Beide Scheinwerfer sind parallel geschaltet und können gedimmt werden. Z.B. bei normaler Fahrt mittelhell, als Bremslicht hell
- Rückfahrscheinwerfer, weiß (D-Pin 2)  
Funktion: Ein/Aus
- Blinklicht rechts, gelb (Pin D-13)  
Beide Blinklichter rechts (vorne und hinten) sind in Reihe geschaltet.  
Funktion: Ein/Aus oder Blinken über Programm
- Blinklicht links, gelb (Pin D-4)  
Beide Blinklichter links (vorne und hinten) sind in Reihe geschaltet.  
Funktion: Ein/Aus oder Blinken über Programm
- Motor 1, rechts (Pin D-8 D-9 D-3~)  
Der Motor wird über einen L293 gesteuert.  
Funktion: Vorwärts/Rückwärts, Beschleunigen/Bremsen, Stopp
- Motor 2, links (Pin D-0 D-1 D-5~)  
Der Motor wird über einen L293 gesteuert.  
Funktion: Vorwärts/Rückwärts, Beschleunigen/Bremsen, Stopp
- Fotosensor (Analog-Pin 1)  
Funktion: Der Fotosensor misst die Umgebungshelligkeit. Damit kann z.B. ab einem bestimmten Schwellwert die Fahrzeugbeleuchtung ein/aus geschaltet werden.
- Ultraschall-Distanzmesser (D-Pin 6~) Rückmeldung, (D-Pin 7) Trigger  
Funktion: Gibt die Distanz vom Distanzmesser zu einem Objekt (Wand) in cm aus und kann im Programm für automatisches Bremsen/Anhalten verwendet werden.
- IR-Fernbedienung (D-Pin 11~)  
Funktion: Mit dem Sendemodul können 17 Codes erzeugt werden, die für beliebige Funktionen genutzt werden können (z.B. vor/rück, rechts/links, bremsen, beschleunigen, auf der Stelle drehen, Warnblinker etc.)

Nachfolgend finden Sie kleine Programmbeispiele für die oben genannten Funktionen. Rot ist der eigentliche Programmtext, Blau die Erklärung, der Rest ist vorgegeben. Die Beschreibung ist kursiv.

Am besten ist es, das Fahrzeug während der Programmierung und zum ersten Test auf die Hebebühne zu stellen. Wenn der Arduino mit dem Computer verbunden ist kann das Fahrzeug so nicht eigenmächtig vom Tisch fahren. Nach dem Hochladen sehen Sie sofort, was das Programm bewirkt. Sie können dann das Fahrzeug vom Computer trennen und auf dem Boden mit Batterie testen.

## Testprogramme

### 1. LED ein / aus / Blinklicht

*Im folgenden Programm wird der rechte Blinker (D-Pin 13) programmiert. Er wird mit dem Befehl `digitalWrite(x,y)` angesteuert. `x` ist dabei die Pin-Nummer (hier 13) und `y` ist der gewünschte Ausgabewert. 0=aus, 1=ein. In diesem Programm wird die LED für 0,3 Sekunden ein- und wieder ausgeschaltet. Da das Programm ständig wiederholt wird, entsteht ein Blinklicht.*

*Wird Pin 13 durch Pin 4 ersetzt, wird der linke Blinker angesteuert.*

*Sie können auch beide Pins ansprechen um einen Warmlinker zu erhalten. Versuchen Sie auch, einen Wechselblinker zu programmieren.*

```
void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT); // Macht Pin 13 zu einem Output
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13,1); // Schaltet Pin 13 auf 5V
  delay(300); // Pause für 0,3 Sekunden (300msec)
  digitalWrite(13,0); // Schaltet Pin 13 auf 0V
  delay(300); // Pause für 0,3 Sekunden (300msec)
}
```

### 2. LED in zwei Helligkeitsstufen, hier Rücklicht/Bremslicht

*Im folgenden Programm wird D-Pin 10~ mit dem Befehl `analogWrite(x,y)` angesteuert, damit das Rücklicht unterschiedlich hell leuchten kann. `x` ist dabei die Pin-Nummer (hier 10) und `y` ist der gewünschte Ausgabewert zwischen 0 und 255. 0=dunkel, 255=hell. Experimentieren Sie mit verschiedenen Werten. Achtung! Analog können nur PWM-Ausgänge wie D-Pin 10~ angesteuert werden. Bei den LEDs ist das nur das Rücklicht. Alle anderen LEDs können nur digital (ein/aus) geschaltet werden wie in Programm 1.*

```
void setup() {
  pinMode(10,OUTPUT); // Macht Pin 10 zu einem Output
}

void loop() {
  analogWrite(10,63); // Schaltet Pin 10 auf viertel Helligkeit (Rücklicht)
  delay(1000); // Pause für 1,0 Sekunden (1000msec)
  analogWrite(10,255); // Schaltet Pin 10 auf volle Helligkeit (Bremslicht)
  delay(500); // Pause für 0,5 Sekunden (500msec)
  analogWrite(10,0); // Schaltet Pin 10 auf 0V (aus)
  delay(1000); // Pause für 1,0 Sekunden (1000msec)
}
```

### 3. Motor 1 (rechtes Rad) vor / rück, schnell / langsam

Im folgenden Programm wird Motor 1, rechtes Rad (Pin D-8 D-9 D-3~) angesteuert. Dabei geben Pin 8 und Pin 9 die Drehrichtung vor und über Pin 3 wird die Geschwindigkeit gesteuert (0=aus, 255=maxSpeed. Der Bereich zwischen 1 und 119 macht wenig Sinn, da die Kraft des Motors zum Antrieb des Fahrzeugs noch nicht ausreicht). Das rechte Rad wird erst langsam vorwärts gedreht, dann schnell rückwärts.

Experimentieren Sie auch hier mit verschiedenen Werten. Auf die gleiche Weise wird Motor 2, linkes Rad (Pin D-0 D-1 D-5~) angesteuert. Versuchen Sie, die Räder so anzusteuern, dass das Fahrzeug gerade vorwärts/rückwärts fährt, auf der Stelle rechts oder links herum dreht, vorwärts/rückwärts nach rechts oder links lenkt.

```
void setup() {
  pinMode(8,OUTPUT); // Macht Pin 8 zu einem Output
  pinMode(9,OUTPUT); // Macht Pin 9 zu einem Output
  pinMode(3,OUTPUT); // Macht Pin 3 zu einem Output
}

void loop() {
  digitalWrite(8,1); digitalWrite(9,0); // Motor 1 vorwärts
  analogWrite(3,127); // Motorgeschwindigkeit 127 (langsam)
  delay(4000); // Pause für 4,0 Sekunden
  analogWrite(3,0); // // Motor 1 aus
  delay(2000); // Pause für 2,0 Sekunden
  digitalWrite(8,0); digitalWrite(9,1); // Motor 1 rückwärts
  analogWrite(3,255); // Motorgeschwindigkeit 255 (schnell)
  delay(4000); // Pause für 4,0 Sekunden
  analogWrite(3,0); // // Motor aus
  delay(2000); // Pause für 2,0 Sekunden
}
```

### 4. Beschleunigen und Bremsen

Im folgenden Programm soll das Fahrzeug erst vorwärts, dann rückwärts beschleunigen und wieder bremsen. Hierbei wird der Analogwert von Pin 5 (Geschwindigkeit rechtes Rad) und Pin 3 (Geschwindigkeit linkes Rad) in einer Schleife langsam erhöht bzw. reduziert. Die Brems-/Beschleunigungszeit ist durch den Wert in der Delay-Zeile vorgegeben. Experimentieren Sie hier, um eine realistische Brems- bzw. Beschleunigungszeit zu erreichen.

```
void setup() {
  pinMode(8,OUTPUT); // Macht Pin 8 zu einem Output
  pinMode(9,OUTPUT); // Macht Pin 9 zu einem Output
  pinMode(3,OUTPUT); // Macht Pin 3 zu einem Output
  pinMode(0,OUTPUT); // Macht Pin 0 zu einem Output
  pinMode(1,OUTPUT); // Macht Pin 1 zu einem Output
  pinMode(5,OUTPUT); // Macht Pin 5 zu einem Output
}

void loop() {
  // Beschleunigen vorwärts
  digitalWrite(8,1); digitalWrite(9,0); // Motor 1 vorwärts
  digitalWrite(0,1); digitalWrite(1,0); // Motor 2 vorwärts
  for(int i = 120; i <255; i++){ // Festlegung des Wertebereiches (100-255)
    analogWrite(5, i);analogWrite(3, i); // Beschleunigen vorwärts
    delay(20);
  }
  delay(2000); // 2,0 Sekunden schnelle Fahrt
  // Bremsen vorwärts
  for(int i = 255; i >120; i--){ // Festlegung des Wertebereiches (255-100)
```

```

        analogWrite(5, i);analogWrite(3, i); // Bremsen vorwärts
        delay(20);
    }
    analogWrite(5, 0);analogWrite(3, 0); // Anhalten
    delay(4000); // Pause für 4,0 Sekunden (4000msec)
    // Beschleunigen rückwärts
    digitalWrite(8,0); digitalWrite(9,1); // Motor1drehrichtung rechts (rückwärts)
    digitalWrite(0,0); digitalWrite(1,1); // Motor2drehrichtung rechts (rückwärts)
    for(int i = 100; i <255; i++) { // Festlegung des Wertebereiches (100-255)
        analogWrite(5, i);analogWrite(3, i); // Beschleunigen rückwärts
        delay(20);
    }
    delay(2000); // 2,0 Sekunden schnelle Fahrt
    // Bremsen rückwärts
    for(int i = 255; i >100; i--) { // Festlegung des Wertebereiches (255-100)
        analogWrite(5, i);analogWrite(3, i); // Bremsen vorwärts
        delay(20);
    }
    analogWrite(5, 0);analogWrite(3, 0); // Anhalten
    delay(4000); // Pause für 4,0 Sekunden (4000msec)
}

```

## 5. Beleuchtung steuern mit dem Lichtsensor

Im folgenden Programm wird die Beleuchtung des Fahrzeugs durch Auswerten des Lichtsensors (LDR) automatisch ein- und ausgeschaltet. Der LDR ist auf der Platine fest verbaut und mit dem Analog-Pin 1 des Arduino verbunden.

Der Wertebereich des Analogeingangs liegt zwischen 0 und 5 Volt. Im Programm werden entsprechend Werte zwischen 0 und 1023 gemessen. Unser Sensor erzeugt Werte zwischen ca. 20 (maximal beleuchtet) und ca. 1000 (völlig dunkel).

Das Licht soll nun eingeschaltet werden, sobald der gemessene Wert größer als 500 (Schwellwert) ist. Damit das Licht nicht flackert, wenn sich die Lichtwerte schnell um den Schwellwert bewegen, wird die Beleuchtung erst wieder ausgeschaltet, wenn der gemessene Wert unter 300 liegt, jedoch frühestens nach 3 Sekunden. Experimentieren Sie mit unterschiedlichen Schwellwerten und versuchen Sie, das Programm zu optimieren.

Wenn Sie wissen möchten, welche Werte Ihr Sensor bei einer bestimmten Helligkeit erzeugt, können Sie die grauen Zeilen verwenden um den Wert im seriellen Monitor (Lupensymbol in der Menüleiste) anzeigen zu lassen. Für den Betrieb des Fahrzeugs müssen die Zeilen wieder gelöscht oder mit // am Anfang der Zeile als Kommentar deklariert werden.

```

void setup() {
    Serial.begin(9600); // Diese Zeile ist nur zu Testzwecken gedacht
    pinMode(10,OUTPUT); // Macht Pin 10 zu einem Output
    pinMode(12,OUTPUT); // Macht Pin 12 zu einem Output
}

void loop() {
    int LDR = analogRead(A1); // Analogwert wird ausgelesen
    Serial.println(LDR); // Diese Zeile ist nur zu Testzwecken gedacht
    if (LDR>500){
        analogWrite(10,63); digitalWrite(12,1); // Beleuchtung ein
        delay(3000); // Pause für 3,0 Sekunden (3000msec)
    }
    if (LDR<300){
        analogWrite(10,0); digitalWrite(12,0); // Beleuchtung aus
        delay(100); // Pause für 0,1 Sekunden (100msec)
    }
}

```

## 6. Entfernung messen mit dem Ultraschall-Entfernungssensor (UES)

Im folgenden Programm wird der UES verwendet, um die Entfernung zwischen Sensor und einem Objekt (Hand, Wand etc.) zu erkennen. Das Ergebnis wird im seriellen Monitor (Lupensymbol in der Menüleiste) angezeigt und alle 0,5 Sekunden aktualisiert. Der UES verwendet 2 Pins. (D-Pin 7) Trigger, (D-Pin 6-) Rückmeldung

Bei der Verwendung des UES in einem Arduino-Programm muss zunächst die Bibliothek SR04.h eingebunden werden. Befindet sich diese nicht in Ihrem Bibliotheksordner, muss sie zunächst hinzugefügt werden (Bibliotheken verwalten).

Funktion: Gibt die Distanz vom UES zu einem Objekt in cm aus. Beachten Sie, dass das Objekt eine gewisse Größe haben muss, um erkannt zu werden. Experimentieren Sie hierzu. Auch kann es evtl. sinnvoll sein, auf den US-Empfänger oder Sender eine Hülse aufzustecken um den Sendewinkel oder Empfangswinkel zu reduzieren.

```
#include <SR04.h>
#define TRIG_PIN 7
#define ECHO_PIN 6
SR04 sr04 = SR04(ECHO_PIN,TRIG_PIN);
long a;

void setup() {
  Serial.begin(9600);
  delay(1000);
}

void loop() {
  a=sr04.Distance();
  Serial.print(a);
  Serial.println("cm");
  delay(500);
}
```

## 7. Automatische Gefahrenerkennung mit dem Ultraschall-Entfernungssensor (UES)

Im folgenden Programm wird der UES verwendet, um die Entfernung zwischen Fahrzeug und einem Hindernis zu erkennen. Der Serielle Monitor wird nicht verwendet da sich das Fahrzeug frei bewegen soll. Nach dem Programmstart fährt das Fahrzeug vorwärts. Fährt es auf eine Wand zu, so bleibt es 50cm vor der Wand stehen. Bitte berücksichtigen Sie, dass der Sensor in unserem Fahrzeug ca. 10cm zurückversetzt eingebaut ist. Daher wird im Programm ein Wert von 60cm verwendet. Experimentieren Sie mit verschiedenen Werten. Kleiner als 15cm sollte der Wert allerdings nicht sein, damit für den Bremsweg noch genügend Platz bleibt.

```
#include <SR04.h>
#define TRIG_PIN 7
#define ECHO_PIN 6
SR04 sr04 = SR04(ECHO_PIN,TRIG_PIN);
long a;

void setup() {
  pinMode(8,OUTPUT); // Macht Pin 8 zu einem Output
  pinMode(9,OUTPUT); // Macht Pin 9 zu einem Output
  pinMode(3,OUTPUT); // Macht Pin 3 zu einem Output
  pinMode(0,OUTPUT); // Macht Pin 0 zu einem Output
  pinMode(1,OUTPUT); // Macht Pin 1 zu einem Output
  pinMode(5,OUTPUT); // Macht Pin 5 zu einem Output
}

void loop() {
  a=sr04.Distance();
  digitalWrite(8,1); digitalWrite(9,0); // Motor 1 vorwärts
```

```

digitalWrite(0,1); digitalWrite(1,0); // Motor 2 vorwärts
if (0<a<51){
    digitalWrite(5, 0);digitalWrite(3, 0); // Anhalten
    delay(50);
}
if (a>50){
    digitalWrite(5, 1);digitalWrite(3, 1); // Fahren
    delay(50);
}
}

```

## 8. Selbstfahrendes Roboterfahrzeug mit Gefahrerkennung (mit UES)

Im folgenden Programm wird das obenstehende Programm erweitert. Bremslicht beim Bremsen, kurzes Rückwärtsfahren mit Rückfahrcheinwerfer, Drehen des Fahrzeugs bis Entfernung zu nächstem Hindernis >70cm, dann wieder vorwärtsfahren. Bei Hinderniserkennung in 70cm Abstand bremsst das Fahrzeug bereits ab.

Experimentieren Sie mit verschiedenen Werten.

```

#include <SR04.h>
#define TRIG_PIN 7
#define ECHO_PIN 6
SR04 sr04 = SR04(ECHO_PIN,TRIG_PIN);
long a;

void setup() {
    pinMode(8,OUTPUT); // Macht Pin 8 zu einem Output
    pinMode(9,OUTPUT); // Macht Pin 9 zu einem Output
    pinMode(3,OUTPUT); // Macht Pin 3 zu einem Output
    pinMode(0,OUTPUT); // Macht Pin 0 zu einem Output
    pinMode(1,OUTPUT); // Macht Pin 1 zu einem Output
    pinMode(5,OUTPUT); // Macht Pin 5 zu einem Output
    pinMode(2,OUTPUT); // Macht Pin 2 zu einem Output
    pinMode(10,OUTPUT); // Macht Pin 10 zu einem Output
    delay(500);
}
void loop() {
    a=sr04.Distance();
    if (a>0 and a<41){
        digitalWrite(5, 0); digitalWrite(3, 0); // Anhalten
        digitalWrite(10,1); // Bremslicht ein
        delay(1000);
        digitalWrite(2,1); // Rückfahrlicht ein
        delay(500);
        digitalWrite(10,0); // Bremslicht aus
        delay(500);
        digitalWrite(8,0); digitalWrite(9,1); // Motor 1 rückwärts
        digitalWrite(0,0); digitalWrite(1,1); // Motor 2 rückwärts
        digitalWrite(5, 1);digitalWrite(3, 1); // Fahren
        delay(200);
        digitalWrite(5,0); digitalWrite(3,0); // Anhalten
        digitalWrite(10,1); // Bremslicht ein
        delay(500);
        digitalWrite(2,0); // Rückfahrlicht aus
        digitalWrite(10,0); // Bremslicht aus
        delay(500);
        digitalWrite(8,0); digitalWrite(9,1); // Motor 1 rückwärts
    }
}

```

```

digitalWrite(0,1); digitalWrite(1,0); // Motor 2 vorwärts
digitalWrite(5,1); digitalWrite(3,1); // Fahren
delay(200);
digitalWrite(5,0); digitalWrite(3,0); // Anhalten
digitalWrite(10,1); // Bremslicht ein
delay(500);
digitalWrite(10,0); // Bremslicht aus
delay(500);
}
if (a>40){
digitalWrite(8,1); digitalWrite(9,0); // Motor 1 vorwärts
digitalWrite(0,1); digitalWrite(1,0); // Motor 2 vorwärts
digitalWrite(5,1);digitalWrite(3,1); // Fahren
}
}

```

## 9. Ferngesteuertes Roboterfahrzeug mit IR-Fernbedienung

Hier wird das IR-Empfängermodul verwendet. Mit der Handfernbedienung lassen sich 17 Codes erzeugen, die jeweils ein Unterprogramm aufrufen. Im folgenden Programm werden alle Codes exemplarisch aufgeführt. Dadurch kann das Fahrzeug und die Beleuchtung gesteuert werden. Überlegen Sie sich eigene Funktionen und ordnen Sie diese den Tasten zu.

```

#include <IRremote.h>
int receiver = 11; // Signal-Pin von IR receiver an D-Pin 11~
int FB;
IRrecv irrecv(receiver); // create instance of 'irrecv'
decode_results results; // create instance of 'decode_results'

void translateIR(){ // Übersetzung des IR-Codes, der empfangen wird
// Taste an der Fernbedienung = Aktion am Fahrzeug

digitalWrite (2,0);
digitalWrite (3,0);
digitalWrite (4,0);
digitalWrite (5,0);
digitalWrite (10,0);
digitalWrite (12,0);
digitalWrite (13,0);

switch(results.value){

case 0xFFA25D: ;// Taste 1 = Blinker links 5 X blinken
for(int i=0;i<5;i++){
digitalWrite (4,1);
delay(300);
digitalWrite(4,0);
delay(300); }
break; // 1

case 0xFF629D: ; // Taste 2 = Rückfahrscheinwerfer
digitalWrite (2,1);
break; // 2

case 0xFFE21D: ; // Taste 3 = Blinker rechts 5 X blinken
for(int i=0;i<5;i++){
digitalWrite (13,1);
delay(300);
}
}
}

```

```

digitalWrite (13,0);
delay(300); }
break; // 3

case 0xFF22DD: ; // Taste 4 = Scheinwerfer vorne
digitalWrite (12,1);
break; // 4

case 0xFF02FD: ; // Taste 5 = Rücklicht
analogWrite (10,50);
break; // 5

case 0xFFC23D: ; // Taste 6 = Bremslicht
analogWrite (10,255);
break; // 6

case 0xFFE01F: ; // Taste 7 = Hier ohne Funktion
break; // 7
case 0xFFA857: ; // Taste 8 = Hier ohne Funktion
break; // 8
case 0xFF906F: ; // Taste 9 = Hier ohne Funktion
break; // 9

case 0xFF9867: ; // Taste 0 = Warnblinker 5 X Blinken
for(int i=0;i<5;i++){
digitalWrite (4,1); digitalWrite (13,1);
delay(300);
digitalWrite (4,0); digitalWrite (13,0);
delay(300); }
break; // 0

case 0xFFB04F: ; // Taste # = Drehen auf der Stelle rechts
digitalWrite (8,0); digitalWrite (9,1); digitalWrite (3,1);
digitalWrite (0,1); digitalWrite (1,0); digitalWrite (5,1);
break; // #

case 0xFF6897: ; // Taste * = Drehen auf der Stelle links
digitalWrite (8,1); digitalWrite (9,0); digitalWrite (3,1);
digitalWrite (0,0); digitalWrite (1,1); digitalWrite (5,1);
break; // *

case 0xFF18E7: ; // Taste ▲ = Vorwärts fahren
digitalWrite (8,1); digitalWrite (9,0); digitalWrite (3,1);
digitalWrite (0,1); digitalWrite (1,0); digitalWrite (5,1);
break;

case 0xFF10EF: ; // Taste ◀ = Nach links lenken
digitalWrite (8,1); digitalWrite (9,0); digitalWrite (3,1);
break;

case 0xFF5AA5: ; // Taste ▶ = Nach rechts lenken
digitalWrite (0,1); digitalWrite (1,0); digitalWrite (5,1);
break;

case 0xFF4AB5: ; // Taste ▼ = Rückwärts fahren
digitalWrite (8,0); digitalWrite (9,1); digitalWrite (3,1);

```



```

digitalWrite (0,0); digitalWrite (1,1); digitalWrite (5,1);
break;

case 0xFF38C7: ; // Taste OK = Stopp alle Funktionen
digitalWrite (2,0); digitalWrite (3,0); digitalWrite (4,0);
digitalWrite (5,0); digitalWrite (10,0); digitalWrite (12,0);
digitalWrite (13,0);
break;

} // Ende der Tastenauswertung

delay(500); // Pause

} // Ende des Übersetzungs-Unterprogramms

void setup() // Setup
{

irrecv.enableIRIn(); // Start the receiver
int rechts;
int links;

pinMode(0,OUTPUT);
pinMode(1,OUTPUT);
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(12,OUTPUT);
pinMode(10,OUTPUT);
pinMode(13,OUTPUT);
}/*--(end setup )---*/

void loop() /*----( LOOP: RUNS CONSTANTLY )----*/
{
  if (irrecv.decode(&results)) // Abfrage: wurde ein Signal empfangen?
  {
    translateIR();
    irrecv.resume(); // Erhalte den nächsten Wert
  }
}

```

---

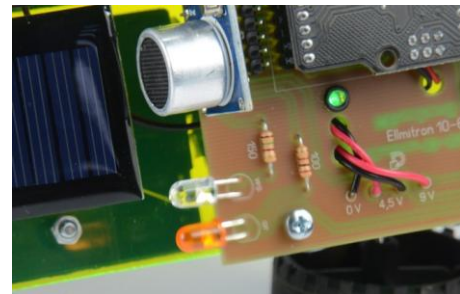
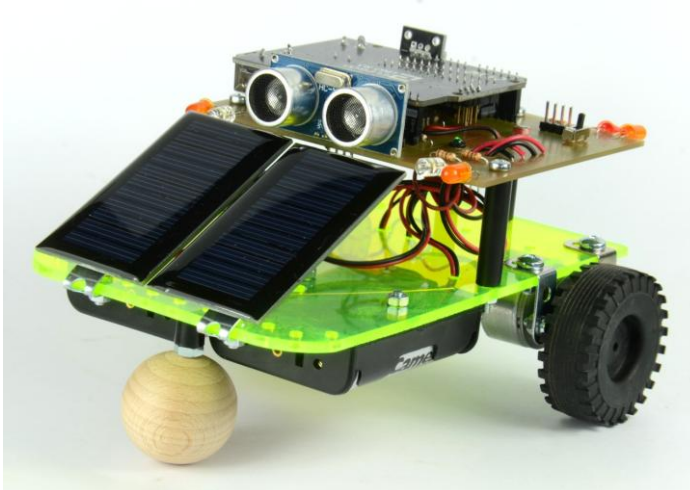
Sollten in nächster Zeit weitere Programme entstehen, werde ich sie auf [www.ellmitron.de](http://www.ellmitron.de) bereitstellen. Über Programme, die Sie gerne teilen möchten, würde ich mich sehr freuen. Sie helfen damit Kollegen, vielfältiger mit dem Arduino zu arbeiten.

Ihr Carsten Engelhardt

## Anhang

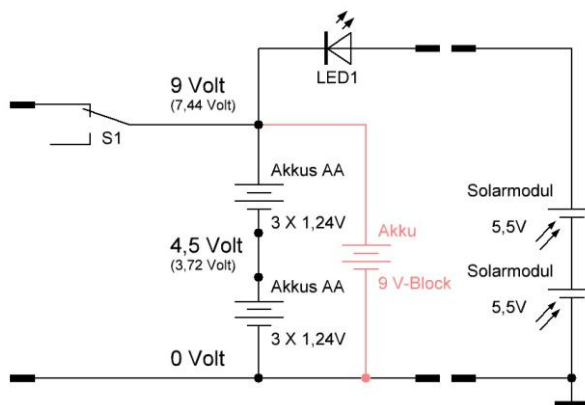
### 1. Solar-Ergänzung

Zur Ergänzung der Roboterfahrzeuge mit Akkubetrieb (9 Volt oder 6 X AA) gibt es den Solar-Ergänzungspack 10-654-SE. Mit dieser Ergänzung ist das Fahrzeug incl. Arduino komplett solarbetrieben. Die Montage ist sehr einfach. Die grüne ultrahelle LED dient dabei gleichzeitig als Lade-Diode und -Anzeige.



Geeignet für die Anbringung der Diode ist das innere Loch wie in der Abb. oben. Dazu muss es auf 4,4mm aufgefeilt werden. Andere Orte sind natürlich auch möglich.

### Schaltplan für die Solar-Ergänzung



Die beiden Solarmodule werden in Reihe geschaltet und über die LED mit dem +9V-Anschluss, die Minusleitung mit dem 0V-Anschluss auf der Platine verbunden. Dies kann auf der Unterseite der Platine geschehen.

Es können sowohl 6 X AA-Akkus oder ein 9V-Block-Akku geladen werden.

Nach Gebrauch wird das Fahrzeug einfach an eine gut beleuchtete Stelle gestellt.

### 2. Stücklisten

Materialpack	Antrieb	Stromversorgung	Best.Nr.
Roboterfahrzeug S1	Schneckenantrieb mit 2 X Solarmotor	9 Volt Batterie oder Akku	<b>10-654-1</b>
Roboterfahrzeug S2	Schneckenantrieb mit 2 X Solarmotor	6 X AA Batterien oder Akkus	<b>10-654-2</b>
Roboterfahrzeug G1	Direktantrieb mit 2 X Solar-Getriebemotor	9 Volt Batterie oder Akku	<b>10-654-G1</b>
Roboterfahrzeug G2	Direktantrieb mit 2 X Solar-Ggetriebemotor	6 X AA Batterien oder Akkus	<b>10-654-G2</b>
Solar-Ergänzung	2 X Solarmodul 5,5 Volt + Material	Nur für Akkubetrieb 9V oder AA	<b>10-654-SE</b>

Stückliste - Grundbauteile - Fahrzeug					RF-1	RF-2	RF-G1	RF-G2	SE
Anzahl	Bezeichnung	Material / Spezifikation	Maße (in mm)	Best.Nr.	x	x	x	x	
1	Grundplatte Fahrzeug	Acryl, neongrün	150 X 100 X 3	25-7182	x	x	x	x	
1	Holzkuigel	D = 30mm ohne Bohrung		34-062	x	x	x	x	
2	Räder	Kunststoff für Achse 3mm	D = 43mm	54-350	x	x	x	x	
4	Distanzrolle	Kunststoff für Schraube M3	L = 30mm	42-6444	x	x	x	x	
1	Distanzrolle	Kunststoff für Schraube M4	L = 12mm	42-6344	x	x	x	x	
1	Zylinderkopfschraube	Kreuzschlitz M4	L = 40mm	53-440	x	x	x	x	
4	Zylinderkopfschraube	Kreuzschlitz M3	L = 40mm	53-340	x	x	x	x	
1	Mutter	M4		40-041	x	x	x	x	
4	Mutter	M3		40-031	x	x	x	x	

Stückliste - Grundbauteile - Elektronik					RF-1	RF-2	RF-G1	RF-G2	SE
Anzahl	Bezeichnung	Material / Spezifikation	Maße (in mm)	Best.Nr.					
1	Trägerplatte	Für Arduino	100mm X 80mm	10-654 PL	x	x	x	x	
1	Schiebeschalter	Print-Schiebeschalter 1 X UM	RM 2,5	47-209	x	x	x	x	
1	LDR	Fotowiderstand A-9013	D = 5mm	98-621	x	x	x	x	
2	Stiftreihe	40pol.	RM 2,5	43-202	x	x	x	x	
2	Widerstand	Kohleschicht 1/3 Watt	0 Ohm	70-800	x	x	x	x	
2	Widerstand	Kohleschicht 1/3 Watt	100 Ohm	70-210	x	x	x	x	
2	Widerstand	Kohleschicht 1/3 Watt	150 Ohm	70-215	x	x	x	x	
2	Widerstand	Kohleschicht 1/3 Watt	270 Ohm	70-410	x	x	x	x	
1	Widerstand	Kohleschicht 1/3 Watt	330 Ohm	70-233	x	x	x	x	
1	Widerstand	Kohleschicht 1/3 Watt	10 KOhm	70-410	x	x	x	x	
1	IC	Treiber IC L-293D für 2 Motoren	16pol.	96-013	x	x	x	x	
2	LED	Leuchtdiode Superhell, Low Current	rot	91-5165	x	x	x	x	
3	LED	Leuchtdiode ultrahell	weiß	91-5461	x	x	x	x	
4	LED	Leuchtdiode Superhell, Low Current	gelb	91-5185	x	x	x	x	
1	Litze	PVC, isoliert, 0,14qmm, rot	50cm	46-000	x	x	x	x	
1	Litze	PVC, isoliert, 0,14qmm, schwarz	50cm	46-001	x	x	x	x	

Stückliste - Antrieb mit Solarmotor + Schneckenantrieb					RF-1	RF-2	RF-G1	RF-G2	SE
Anzahl	Bezeichnung	Material / Spezifikation	Maße (in mm)	Best.Nr.					
2	Zahnräder, grün	Modul 1 für Achse 3mm	30 Zähne	54-3043	x	x			
2	Schnecken	Modul 1 für Achse 2mm	D = 9mm	54-3212	x	x			
3	Wellenkupplung	Kupplung für Achse 4mm	L = 20mm	00-01403	x	x			
6	Lagerbuchse	ID = 3,1 AD = 4,0	L = 10mm	54-1136	x	x			
6	Distanzrolle	Kunststoff für Schraube M4	L = 15mm	42-6444	x	x			
6	Zylinderkopfschraube	Kreuzschlitz M4	L = 20mm	53-420	x	x			
2	Zylinderkopfschraube	Kreuzschlitz M3	L = 60mm	41-360	x	x			
4	Zylinderkopfschraube	Kreuzschlitz M2,5	L = 5mm	53-2505	x	x			
2	Mutter	M3		40-031	x	x			
4	Sicherungs-Mutter	M3		40-034	x	x			
2	Abstandsbolzen	M3 II	L = 5mm	42-650	x	x			
2	Solarmotor	Groß, mit Achse 2mm	D = 32mm	54-109	x	x			

Stückliste - Antrieb mit Solar-Getriebemotor 60 Upm					RF-1	RF-2	RF-G1	RF-G2	SE
Anzahl	Bezeichnung	Material / Spezifikation	Maße (in mm)	Best.Nr.					
2	Getriebemotor	60 upm		54-100			x	x	
4	Winkel	Construction 1 Langloch X 2 Loch		00-01253			x	x	
4	Zylinderkopfschraube	Kreuzschlitz M3	L = 6mm	53-306			x	x	
4	Zylinderkopfschraube	Kreuzschlitz M4	L = 8mm	53-408			x	x	
4	Mutter	M4		40-041			x	x	
4	Mutter	M3		40-031			x	x	

Stückliste - Stromversorgung für 9 Volt-Blockbatterie (ohne Batterie oder Akku)					RF-1	RF-2	RF-G1	RF-G2	SE
Anzahl	Bezeichnung	Material / Spezifikation	Maße (in mm)	Best.Nr.					
1	Flachstab	Construction	4-Loch	00-01003	x		x		
2	Zylinderkopfschraube	Kreuzschlitz M4	L = 25mm	53-425	x		x		
2	Mutter	M4		40-041	x		x		
1	Batterieclip	Für 9 Volt-Block	I-Form	36-441	x		x		

Stückliste - Stromversorgung für 6 AA Batterien (ohne Batterien oder Akkus)					RF-1	RF-2	RF-G1	RF-G2	SE
Anzahl	Bezeichnung	Material / Spezifikation	Maße (in mm)	Best.Nr.					
2	Batteriehalter	3 X AA mit Clip		36-453		x		x	
4	Senkkopfschraube	Kreuzschlitz M3	L = 8mm	53-308S		x		x	
4	Mutter	M3		40-031		x		x	
2	Batterieclip	Für Clipanschlüsse	I-Farm	36-441		x		x	

Stückliste - Solarergänzung					RF-1	RF-2	RF-G1	RF-G2	SE
Anzahl	Bezeichnung	Material / Spezifikation	Maße (in mm)	Best.Nr.					
2	Solarmodul	5 Volt, 30 mA	66mm X 37mm	91-196					x
2	Flachstab	Construction	5 Loch	00-01004					x
2	Zylinderkopfschraube	Kreuzschlitz M4	L = 8mm	53-408					x
2	Mutter	M4		40-041					x
1	LED	3mm, ultrahell grün		91-5253					x
1	LED-Clip	Für 3mm LED	DA = 4,4mm	37-850					x
1	Litze	PVC, isoliert, 0,14qmm, rot	20cm	46-000					x
1	Litze	PVC, isoliert, 0,14qmm,schwarz	20cm	46-001					x

### 3. Trägerplatine

Die Trägerplatine (Shield) kann auch separat mit der Best.Nr.: 10-654-PL oder mit der Elektronik 10-654-PLE (siehe Stückliste Grundbauteile-Elektronik) bestellt werden.

Hier können die Bauteile für eigene Fahrzeuge direkt wie in dieser Anleitung eingebaut werden oder die Platine wird für die Verbraucher oder Sensoren, die an einem anderen Ort untergebracht werden sollen, mit Stiftleisten versehen. Bei den LEDs und dem Schiebeschalter müssen die Bohrungen zunächst auf 1,0mm erweitert werden.



Die Platine kann so in unterschiedlichsten Projekten eingesetzt werden. Lassen Sie Ihrer Phantasie freien Lauf und stöbern Sie in unserem reichhaltigen Materialangebot auf: [www.ellmitron.de](http://www.ellmitron.de)

- Aufrecht stehender Roboter
- Kran - Ein Motor für das Drehen ein Motor für den Lastenhub. Der Ausgang einer LED kann für einen Elektromagnet (mit Transistor) verwendet werden.
- Flipper - die Analogeingänge können als Sensoreingänge verwendet werden) Die beiden Leitungen SCL und SDA sind noch frei und können für ein Display zum Punkte zählen verwendet werden.)
- uvm...

Stückliste - Grundbauteile - Elektronik					PL	PLE
Anzahl	Bezeichnung	Material / Spezifikation	Maße (in mm)	Best.Nr.		
1	Trägerplatine	Für Arduino	100mm X 80mm	10-654 PL	x	x
1	Schiebeschalter	Print-Schiebeschalter 1 X UM	RM 2,5	47-209		x
1	LDR	Fotowiderstand A-9013	D = 5mm	98-621		x
2	Stiftreihe	40pol.	RM 2,5	43-202		x
2	Widerstand	Kohleschicht 1/3 Watt	0 Ohm	70-800		x
2	Widerstand	Kohleschicht 1/3 Watt	100 Ohm	70-210		x
2	Widerstand	Kohleschicht 1/3 Watt	150 Ohm	70-215		x
2	Widerstand	Kohleschicht 1/3 Watt	270 Ohm	70-410		x
1	Widerstand	Kohleschicht 1/3 Watt	330 Ohm	70-233		x
1	Widerstand	Kohleschicht 1/3 Watt	10 KOhm	70-410		x
1	IC	Treiber IC L-293D für 2 Motoren	16pol.	96-013		x
2	LED	Leuchtdiode Superhell, Low Current	rot	91-5165		x
3	LED	Leuchtdiode ultrahell	weiß	91-5461		x
4	LED	Leuchtdiode Superhell, Low Current	gelb	91-5185		x
1	Litze	PVC, isoliert, 0,14qmm, rot	50cm	46-000		x
1	Litze	PVC, isoliert, 0,14qmm,schwarz	50cm	46-001		x