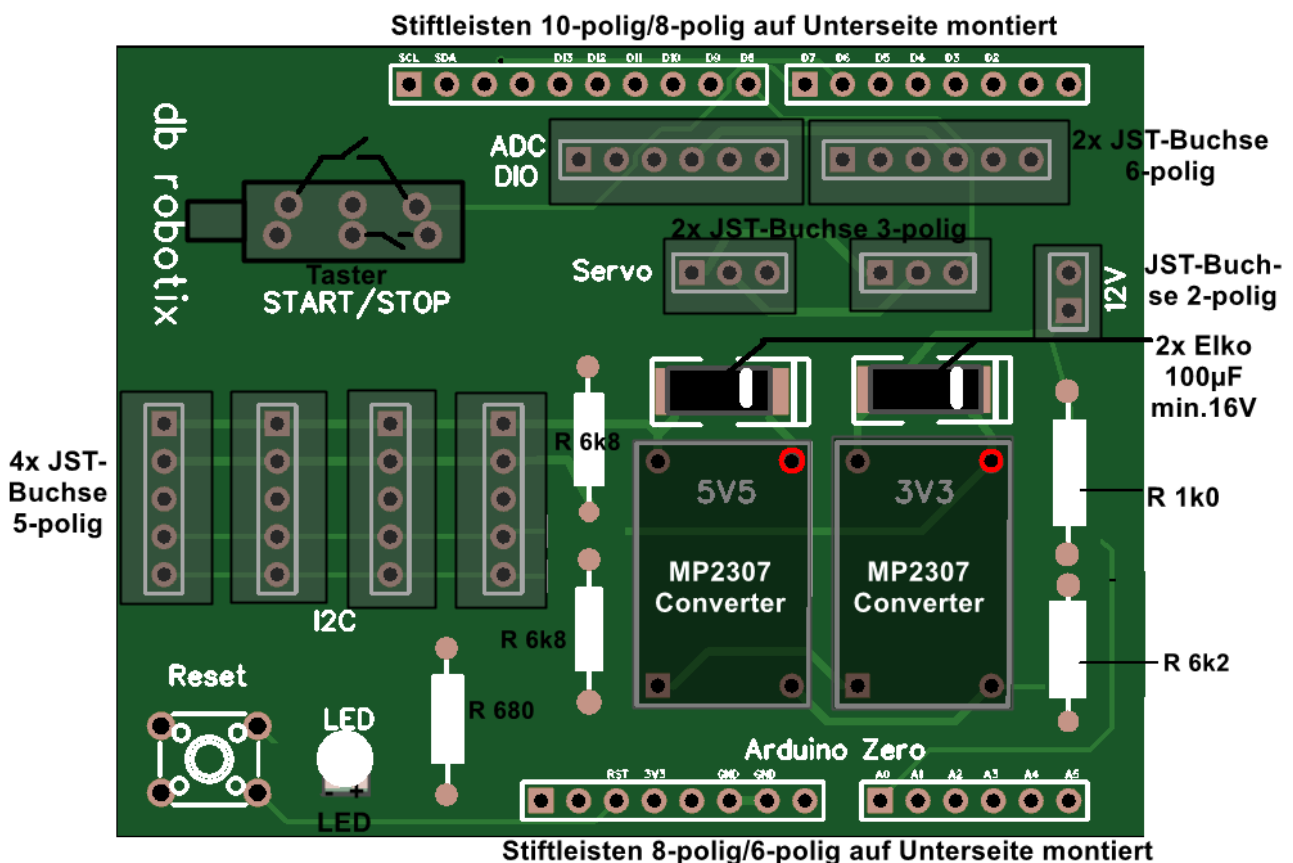


Herstellung des Arduino Shield

Stückliste:

- 1x Leiterkarte Arduino Shield
- 2x MP2307 (360) Step-Down Converter 17mm x 11mm
- 2x Tantal-Elko SMD 100µF/16 V (Gehäusetyp D oder X)
- 2x Widerstand 0.25W 6.8 kΩ
- 1x Widerstand 0.25W 680 Ω
- 1x Widerstand 0.25W 1.0 kΩ ±1%
- 1x Widerstand 0.25W 6.2 kΩ ±1%
- 1x LED (gelb/orange) 3 mm Ø
- 1x Taster/Mikroschalter RM 5.08, z.B. DAOKAI
- 2x JST-Buchse 6-polig
- 4x JST-Buchse 5-polig
- 2x JST-Buchse 3-polig oder Stiftleiste 3-polig RM 2.54
- 1x JST-Buchse 2-polig
- 4x Stiftleiste RM 2.54 10-/8-/8-/6-polig
- versilberter Kupferdraht 0.6 mm

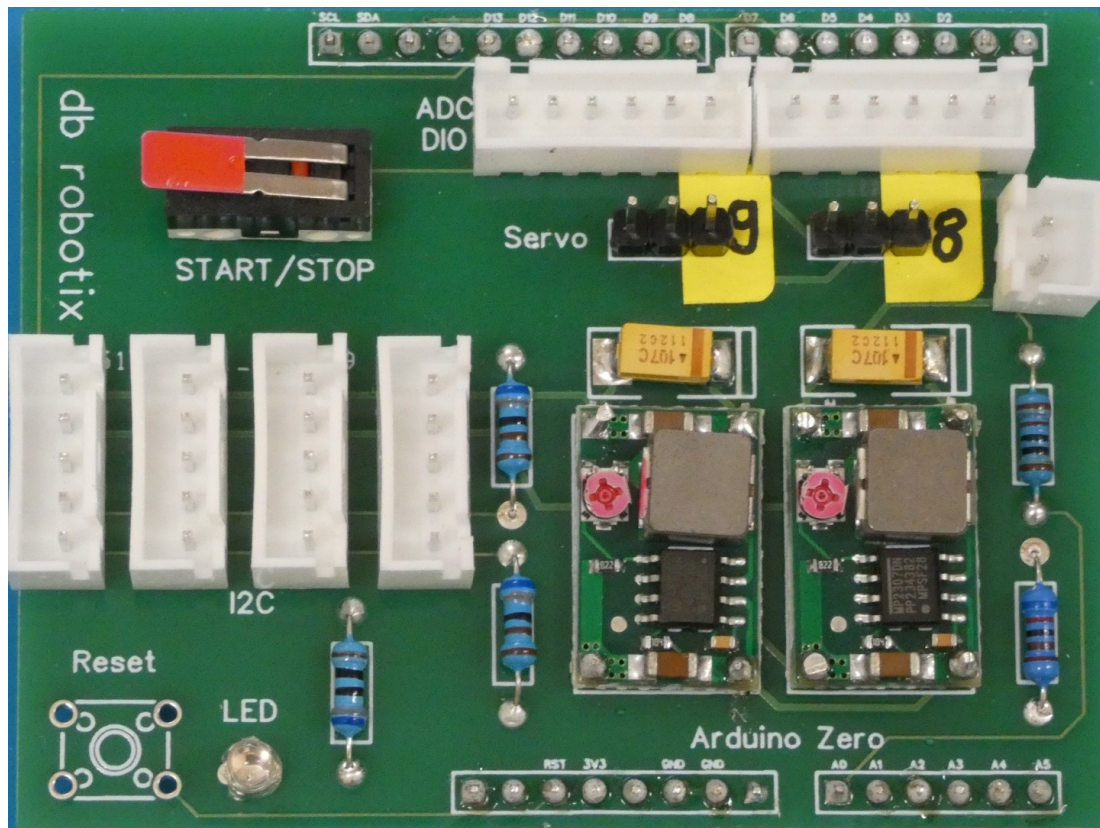
Bestückungsplan:



- Bei der Bestückung der Elkos auf die Polarität achten (Markierung auf Gehäuse = +)
- Bei der Bestückung der LED auf die Polarität achten (langer Draht = +)
- Die beiden Converter mittels jeweils 4 kurzer Silberdrähte verlöten.
- Der Reset-Taster (links unten) bleibt unbestückt, falls die Regeln einen weiteren Taster verbieten.

Den Start-Stop-Taster (links oben) entsprechend seinen Anschlüssen montieren (gewöhnlich untere Reihe)

Bestückte Leiterkarte:



Der mittlere Anschluss der linken I2C-Buchse ist mit D13 des Arduino verbunden (für den Farbsensor A V2). Die anderen mittleren Anschlüsse der I2C-Buchsen sind standardmäßig nicht kontaktiert.

Kalibrierung:

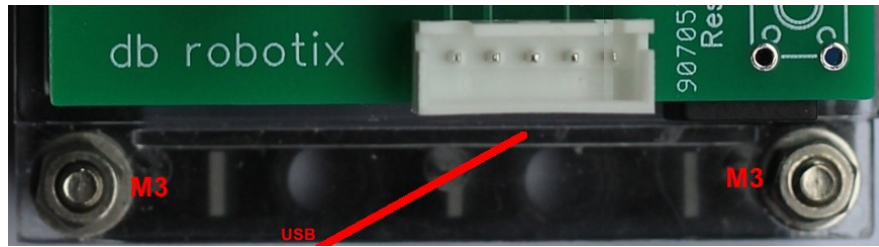
Die beiden Spannungs-Converter sind mit Hilfe der Potentiometer auf die erforderlichen Ausgangsspannungen einzustellen. Das muss unbedingt VOR der ersten Verbindung mit dem Arduino Zero oder anderen Komponenten geschehen, sonst werden diese zerstört!

- Schließe eine Spannung von ca. 12 V an die 2-polige Buchse an, z.B. von einem Labor-Netzteil oder vom Akku-Schutz-Modul.
- Miss mit einem Voltmeter die Ausgangsspannung des 3.3V Converters am Elko oder an der unteren Stiftleiste (3V3 und GND) oder an einer I2C-Buchse (zwischen den beiden oberen Stiften). Stelle das Potentiometer so ein, dass eine Spannung von 3.25 bis 3.35 Volt entsteht.
- Miss mit einem Voltmeter die Ausgangsspannung des 5.5V Converters am Elko oder an einer Servo-Buchse (zwischen den beiden linken Stiften). Stelle das Potentiometer so ein, dass eine Spannung von 5.4 bis 5.6 Volt entsteht.
- Sichere am besten die Poti-Einstellungen mit einem Lacktropfen (danach nicht mehr verstellbar).

Vorsichtshalber sollte noch die Spannung am Arduino-Pin A0 (rechts unten, linke Lötstelle) überprüft werden. Sie muss etwa 1.5V bis 1.7V betragen

Verbindung mit dem Arduino Zero:

Das Shield kann direkt auf ein Arduino Zero Board gesteckt werden. Es empfiehlt sich, die mitgelieferte transparente Trägerplatte des Zero zu verwenden und an der Seite der USB-Buchsen mit einem Lego-Rahmen 88 mm x 56 mm zu verschrauben (2x M3):



Wichtiger Hinweis zum Programmieren

Beim Verbinden des Arduino Zero mit dem PC über die USB-Schnittstelle ist unbedingt darauf zu achten, dass **vor** dem USB-Anschluss die externe 12V-Versorgung (gewöhnlich Akku + Akku-Schutz) eingeschaltet ist!

Ansonsten fließt vom USB über die Arduino-Platine ein hoher Strom in den 3,3V-Spannungswand-
lerausgang des Shields, so dass dieser heiß wird und dadurch geschädigt oder sogar zerstört werden kann. In der Folge kann dann auch der Arduino Zero beim späteren Anschluss des Akkus zerstört werden.

Also folgende Reihenfolge einhalten:

- 1. Akku anschließen und einschalten**
- 2. Computer und Arduino über USB-Kabel verbinden**
- 3. Programm an den Arduino vollständig übertragen**
- 4. USB-Verbindung wieder lösen**
- 5. Akku ausschalten oder Roboter direkt testen**

Beim späteren Betrieb des Roboters ist die USB-Verbindung nicht erforderlich, kann aber (bei eingeschaltetem Akku!) z.B. für Debugging-Zwecke verwendet werden.

Technische Daten:

Controller-Baustein	Arduino Zero
Versorgungsspannung	8 – 16 V
Stromaufnahme	Leerlauf: 32 mA @ 12 V
Spannungswandler	Verlustarmer Step-Down-Wandler MP1584, eingestellte Ausgangsspannungen: 3.3V ($\pm 0.05V$), max. 1.5 A und 5.5V ($\pm 0.2V$), max. 1.5 A Ruhestrom @ 12 V: 2×9 mA
Interne Kapazität an 3.3 V	100 μF
I2C-Bus	Master-Funktion Pullup-Widerstände zu SDA, SCL: 4 k Ω Clock-Frequenz: 100 kHz 4 x Onboard-Anschlüsse JST-Buchse 5-polig (GND, 3.3V, DIO, SDA, SCL) (DIO nur optional verbunden)
Universal-ADC/DIO-Anschlüsse	2 x JST-Buchse 6-polig (2 x ADC, GND, 3.3V, 2 x DIO) 1. Anschluss: Ports A3, A4, D3, D2 2. Anschluss: Ports A1, A2, D5, D4 (A1-A4 alternativ D15-D18 per Software)
ADC / Referenzspannung	10 bit / 2.23 V
Servomotor-Anschlüsse	2 x Stiftleiste oder JST-Buchse 3-polig (GND, 5V, DIO) eingestellte Versorgungsspg.: 5.5V ($\pm 0.2V$), max. 1.5 A Ports: D8 / D9
Unterstützende Bibliotheken	anadigMaster: Battery, Button, Led, ServoMotor
Zusatz-Komponenten	LED orange (Port D6, 2 mA): Warten nach Booten Taster rot (Port D7): Start/Stop Reset-Taster optional Spannungsteiler 7.2:1 für 12V-Messung mit ADC A0
Bluetooth/WiFi	nicht verfügbar
Größe	80 mm x 58 mm
Höhe	32 mm
Masse	60 g
Befestigungslöcher	2 x 3 mm \varnothing , Abstand 48 mm (6 Lego-Löcher)

Belegung der Arduino-Zero-Anschlüsse:

DIO = digital input/output ; nc = not connected

obere linke Reihe:

SCL	SDA	Aref	GND	D13	D12	D11	D10	D9	D8
I2C	I2C	nc	GND	I2C-Buchse (optional)	(optional zu verdrahten)	(optional zu verdrahten)	(optional zu verdrahten)	Servo links	Servo rechts

obere rechte Reihe:

D7	D6	D5	D4	D3	D2	TX	RX
Taster	LED	DIO links	DIO links	DIO rechts	DIO rechts	nc	nc

untere linke Reihe:

ATN	IOref	Reset	3V3	5V	GND	GND	Vin
nc	nc	R-Taster (optional)	3V3	nc	GND	GND	nc


untere rechte Reihe:

A0	A1	A2	A3	A4	A5
12V-Messung	ADC links	ADC links	ADC rechts	ADC rechts	nc


Belegung der JST-Buchsen:

(Blick auf Buchse)

ADC-DIO-Universal (6-polig):


Kerbe						
Stift	•	•	•	•	•	•
#	1	2	3	4	5	6
Funktion	ADC (oder DIO)	ADC (oder DIO)	GND	3V3	DIO	DIO
Port 1 (rechts)	A3 (D17)	A4 (D18)	GND	3V3	D3	D2
Port 2 (links)	A1 (D15)	A2 (D16)	GND	3V3	D5	D4

I2C-Bus (5-polig):


Kerbe					
Stift	•	•	•	•	•
#	1	2	3	4	5
Funktion	GND	3V3	(DIO)	SDA	SCL

optional

Servo (3-polig):

Kerbe			
Stift	•	•	•
#	1	2	3
Funktion	GND	5V5	D8 / D9
Kabel	br	rt	or

12 V (2-polig):

	
•	•
1	2
-	+
sw	rt