

DCC-EX v 5.0.9

03c - Erweiterungen - Servo



Tests wurden unter Windows 11 durchgeführt

Die folgenden Einstellungen und Vorgehensweise kann natürlich jeder selbst bestimmen.
Es sind nur meine Erfahrungen.

Wichtig	2
Servo's - PCA9685	3
Händische Einrichtung	4
Servo Tests ohne RailControl	4
Servo vorbereiten z.B. für RailControl	6
Einrichtung in der Datei „myAutomation.h“	8

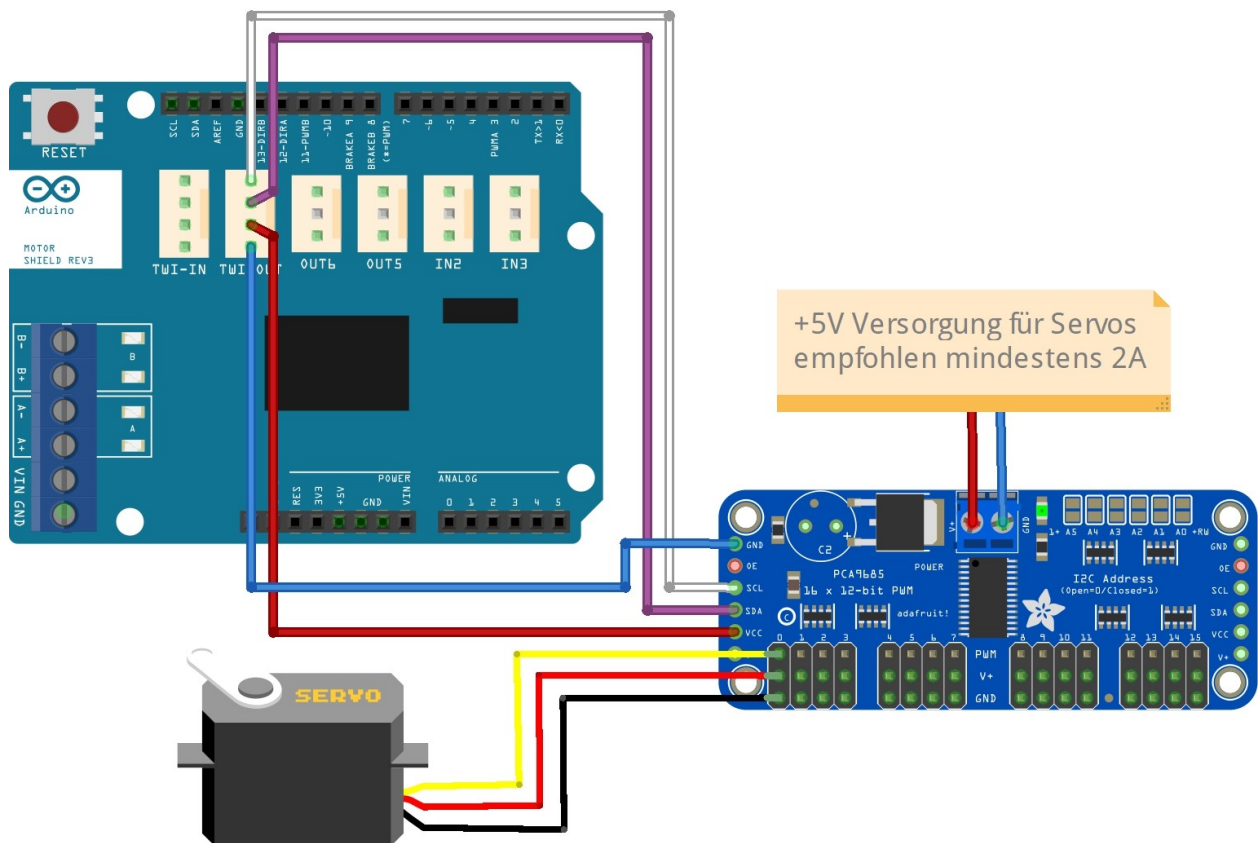
Wichtig

- **Ich übernehme keine Garantie/ Haftung auf Richtigkeit, Vollständigkeit usw. Es beruht alles auf eigener Erfahrung.**
- **Bei den nachfolgenden Schaltungs-/ Anschlussbeispielen ist unbedingt selbst auf die Pinbelegung zu achten, es gibt Bauelemente mit gleichen Daten aber anderer Pinbelegung.**
- **Auch auf die Spannungsversorgung muss unbedingt selbst geachtet werden, manche Komponenten benötigen 3,3 Volt, andere aber 5 Volt usw.**
- **Bei der Versorgungsspannung für den Arduino über USB unbedingt darauf achten, daß es nicht zu einer Überlastung des speisenden Raspberry Pi kommt.**

Ich habe daher, nachdem ich die Hardwareinstallation (siehe DCC-EX Beschreibung) vorgenommen habe, zusätzlich zum USB-Anschluss des Arduinos noch ein Netzteil für den Arduino und ein weiteres natürlich für die Gleisspannung vorgesehen.



Servo's - PCA9685



fritzing

Eigene Anschlussbelegung beachten!

Das Servomodul selbst wird vom Arduino gespeist. Da die Servos jedoch Motoren enthalten, können diese zumindest in der Summe nicht über den Arduino gespeist werden. Dadurch wird eine eigene 5V-Spannungsversorgung unbedingt erforderlich.

Die Verbindungen vom Servomodul zum Arduino werden mit vier Überbrückungsdrähten (+5 V und GND sowie SCL und SDA) hergestellt.

In der EX-CommandStation sind die Treiber für das PCA9685 Modul bereits installiert und für die 16 Servos stehen die **PIN-Nummern 100 - 115** zur Verfügung.

In der obigen Abbildung ist ein Servo dargestellt, das mit dem ersten Pin-Satz (0V/5V/Signal) am Modul verbunden ist. Dieser Servo wird somit über die PIN-Nummer 100 adressiert.

Händische Einrichtung

Die manuelle Installation habe ich getestet, um die Befehle von DCC-EX besser zu verstehen. Es ist viel Arbeit und man darf keinen Schritt vergessen, sonst war alles umsonst.

Es funktioniert aber alles einwandfrei.

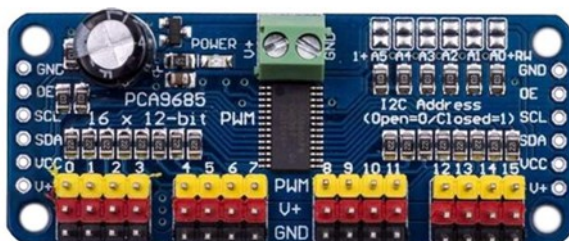
Der bessere Weg ist jedoch, die Einstellungen in der Datei „myAutomation.h“ vorzunehmen.

Hier werden dann gemäß des Inhaltes der Konfigurationsdateien bei jedem Neustart alle Einstellungen automatisch vorgenommen und man erspart es sich - auch bei einem Wechsel des Arduinos - wieder alles händisch neu einrichten zu müssen.

Einfach nur das Programm in den neuen Arduino Mega einspielen - fertig.

Servo Tests ohne RailControl

1. Servomodul Adr. 0x40		
Servo	Adresse	Vpin
01	100	100
02	101	101
03	102	102
04	103	103
05	104	104
06	105	105
07	106	106
08	107	107
09	108	108
10	109	109
11	110	110
12	111	111
13	112	112
14	113	113
15	114	114
16	115	115

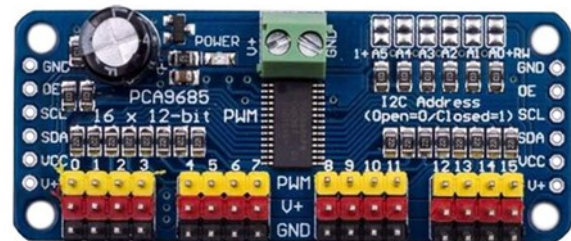


PCA9685 Servo Modul



1. Servomodul Adresse 0x40

2. Servomodul Adr. 0x41		
Servo	Adresse	Vpin
17	116	116
18	117	117
19	118	118
20	119	119
21	120	120
22	121	121
23	122	122
24	123	123
25	124	124
26	125	125
27	126	126
28	127	127
29	128	128
30	129	129
31	130	130
32	131	131



PCA9685 Servo Modul



2. Servomodul Adresse 0x41

Befehl: <D SERVO Vpin pos>

- Vpin = 100 - 115 für I2C Adresse 0x40
- Vpin = 116 - 132 für I2C Adresse 0x41

Die Adresse 100, 101, usw. sind nicht die DCC Adressen für z.B. RailControl Weichensteuerung.

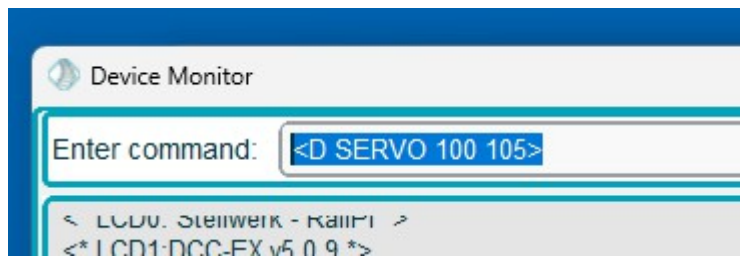
Bei den folgenden Tests bitte nicht vergessen DCC-EX mit dem Befehl

- <1> einschalten
und mit
- <0> ausschalten.

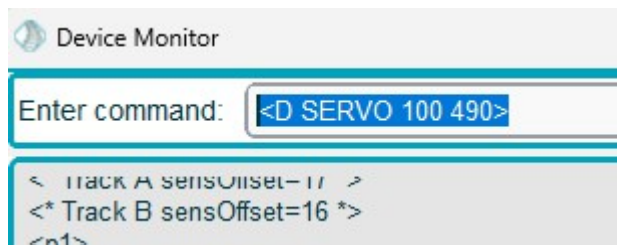
Die Tests können mit einem Terminalprogramm oder auch mit dem „Device Monitor“ vom Programm „EX-Installer“ durch geführt werden.

Railcontrol wird hierfür noch nicht benötigt.

- Befehl: <D SERVO 100 105> Servo dreht in die eine Richtung



- Befehl: <D SERVO 100 490> Servo dreht in die jetzt in die andere Richtung

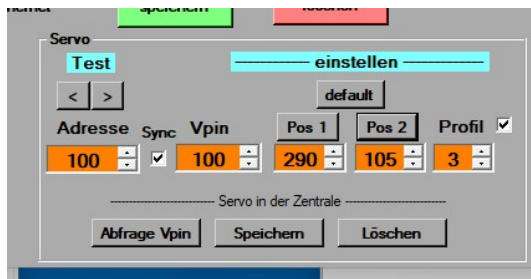


Hinter der Angabe der Position kann noch ein weiterer Parameter angegeben werden.

- Befehl: <D SERVO 100 105 0> Servo dreht sofort
 - 0 = sofort
 - 1 = schnell
 - 2 = mittel
 - 3 = langsam
 - 4 = wippen

Bevor man die Servos einbaut, sollten die richtigen Positionen ermittelt werden, damit nach dem Einbau nichts kaputt geht.

Die Einstellungen kann man wunderbar mit dem Programm „PinSetting“ von Wilfried (siehe unter „01 - Allgemeines und Wichtiges.pdf“) testen.

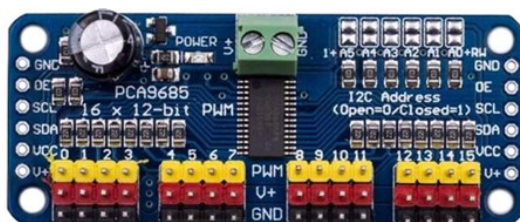


Servo vorbereiten z.B. für RailControl

- Befehl: <T id SERVO Vpin pos1 pos2 profile>
 - id = 0 - 32767 (DCC - Adresse)
 - Vpin = 100 - 115 für I2C Adresse 0x40
 - pos1 = 105 - 490
 - pos2 = 490 - 105
 - Profile = 0 - 4 (0 = sofort, 1 = schnell, 2 = mittel, 3 = langsam, 4 = wippen)

In der Regel benutzt man als id die Nummer des Vpins, damit man es sich besser merken kann.

1. Servomodul Adr. 0x40		
Servo	Adresse	Vpin
01	100	100
02	101	101
03	102	102
04	103	103
05	104	104
06	105	105
07	106	106
08	107	107
09	108	108
10	109	109
11	110	110
12	111	111
13	112	112
14	113	113
15	114	114
16	115	115

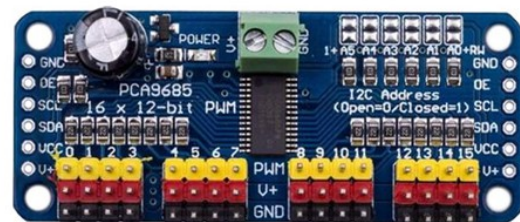


PCA9685 Servo Modul



1. Servomodul Adresse 0x40

2. Servomodul Adr. 0x41		
Servo	Adresse	Vpin
17	116	116
18	117	117
19	118	118
20	119	119
21	120	120
22	121	121
23	122	122
24	123	123
25	124	124
26	125	125
27	126	126
28	127	127
29	128	128
30	129	129
31	130	130
32	131	131



PCA9685 Servo Modul



2. Servomodul Adresse 0x41

- **Befehl: <T 100 SERVO 100 105 490 3>** mit der Ausführung dieses Befehls werden die ermittelten Positionen im Arduino Mega temporär hinterlegt. Hier für die DCC Adresse 100 an Vpin 100 (die Adresse kann auch zwischen 0 - 32767 liegen) die Positionen 105 und 490.

```
<* LCD4:Free RAM= 4906b *>
<T 100 SERVO 100 105 490 3>
<O>
<* LCD4:Free RAM= 4892b *>
```

Eine Rückmeldung vom Arduino Mega <O> gibt an, daß der Befehl angenommen wurde. Bei einer Rückmeldung von <X> würde es einen Fehler bedeuten.

Jetzt müssen noch die Parameter des temporären Befehls gespeichert werden:

- **Befehl: <E>** speichert Befehl in das EEPROM des Arduino Mega
- **Befehl: <e>** löscht das EEPROM
(aber Achtung, das gesamte EEPROM wird damit gelöscht - siehe auch Beschreibung auf Seite 8 in Kapitel „02a - Erweiterungen MEGA 2560.pdf“ und vor allem in Kapitel „03 - Das EEPROM im MEGA 2560.pdf“)

Nun kann man in RailControl die DCC Adresse in den Weichenparametern hinterlegen.

W100

Basisdaten	Position
Name:	<input type="text" value="W100"/>
Typ:	<input type="text" value="rechts"/>
Adresse:	<input type="text" value="100"/> <input type="button" value="-"/> <input type="button" value="+"/>
Schaltzeit (ms):	<input type="text" value="250"/>
Invertiert:	<input type="checkbox"/>

Einrichtung in der Datei „myAutomation.h“

Gemäß der [EX-Rail-Referenz](#) werden Weichen hier mit der folgenden Syntax definiert:

SERVO_TURNOUT(id, pin, active_angle, inactive_angle, profile [, "description"])

- **id** = eindeutige ID innerhalb der DCC-EX-CommandStation.
Die Werte können im Bereich von 0 - 32767 liegen (DCC - Adresse)
- **pin** = die ID des Pins, mit dem das Servo verbunden ist, was normalerweise die VPin-ID der PCA9685 Controller-Platine ist.
- **active_angle** = Der Winkelwert, um den sich das Servo bewegt, wenn die Weiche gestellt wird.
- **inactive_angle** = Der Winkelwert, um den sich das Servo bewegt, wenn die Weiche geschlossen ist.
- **profile** = Bewegungsprofil der Weiche:
 - **Instant** = sofort
 - **Fast** = schnell
 - **Medium** = mittel
 - **Slow** = langsam
 - **Bounce** = **wippen** (in der Regel beim Formsignal genutzt)
- **description** = Eine benutzerfreundliche Beschreibung der Weiche, die in den Wi-Throttle-Apps und Engine Driver angezeigt wird. Diese Beschreibung muß in Anführungszeichen "" gesetzt werden.

Auszug aus meiner Datei „myAutomation.h“ für 2 Servos mit den Adressen 100 und 101:

```
// Servo Einstellung
SERVO_TURNOUT(100, 100, 350, 250, Slow)
SERVO_TURNOUT(101, 101, 350, 210, Slow)
```

Der Übersichtlichkeit halber ist bei mir immer die „id“ und „pin“ identisch.
Die Werte für „active_angle“ **und** „inactive_angle“ sind die, die man z.B. in der händischen Einrichtung (s.o.) ermittelt hat.

Jetzt kann man - wie auf der vorigen Seite bereits gesehen - in RailControl die DCC Adresse in den Weichenparametern hinterlegen.

W100

Basisdaten

Position

Name:	<input type="text" value="W100"/>
Typ:	<input type="text" value="rechts"/>
Adresse:	<input type="text" value="100"/> <input type="button" value="-"/> <input type="button" value="+"/>
Schaltzeit (ms):	<input type="text" value="250"/>
Invertiert:	<input type="checkbox"/>