

Die Multifunktionale DCC-Decoder für Servos und Zubehör mit DCCNext für alle



Autor: Nico Teering
dezember 2021
Mardec version: 7.0
Document version: 7.0D (Übersetzung Carl Jaeger)
Info: info@Arcomora.com oder www.Arcomora.com/Contact
www.Arcomora.com/Mardec/

Einführung

Damit Sie selbst für Ihre Anlage einfache und kostengünstige Elektronik Komponenten bauen können, habe ich ein Programm namens MARDEC für einen multifunktionalen DCC-Decoder auf Basis eines Arduino-Prozessors entwickelt. Mit MARDEC können Servos für Weichen- oder Torantriebe, aber auch anderes Zubehör gesteuert werden. Das MARDEC-Programm ermöglicht es Ihnen, Arduino-Prozessoren zu programmieren, ohne die eigentliche Programmiersprache zu beherrschen.

Wie jeder andere DCC-Decoder muss auch ein MARDEC Decoder konfiguriert werden.

Diese Konfiguration wird in der Regel fälschlich als Programmierung bezeichnet.

Um Verwirrung zu vermeiden, verwenden wir den Begriff **Programmierung** um das eigentliche Arduino-Programm zu schreiben, dies betrifft Sie als Anwender jedoch nicht.

Der Anwender arbeitet bei der Einstellung mit der **Konfiguration**. Bei der Konfiguration werden die Servo's und das Zubehör eingestellt. Mit Zubehör sind z.B. Schaltaufgaben gemeint.

Das bedeutet, dass MARDEC in zwei verschiedenen "Zuständen" arbeitet.

Im **Konfigurationsmodus** können Servos und Schaltaufgaben mit Befehlen, die nur aus einem einzigen Buchstaben bestehen, über die Tastatur des PC's konfiguriert werden. Dabei kommuniziert der PC über ein USB-Kabel mit dem Arduino, auf dem sich MARDEC befindet. Dazu wird kein DCC-Signal aus der Modellbahn-Zentrale benötigt.

Durch den Befehl 'E' wechselt MARDEC in den **Betriebsmodus**.

Im Betriebsmodus werden die Servos und das Zubehör mit dem DCC-Signal Ihrer Modellbahn-Zentrale gesteuert. Das USB-Kabel wird dabei **nicht** mehr benötigt.

Die Installation der gesamten Software ist sehr benutzerfreundlich. Die gesamte benötigte Software wird in einem Arbeitsgang installiert. Das Hochladen des Programms auf den Decoder erfolgt ebenfalls nahezu vollautomatisch. Bitte beachten Sie hierzu die Installationsanleitung.

Der Decoder

Für den Bau des Decoders selbst haben Sie drei Möglichkeiten:

- Sie verwenden den **Arduino UNO** oder **Arduino MEGA** mit dem DCC/Power Shield wie auf www.arcomora.com/mardec, oder
- den **Arduino NANO** mit einer selbstgebauten DCC-Schaltung, oder
- den **DCCNext-Decoder** wie auf www.arcomora.com/dccnext beschrieben.

Die MARDEC Version 5 und 6 sind für den DCCNext konzipiert, kann aber auch auf einem Arduino (Uno oder Mega) mit DCC/Power Shield verwendet werden.

ArCoMoRa

MARDEC ist ein Teil des **ArCoMoRa**-Konzeptes. Dieses steht für **Arduino Controlled Model Railway**.

Hierzu gehört außerdem auch ARSIGDEC, ein DCC-Signaldecoder mit dem Sie sämtliche Signale Ihrer Anlage steuern können, sowie der ARLOCO, ein konfigurierbarer LocoNet-Gleisbesetzmelder.

Weiter Infos hierzu unter www.Arcomora.com

Neu in Version 7

In dieser Version wurde die DCC_library von Mynabay durch die NMRA-Bibliothek ersetzt.

Mit Dank an Karst Drenth (Drenth Design & Consultancy)

Durch diese Änderung wurde das Problem mit Railcom gelöst

Ein Überblick über die MARDEC-Möglichkeiten

- Steuern Sie maximal 12 Servos für Weichen ohne Herzstückpolarisierung oder maximal 8 Servos für Weichen mit Herzstückpolarisierung. Die Polarisierung erfolgt durch ein externes Relais
- Steuerung von Zubehör auf 9 Arten, einschließlich der Steuerung von Weichenspulen und variabler PWM Kontrolle.
- Ein Arduino-Port kann als Eingang konfiguriert werden. Damit kann ein Servo/Zubehör an einem anderen Port ein- oder ausgeschaltet oder auch manuell bedient werden.
- Interaktiv über Bildschirm und Tastatur die Servos und das Zubehör konfigurieren und dies unabhängig von der verwendeten DCC-Zentrale. Die eigentliche Arduino Software (IDE) ist hierfür **nicht** erforderlich
- Der Start- und Endwinkel jedes Servos kann Grad-genau eingestellt werden
- Jedem Servo, Zubehör oder Eingang kann eine beliebige DCC-Adresse (1-2000) zugewiesen werden, also sind aufeinanderfolgende Adressen nicht zwingend notwendig.
- Jedes Servo (max. 8) kann mit einem Relais für die Herzstückpolarisierung gekoppelt werden. Wenn das Servo dreht, wird dieses Relais nach der Hälfte der Drehung umgestellt
- Jedes Servo (max. 5) kann mit zwei Relais für die Herzstückpolarisierung gekoppelt werden. Dies führt zu einer noch genaueren Umschaltung des Herzstücks.
- Für jedes Servo kann eine eigene Drehgeschwindigkeit eingestellt werden, wodurch Servos einfacher für andere Zwecke als die Umstellung von Weichen eingesetzt werden können
- "Bouncing", nachwippen der Servos am Ende der Drehung. Dies kann in 4 verschiedenen Winkeln erfolgen. Hiermit kann z.B. das Nachwippen von Schrankenbäumen oder Flügelsignalen simuliert werden
- Eine Testoption. Hiermit wird die Änderung des DCC-Signals von 0 > 1 oder umgekehrt simuliert werden
- Umkehr der Zustände für Servos und Zubehör. Hiermit kann eingestellt werden, ob ein Servo bei der Schalterstellung 1 bzw. 0 an der Zentrale den "kleinen" oder "großen" Winkel anfährt. Dies kann z.B. erforderlich werden, wenn das Servo auf verschiedene Arten montiert wird. das Gleiche gilt für Zubehör
- Eine Dokumentationsoption, die alle Einstellungen anzeigt
- Zuweisen eines Verwaltungscodes zu jedem Servo, Zubehör oder Eingang
- Eine Reset-Option, bei der alle Einstellungen aus dem Speicher des MARDEC gelöscht werden.
- Eine Reset-Option zum Löschen der Einstellungen eines **einzelnen** Ports.
- Einstellbare Standard- Drehgeschwindigkeit des Servoarms (5-100 ms pro Grad).
- Rückkehr zum Konfigurationsmodus, indem das USB-Kabel angeschlossen und der Befehl C eingegeben wird
- Korrekturmöglichkeit des Adressoffsets bei der Verwendung von Roco-Zentralen (MM, z / Z21)
- Eine Hilfeoption zeigt alle Befehle zur Konfiguration an.
- Eine Protokolloption speichert alle Konfigurationssitzungen
- Die Konfiguration ist unabhängig von der verwendeten Busstruktur (LocoNet, S88 und dergleichen). Die Steuerung erfolgt jedoch nur über das DCC-Protokoll.
- Zubehör-Ports haben folgende Schalt-Optionen:
 1. **Single Steady**: der Port ist abhängig von der Zentrale permanent "hoch" oder "niedrig" (ein- bzw. ausgeschaltet)
 2. **Double Steady**: wie Single Steady, aber ein zweiter Port hat den entgegengesetzten Wert
 3. **Single Flashing**: ein Port geht abwechselnd auf "hoch" (ein) und "niedrig" (aus). Die Ein- bzw. Ausschaltzeit kann separat eingestellt werden
 4. **Double Flashing**: Wie "Single Flashing", aber ein zweiter Port hat den umgekehrten Wert
 5. **Single One Shot**: hierbei geht ein Port für eine kurze, einstellbare Zeit auf "hoch" (ein), wenn die entsprechende Adresse über die Zentrale ausgelöst wird. Diese Funktion kann auch verwendet werden, um am Ende des Impulses ein Servo / Zubehör an einem anderen Port zu schalten
 6. **Double One Shot**: Ein Port geht für eine einstellbare Zeit auf "hoch", wenn die entsprechende Adresse über die Zentrale eingeschaltet wird. Ein zweiter (Buddy-) Port, geht für eine einstellbare Zeit auf "hoch", wenn die gleiche Adresse ausgeschaltet wird. Mit dieser Funktion können z.B Weichen mit Spulenantrieben gestellt werden
 7. **Analog PWM** (Pulse Width Modulation): in diesem Fall wechselt ein Port für eine separat einstellbare Zeit von 0 auf ein einstellbares Maximum (max. 255), wenn die Adresse durch die Zentrale von "niedrig" auf "hoch" wechselt, und nach einer ebenfalls einstellbaren Zeit zurück auf 0, wenn die Zentrale von "hoch" auf "niedrig" wechselt
 8. **Flicker Mode**. Hiermit können Sie bei einer angeschlossenen LED ein Flackern erzeugen. Mit geeigneten LEDs können Sie zum Beispiel Feuer oder das Licht beim elektrischen Schweißen simulieren
 9. **Random**. Hiermit können Sie ein angeschlossenes Zubehör ständig ein- und ausschalten. Die Ein- und Ausschaltzeiten werden zufällig zwischen zwei einstellbaren Grenzwerten erzeugt (20 ms - 600ms). Sie können aber auch eine feste Ein- oder Ausschaltzeit festlegen.

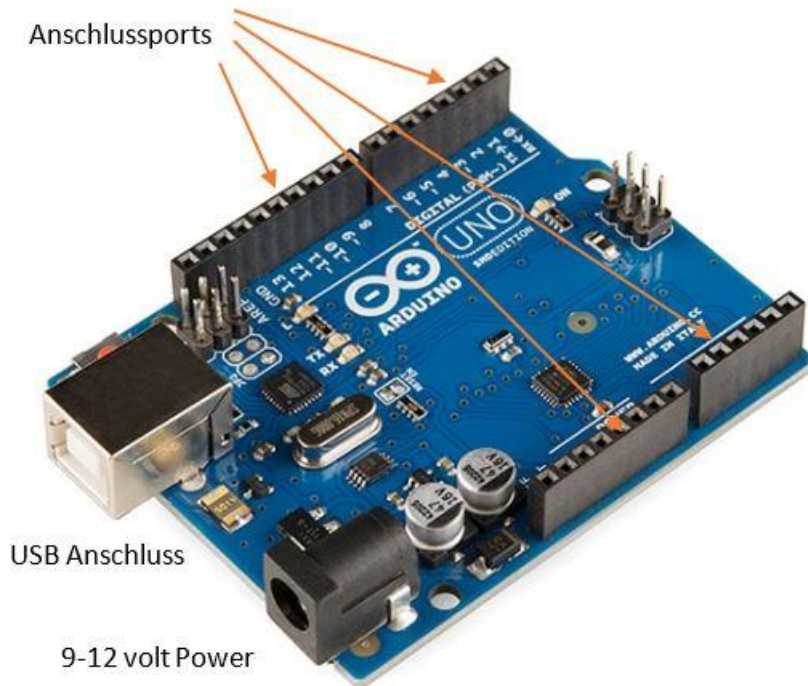
Was braucht man?

Die Hardware

Hier haben Sie eine Reihe von Möglichkeiten.

Option 1: Ein Arduino UNO, Arduino Mega2560 oder Arduino Nano.

Ein Arduino ist ein Mikrocomputer mit vielen Anschlußmöglichkeiten. Diese werden als Ports bezeichnet.



Diese sind auf der Leiterplatte nummeriert.

Für den MARDEC-Decoder verwenden wir maximal 16 dieser Ports. An diese Ports können Servos, Relaismodule (siehe unten) und LEDs (mit Vorwiderstand) direkt angeschlossen werden.

Ein Arduino-Port hat eine Ausgangsspannung von 0 oder 5 Volt und kann maximal 40 mA verarbeiten. Für Zubehör, das mehr Strom benötigt (z.B. Motoren, Weichenspulen und LED-Streifen), ist eine zusätzliche Verstärkerstufe erforderlich.

Für analoge Anwendungen gibt es 4 Ports, an denen eine Spannung mit variabler Impulsbreite abgegriffen werden kann.

Auf der linken Seite sehen Sie den USB-Anschluss. Der Arduino wird über das USB-Kabel mit Strom versorgt.

Der USB-Anschluss wird benötigt, um das

Programm in den Arduino zu kopieren. Außerdem erfolgt hierüber im Konfigurationsmodus die Kommunikation mit dem PC.

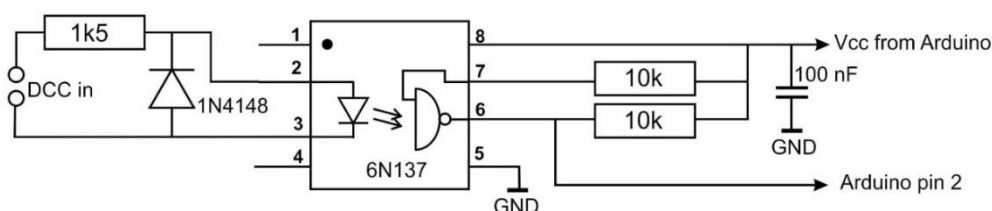
Wenn kein USB-Kabel angeschlossen ist, kann die Stromversorgung über die schwarze Buchse (Power Jack) neben dem USB-Anschluss erfolgen. An diese Buchse kann eine 7-12 Volt Gleichspannungsquelle angeschlossen werden. Der Arduino selbst macht daraus eine stabilisierte 5 Volt Spannung. Beide Anschlüsse können gleichzeitig genutzt werden.

DCC-Schaltung

Um das DCC-Signal für den Arduino geeignet zu machen, wird etwas Elektronik benötigt. Das IC 6N137 ist ein schneller Optokoppler, der das DCC-Signal an den Arduino überträgt.

Die 5 Volt Versorgungsspannung des Optokopplers wird direkt an einen der 5V-Ports des Arduino angeschlossen. Pin 6 des IC's ist mit Port 2 des Arduino verbunden.

Diese zusätzlichen Komponenten werden am besten auf eine Streifenrasterplatte gelötet.



DCC/Power shield

Es ist für die DCC-Schaltung ein Shield entwickelt. Auf diesem Shield befindet sich auch eine Stromversorgung für den Arduino.

Das Board enthält ebenso ein 5V-Netzteil (max. 1 Ampere), das unter anderem für Relais, LEDs und Servos verwendet werden kann.

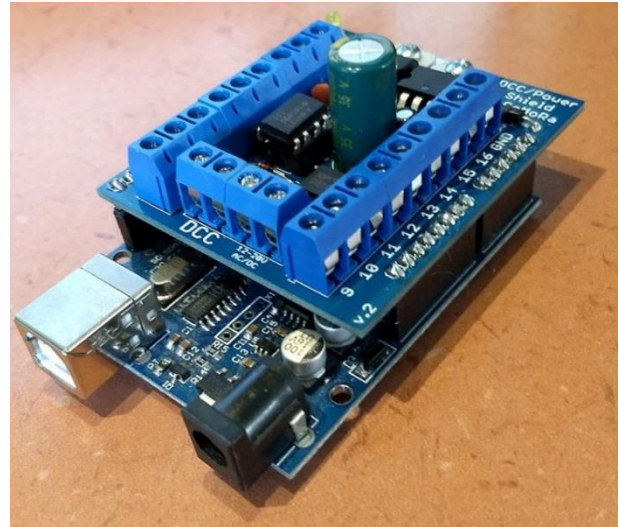
Grundsätzlich wird aber für Servos und Relais eine externe Stromversorgung empfohlen.

Dieses Shield kann mit dem Bestellformular bestellt werden:

www.Arcomora.com/order

Wenn nicht auf Lager, kann die Lieferzeit einige Wochen betragen.

Das Arduino Board (Uno/Mega/Nano) muss von Ihnen selbst gekauft werden.

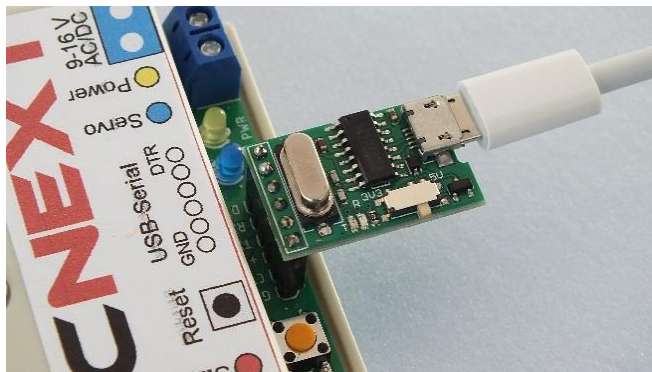


Option 2: Der DCCNext-Decoder

Völlig neu ist der DCCNext-Decoder.

Dieser Decoder vereinigt einen Arduino-Prozessor (ATMEGA328P) mit einem Netzteil und einer DCC-Schaltung und ist somit eine Kombination aus einem Arduino UNO und dem DCC-Shield.

Eine separate USB-Schnittstelle (CH340) stellt die Verbindung zum PC her.



Dieser Decoder kann sowohl mit Schraubklemmen als auch mit Dupont-Steckern ausgestattet werden. Ein Servo kann somit direkt an die Pins angeschlossen werden.

Zu diesem Zweck ist auch eine separate 5V-Servoversorgung vorgesehen.

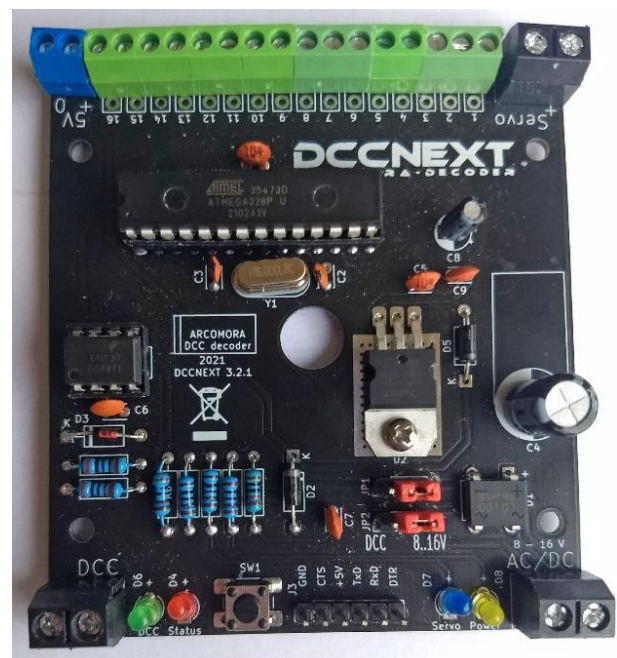


Das Vorhandensein eines DCC-Signals wird durch eine LED sichtbar gemacht,

Weitere Informationen: www.Arcomora.com/DCCNext

Dieses DCCNext kann hier bestellt werden:

<http://www.arcomora.com/order>



Wichtig:

Mardec 4 kann auch mit dem DCCNext verwendet werden.

Mardec 5, 6 und 7 kann auch mit dem DCC-Shield verwendet werden.

In beiden Fällen gilt die untenstehende Umrechnungstabelle:

Mardec 4 Pin	Mardec 5/6/7 Port	Mardec 4 Pin(Mega)	Mardec 5/6/7 Port
3	1	11	9
4	2	12	10
5	3	14(54)	11
6	4	15(55)	12
7	5	16(56)	13
8	6	17(57)	14
9	7	18(58)	15
10	8	19(59)	16

MARDEC 5,6 und 7 verwenden eine vollständig geänderte Datenstruktur zur Speicherung der Konfiguration.

MARDEC 4 kann daher NICHT auf die Version 7 aktualisiert werden.

Wenn Sie MARDEC 4 durch MARDEC 7 auf einem Arduino ersetzen, müssen Sie alles neu konfigurieren.

Machen Sie also zunächst z.B einen Screenshot der aktuellen Konfiguration, bevor Sie Version 6 über Version 4 installieren, hieraus können Sie dann die Werte entnehmen und in MARDEC5 hinein schreiben.

Relais

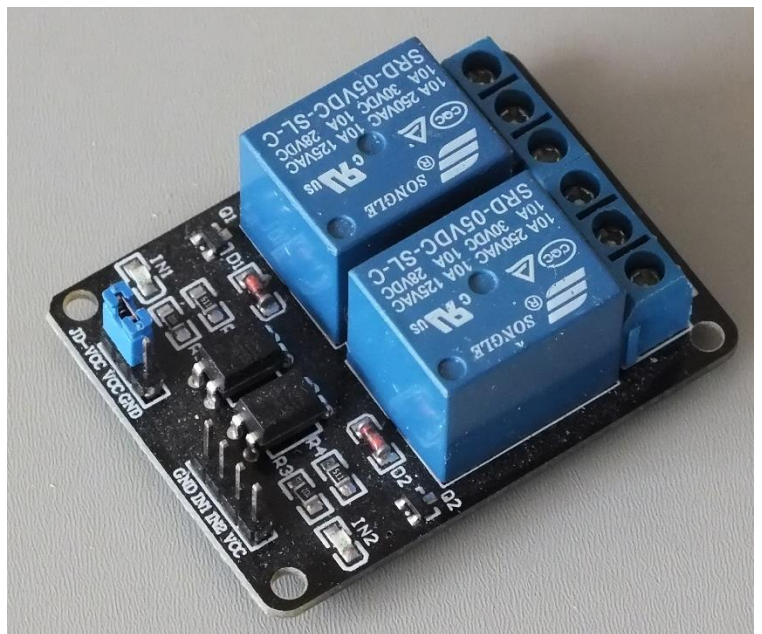
Prinzipiell kann jedes 5V Relais für die Herzstück-Polarisierung oder eine andere Anwendung verwendet werden.

Zu diesem Zweck kann ein Arduino Port ein Relais über einen Transistor schalten.

Die fertigen Relaismodule sind allerdings kostengünstiger und einfacher zu bedienen. Diese können direkt von einem Arduino Port gesteuert werden.

Sie enthalten zwei voneinander unabhängige Relais mit einer Anzeige-LED und einem Optokoppler und sind auch mit 4 oder 8 Relais erhältlich.

Die Spannungsversorgung kann zwar direkt aus dem Arduino erfolgen, es wird aber dringend empfohlen, ein externes 5V-Netzteil zu verwenden.



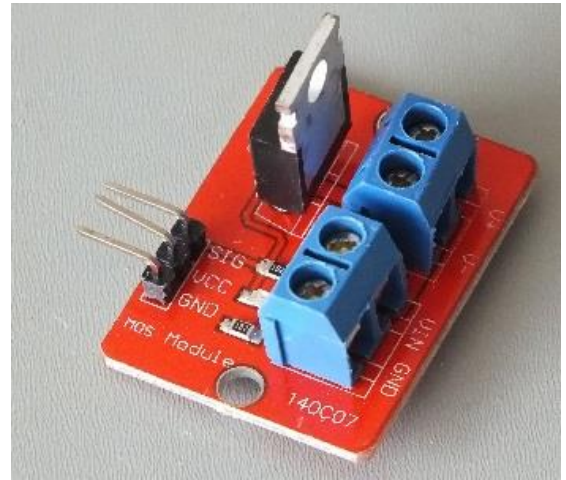
MOSFET-Schalter

Das Schalten ist auch mit MOSFET-Verstärkern möglich.

Diese sind für das Schalten von Weichenspulen und LED-Streifen sehr gut geeignet.

Sie können auch zum Dimmen mittels Pulsweitenmodulation (PWM) eingesetzt werden. Hierfür können Sie ein vorgefertigtes Modul verwenden oder Sie können die Bauteile selbst auf eine Montageplatine löten.

Zu den geeigneten MOSFET-Typen gehören die Typen FQP30N06L und RFP30N06LE. Hierbei steht das 'L' steht für 'Logikpegel'. Das bedeutet, dass diese MOSFETs bei einer Steuerspannung von 5V voll leitfähig sind und fast keinen Widerstand mehr haben.



Software

Natürlich wird auch Software benötigt.

Während der Softwareinstallation von Arcomora auf dem PC wird das MARDEC-Programm als "vorkompilierte" Binärdatei in folgender Datei abgelegt: MARDEC.hex.

Diese Datei muss nun auf dem Arduino installiert werden, dazu benötigen Sie eine Upload-Software. Hiermit wird MARDEC von Ihrem PC über das USB-Kabel auf den Arduino übertragen. Dieser Upload wird von einem Programm ausgeführt, das mit einer Verknüpfung auf dem Desktop gestartet wird.

Für die Kommunikation mit MARDEC wird ein weiteres Programm verwendet (Putty.exe). Dieses wird ebenfalls mit einer Verknüpfung auf dem Desktop gestartet. Hiermit kann der Ablauf des auf dem Arduino laufenden Programms angezeigt werden und es können gleichzeitig auch Eingaben von der Tastatur an den Arduino gesendet werden.

Darüber hinaus können auch weitere Treiber erforderlich sein. Wenn Sie die Arduino IDE (Integrated Development Environment) bereits installiert haben, haben Sie diese Treiber wahrscheinlich bereits. Für den chinesischen Klon Arduino's und den **DCCNext** benötigen Sie einen speziellen Treiber, nämlich den sogenannten CH340-Treiber.

Diese vier Komponenten werden in einer einzigen Installation auf Ihrem PC installiert.

Siehe Installationshandbuch.

[illegible]

MARDEC Konfiguration

Nachdem die gesamte Software korrekt installiert wurde und der DCCNext oder der Arduino angeschlossen ist, kann der Konfigurationsprozess beginnen.

Klicken Sie dazu auf das Icon "MARDEC" auf Ihrem Desktop.

Nur wenn Sie dies zum ersten Mal tun, werden zuerst die USB-Treiber geladen und erst danach erfolgt das Hochladen des Decoderprogramms in den Arduino. Ansonsten startet die Kommunikation mit dem Decoderprogramm immer sofort. (Siehe auch das Installationshandbuch)

- Beachten Sie bitte folgende Punkte:
- beenden Sie jede Eingabe mit <enter>
- alle alphanumerischen Befehle (Buchstaben) werden nicht mit <enter> abgeschlossen
- bei den meisten Einträgen ändert nur der Befehl <enter> den vorhandenen Wert nicht,
- der vorhandene Wert steht dann in Klammern.
- im Konfigurationsmodus leuchtet eine LED am Arduino kontinuierlich
- im Betriebszustand ist diese LED erloschen
- stellen Sie sicher, dass das NumLock vor dem Ziffernblock Ihres PC eingeschaltet ist
- Sie können sowohl Groß- als auch Kleinbuchstaben eingeben.
- Verwenden Sie zur Korrektur nur die Rück-Taste.

Beim ersten Start des MARDEC oder nach einem vollständigen Reset (R-Befehl) werden zunächst einige allgemeine Daten abgefragt.

Deren Bedeutung finden Sie weiter unten im I-Befehl.

Bei jedem Start des Programms MARDEC wird automatisch eine Übersicht über die Konfiguration angezeigt. Nun können Befehle eingegeben werden.

Die Befehle

Es gibt vier Arten von Befehlen:

Allgemeine Befehle,

Servo- Befehle,

hiermit werden Servos konfiguriert

Zubehör-Befehle,

hiermit wird Zubehör konfiguriert

Eingangs-Befehle,

hiermit werden Eingänge konfiguriert

Der Konfigurationsprozess umfasst also die obigen vier verschiedenen Zustände.

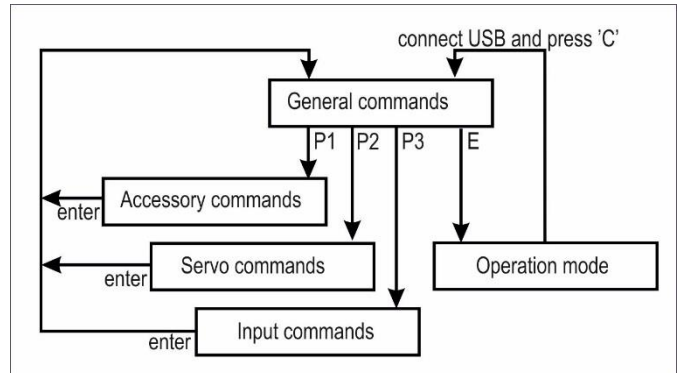
Die Eingabe eines '?' (Fragezeichen) gibt einen Überblick über die verfügbaren Befehle.

Hierbei sehen Sie nur die Befehle, die zum aktuellen Zustand gehören.

Nach dem ersten Programmstart im läuft das Programm automatisch im Konfigurationsmodus.

Jeder alphanumerische Befehl (einzelner Buchstabe) wird **ohne** <enter> abgeschlossen.

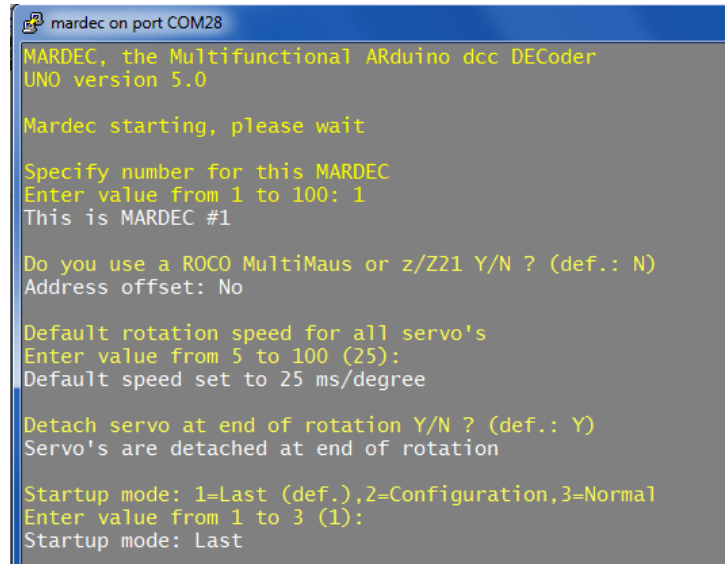
Der wichtigste Befehl ist <P> (Port). Damit wird ein Arduino-Port ausgewählt.



Allgemeine Befehle

I-Befehl

Der Befehl <I> setzt einige Grundeinstellungen zurück. die Verwaltungsnummer für diesen MARDEC, diese Nummer identifiziert den Decoder. verwenden Sie eine Roco Multimaus, z21 oder Z21? in diesem Fall führt MARDEC automatisch eine Adresskorrektur durch. (Standard=Nein)
ACHTUNG: Schalten Sie die Kommunikationsfunktion Railcom/Rail bei der z21 aus!
die Standard-Drehgeschwindigkeit der Servos. Dies betrifft neu hinzugefügte Servos.
im Normalzustand normalerweise werden die Servos nach Erreichen der jeweiligen Endposition vom Arduino getrennt. Dies soll das "Zittern" der Servos verhindern. Dadurch könnte sich das Servo drehen, wenn eine externe Kraft auf das Servo einwirkt. Wenn dies zu Problemen führt, kann die Einstellung so geändert werden, dass das Servo mit dem Arduino verbunden bleibt. MARDEC wird dann jede Abweichung sofort korrigieren. Sie sollten dies nur in Ausnahmefällen tun.
Sie können wählen, wie MARDEC bei diesem Decoder zukünftig starten soll: wie im letzten Modus, immer im Konfigurationsmodus oder immer im Betriebsmodus.



```
mardec on port COM28
MARDEC, the Multifunctional ARduino dcc DECoder
UNO version 5.0

Mardec starting, please wait

Specify number for this MARDEC
Enter value from 1 to 100: 1
This is MARDEC #1

Do you use a ROCO MultiMaus or z/Z21 Y/N ? (def.: N)
Address offset: No

Default rotation speed for all servo's
Enter value from 5 to 100 (25):
Default speed set to 25 ms/degree

Detach servo at end of rotation Y/N ? (def.: Y)
Servo's are detached at end of rotation

Startup mode: 1=Last (def.),2=Configuration,3=Normal
Enter value from 1 to 3 (1):
Startup mode: Last
```

P-Befehl (Port)

Durch die Eingabe von P kann eine Portnummer gewählt werden.
Wenn der Port noch nicht genutzt wurde, müssen Sie zunächst die DCC-Adresse eingeben, dann müssen Sie entscheiden, ob es sich um einen Eingangs-, Servo- oder Zubehörport handeln soll.
Danach können Sie Eingangs-, Servo-, oder Zubehör-Befehle eingeben.

Wenn der Port bereits konfiguriert ist, wird bei einer Eingabe die Einstellung des Ports angezeigt und der entsprechende Befehl sofort ausgeführt.
Der Eingang/Servo/Zubehör/ das an diesem Port angeschlossen ist, kann dann eingestellt werden.
Zubehör vom Typ PWM (Modus 7 und 8; siehe M-Befehl für Zubehör) ist nur an den Ports 1, 3, 4 und 9 verfügbar.



```
Enter command (P/A/D/E/I/R/?): P

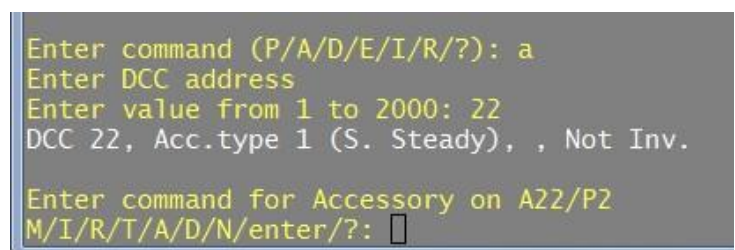
Select port number.
Enter value from 1 to 16: 1
Port is undefined. First enter DCC address

Set DCC address for port 1
Enter value from 1 to 2000: 1
DCC Address set to 1

Select port type: Accessory(A), Servo(S) or Input(I):
```

A-Befehl (Adresse)

Wenn Ports bereits konfiguriert sind, können Sie auch die gewünschte DCC-Adresse eingeben. MARDEC wird dann den richtigen Port finden.



```
Enter command (P/A/D/E/I/R/?): a
Enter DCC address
Enter value from 1 to 2000: 22
DCC 22, Acc.type 1 (S. Steady), , Not Inv.

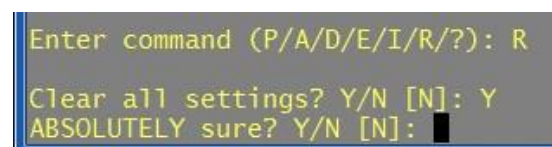
Enter command for Accessory on A22/P2
M/I/R/T/A/D/N/enter/?:
```

D-Befehl (Dokumentierung)

Der Befehl D gibt einen Überblick darüber, wie alle Ports eingerichtet sind.
Machen Sie einen Screenshot davon und speichern Sie diesen auf Ihrem Computer und/oder drucken Sie ihn auf Ihrem Drucker. So kennen Sie immer sämtliche Einstellungen des Decoders.

R-Befehl (Reset)

Der Befehl R löscht sämtliche Einstellungen aus dem Speicher des Arduinos.
Dessen Bestätigung wird zur Sicherheit zweimal angefordert.



```
Enter command (P/A/D/E/I/R/?): R

Clear all settings? Y/N [N]: Y
ABSOLUTELY sure? Y/N [N]:
```

E-Befehl (Exit)

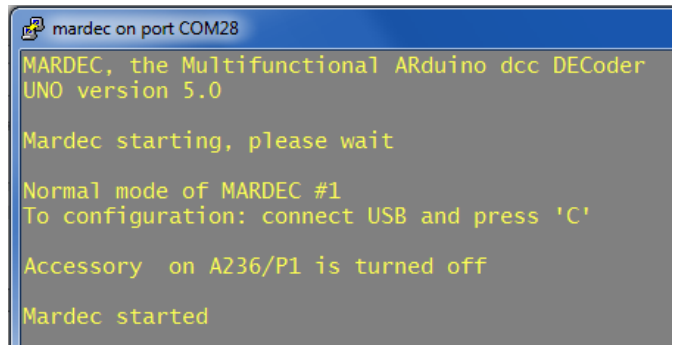
Mit dem Befehl E wird der Betriebsmodus gestartet.

Das USB-Kabel wird nun nicht mehr benötigt.

Sie können jetzt mit Ihrer DCC-Zentrale Servos und Zubehör steuern.

Die Status-LED am Arduino ist jetzt ausgeschaltet. Hieran erkennen Sie, dass sich der Arduino im Betriebsmodus befindet. Für jede konfigurierte DCC-Adresse, die empfangen wird, zeigt die LED einen kurzen Lichtimpuls.

Sie können nun das Bedien-Programm schließen und das USB-Kabel entfernen.



```
mardec on port COM28
MARDEC, the Multifunctional ARduino dcc DECoder
UNO version 5.0

Mardec starting, please wait

Normal mode of MARDEC #1
To configuration: connect USB and press 'C'

Accessory on A236/P1 is turned off

Mardec started
```

Um in den Konfigurationsmodus zurückzukehren, schließen Sie das USB-Kabel wieder an, starten Sie dann das MARDEC Programm auf dem Desktop und drücken anschließend die Taste C.

Zurück zur Konfiguration

Schließen Sie das USB-Kabel wieder an, um zum Konfigurationsmodus zurückzukehren. Starten Sie dann das Konfigurationsprogramm "Mardec" über die entsprechende Desktop-Verknüpfung. Dies startet den Arduino neu, was Sie daran sehen können, dass die Status-LED dreimal kurz blinkt und dann ausgeschaltet bleibt. Drücken Sie jetzt die C-Taste, dadurch wechselt MARDEC in den Konfigurationszustand und die Status-LED leuchtet dauerhaft.

Sie können auch die Taste "P" drücken. Dann wird der Konfigurationsmodus ebenfalls gestartet, aber der Befehl "P" wird ebenfalls sofort gegeben und darum fragt die Software sofort nach einer Portnummer.

Das ist der sogenannte **Quick-Config-Mode**.

Mit dem Quick Config-Mode ist es möglich, schnell einen Port aus dem Betriebsmodus heraus zu ändern. Durch das Aufrufen von "P" starten Sie den Konfigurationsmodus und fordern gleichzeitig eine Portnummer an.

Nach der Änderung muss der Port nicht erst mit <enter> gespeichert werden, sondern der E-Befehl kann ebenfalls direkt gegeben werden.

WICHTIG

Nach jedem empfangenen DCC-Befehl wird der geänderte Status eines Servos oder Zubehörs direkt in den Speicher des Arduinos geschrieben. Nach dem Ausschalten werden beim erneuten Einschalten des Arduinos diese gespeicherten Zustände wieder ausgelesen und die Servos und Zubehöre in den Zustand zum Zeitpunkt des Ausschaltens des Arduino gesetzt.

Es ist möglich, dass sich ein Servo beim Einschalten des Arduinos mehr oder weniger bewegt.

Da Mardec die Servos mit einiger Verzögerung auf die zuletzt eingestellte Position zurücksetzt, haben diese Anlaufeffekte keinen Einfluss auf die Positionierung. Eine Start-Verdrehung der Servos wird von MARDEC wieder ausgeglichen. Eine gute Stromversorgung ist jedoch nach wie vor Voraussetzung, ebenso wie die Verwendung hochwertiger Digitalservos (z.B. TG9d, HXT900, ES9051).

Optional kann bei Bedarf auch ein Ferritkern im Servo-Anschlusskabel in der Nähe des Servos helfen, ebenso ein 4k7-Widerstand zwischen der Signal- und der + 5V-Leitung des jeweiligen Servos.

Quick Config Mode

In configuration mode:

Enter command (P/A/D/E/I/R/?): **P**

Select port number.

Enter value from 1 to 16: 6

DCC 55, Acc.type 3 (S. Flashing), , Not Inv., Time(ms) 200/300

Enter command for Accessory on A55/P6

M/I/R/T/A/D/N/E/enter/?: **M** **Modify**

Select mode for this accessory

0=Help. Enter value from 0 to 9 (3):

Mode set to Single Flashing

Enter 'On' time in millisec.

Enter value from 5 to 30000 (200):

'On' time set to 200 msec.

Enter 'Off' time in millisec.

Enter value from 5 to 30000 (300): **400** **Off time changed**

'Off' time set to 400 msec.

Enter command for Accessory on A55/P6

M/I/R/T/A/D/N/E/enter/?: **E** **Direct E-command**

Port settings are saved

Normal mode starts

MARDEC, the Multifunctional ARduino dcc DECoder
UNO version 6.0

Mardec starting, please wait

Normal mode of MARDEC #1

To configuration: connect USB and press '**C**' or '**P**'

Accessory on A12/P4 is turned off

Accessory on A55/P6 stopped flashing

Servo on A1/P1 set to 126 degrees

Mardec started

P Eingabe von P

configuration mode starts

MARDEC, the Multifunctional ARduino dcc DECoder
UNO version 6.0

Mardec starting, please wait

Configuration mode of MARDEC #1

...

Mardec started

Select port number.

Automatisches P-commando

Enter value from 1 to 16: 6

DCC 55, Acc.type 3 (S. Flashing), , Not Inv., Time(ms) 200/**400**

Enter command for Accessory on A55/P6

M/I/R/T/A/D/N/E/enter/?:

Mit dem Quick-Config-Modus ist es möglich, schnell eine Änderung für einen Port aus dem Betriebszustand heraus vorzunehmen. Eingabe von P Start des Konfigurationsmodus und Abfrage der Portnummer. Nach der Änderung muss der Port nicht erst mit <enter> gespeichert werden, sondern es kann direkt der Befehl E gegeben werden

Eingangs-Befehle

Mit einem als Eingang konfigurierten Port kann z.B. das DCC-Signal für ein Servo oder für Zubehör simuliert werden, das die gleiche Adresse wie der Eingangsport hat.

Beispiel:

An Port 12 ist als Zubehör eine einzelne blinkende LED mit der DCC-Adresse 34 konfiguriert.

An Port 5 wird ein Eingang ebenfalls mit der DCC-Adresse 34 konfiguriert.

Standardmäßig stellt MARDEC einen Eingangsport auf 'High' (5 V). Wird dieser nun auf 'Low' (0V=GND) gesetzt, erkennt MARDEC dies als DCC-Signal für die Adresse 34, und die blinkende LED auf Port 12 erlischt oder beginnt zu blinken, je nach Einstellung. Gibt es auch ein Servo an einem anderen Port mit der Adresse 34 gibt, dreht sich das Servo ebenfalls.

Ein Eingang kann auf 3 Arten aktiviert werden:

indem der Eingang auf 'Low' gesetzt wird. Es passiert dann nichts, wenn der Eingang wieder 'High' wird (Typ D)

indem der Eingang auf 'High' gesetzt wird. Es passiert dann nichts, wenn der Eingang wieder 'Low' wird (Typ U)

Indem Sie den Eingang entweder auf 'Low' oder 'High' schalten. Das Zubehör bzw. das Servo "folgt" dem Eingangssignal, als ob es von einem DCC Signal gesteuert würde. (Typ L)

Mit dieser Option kann beispielsweise eine Weiche auch über einen Weichenschalter umgestellt werden.

Bei den Optionen 1 und 2 wirkt ein Eingang wie ein Momentschalter.

Bei Option 3 wirkt ein Eingang wie ein Umschalter, genau wie bei einem DCC-Befehl.

Die Aktivierungsmethode (D/U/L) wird bei der Konfiguration eines Ports als Eingang angefordert.

Ein Eingangsport kann auf verschiedene Weise angesteuert werden. Z.B. durch einen Reed-Kontakt, der durch einen Magneten auf Masse (GND) geschaltet wird oder durch ein Stromfühlersignal, das beim Durchfahrt eines Zuges 'Low' wird. (Okkie)

Ebenso kann eine Infrarot-Lichtschanke oder einfach nur ein Umschalter verwendet werden.

```
Enter command (P/A/D/E/I/R/?): P
Select port number.
Enter value from 1 to 16: 2
Port is undefined. First enter DCC address

Set DCC address for port 2
Enter value from 1 to 2000: 2
DCC Address set to 2

Select port type: Accessory(A), Servo(S) or Input(I): I
Port 2 set as Input port.

Select trigger moment Up(U), Down(D) or Both(B): B
Trigger set to type Both

Enter command for Input on A2/P2
T/2/3/enter/A/N/D/? : █
```

Achtung:

Bitte beachten Sie, dass ein Zubehör oder ein Servo welches Sie über die Steuerungszentrale per DCC Signal einschalten und dann dieses über einen Eingangsport wieder ausschalten, zu Problemen führen kann, weil die Steuerungszentrale dieses nicht erkennt. Für diese ist der entsprechende Port immer noch eingeschaltet.

Das Ausschalten des DCC Signals an der Zentrale hat dann keine Wirkung mehr.

2. und 3. Befehle (zweite und dritte Adresse)

Eine 2. und 3. DCC-Adresse kann ebenfalls mit einem Eingangsport verknüpft werden. Mit dem Befehl'2' oder'3' kann diese Adresse dann eingegeben werden.

Bei der Aktivierung des Eingangsports wird sowohl das Servo/Zubehör auf der "eigenen" Adresse (1.Adresse) als auch auf der 2. Adresse und, wenn programmiert, auch der 3. Adresse umgesetzt.

Für diese 2. und 3. Adresse kann auch eine Verzögerung von 0 bis 25 Sekunden programmiert werden.

Das Servo/Zubehör an dieser zweiten und dritten Adresse wird dann erst nach Ablauf der Verzögerungszeit geschaltet.

Beide Zeiten werden ab dem Zeitpunkt der Aktivierung des Eingangsports gezählt.

Selbstverständlich müssen alle Adressen auf dem gleichen Arduino konfiguriert sein.

Wenn Sie nur "verzögerte" Aktionen wünschen, können Sie eine Dummy-Adresse als "eigene" Adresse des Eingangsports angeben. Diese beinhaltet dann einen unbenutzten Port, der die 1.Adresse des Eingangs empfängt.

```

Specify port number
Enter value from 1 to 16: 5
Port is undefined. Specify DCC address

Set DCC address for port 5
Enter value from 1 to 2000: 236
Address in use for servo/accessory on A236/P1
Is that OK ? (Y/N, N=default): Y

DCC Address set to 236

Specify Accessory(1), Servo(2) or Input(3)
Enter value from 1 to 3: 3
Port 5 set as Input port.

Specify Type for Input on A236/P5
L=Level, U=Up, D=Down: L
Input set to type Level

Specify action for Input on A236/P5
A/N/?/D/T/2/3/enter: 2
Specify Second control address.
Enter value from 1 to 2000: 120
Specify delay time in 0.1 seconds.
Enter value from 0 to 250: 20
Address 120 set as Second control address with a delay of 2.0 sec.

Specify action for Input on A236/P5
A/N/?/D/T/2/3/enter: 3
Specify Third control address.
Enter value from 1 to 2000: 45
Specify delay time in 0.1 seconds.
Enter value from 0 to 250: 50
Address 45 set as Third control address with a delay of 5.0 sec.

Specify action for Input on A236/P5
A/N/?/D/T/2/3/enter:

Port settings are saved!

Specify action (P/A/R/T/D/E/I/?): D

Settings of MARDEC #1

Default servo rotation speed: 25 ms/degree
Address offset: No
Servo's are detached at end of rotation
Startup mode: Last

Port 1: DCC 236, Acc.type 1 (S. Steady), , Not Inv.
Port 2: DCC 236, Servo , Angles 75/105, Not Inv., Speed 25, Frog port no
Port 3: DCC 120, Acc.type 3 (S. Flashing), , Not Inv., Time(ms) 300/400
Port 4: DCC 45, Servo , Angles 75/105, Not Inv., Speed 25, Frog port 16
Port 5: DCC 236, Input , Type: L, Second addr/delay: 120/2.0 sec, Third addr/delay: 45/5.0 sec

```

Bei diesem Beispiel geschieht folgendes, wenn Port 5 (=Eingang) den Pegel wechselt:

das Zubehör an Port 1 schaltet ein oder aus, da Port 1 die gleiche Adresse wie Port 5 hat

das Servo an Port 2 läuft, da Port 2 ebenfalls die gleiche Adresse wie Port 5 hat

nach 2 Sekunden schaltet sich das Zubehör an Port 3 ein oder aus, da Port 3 die gleiche Adresse wie die zweite Adresse an Port 5 hat

nach 5 Sekunden beginnt das Servo an Port 4 zu laufen, da Port 4 die gleiche Adresse wie die dritte Adresse von Port 5 hat.

Wenn die DCC-Adresse 236 gesendet wird:

das Zubehör an Anschluss 1 schaltet sich ein oder aus

das Servo an Port 2 läuft

Wenn die DCC-Adresse 120 geschaltet wird: das Zubehör an Port 3 geht an oder aus.

Wenn die DCC-Adresse 45 gesendet wird: das Servo an Port 4 läuft.

A-Befehl (Adresse)

Mit dem Befehl A wird die DCC-Adresse des Eingangs eingestellt. Sie erhalten eine Warnung, wenn die Adresse bereits einem anderen Servo, Zubehör oder Eingang zugeordnet wurde. Wenn Sie das jedoch akzeptieren, können Sie mit einer Adresse mehrere Servos/Zubehörteile gleichzeitig steuern.

```
Select port number.  
Enter value from 1 to 16: 1  
DCC 1, Input , Trigger: B, Second addr/delay: none, Third addr/delay: none  
  
Enter command for Input on A1/P1  
T/2/3/enter/A/N/D/? : a  
  
Set DCC address for port 1  
Enter value from 1 to 2000: 5  
DCC Address set to 5
```

N-Befehl (Note)

Mit dem Befehl N können Sie dem Eingangsport eine Verwaltungsnummer zuweisen. Sie hat keine technische Bedeutung und muss nicht eingegeben werden (4 Zeichen).

T-Befehl (Trigger)

Mit dem Befehl T können Sie die Art des Eingangs ändern.

```
Enter command for Input on A5/P1  
T/2/3/enter/A/N/D/? : T  
  
Select trigger moment Up(U), Down(D) or Both(B): D  
Trigger set to type Down-Pulse
```

D-Befehl (Dokumentierung)

Der Befehl D zeigt eine Übersicht aller Einstellungen.

Prüfung

Es gibt keine separate Testfunktion für Eingangsports. Sie können sie jedoch testen, indem Sie mit dem Befehl E (Exit) in den Betriebsmodus wechseln. Indem Sie den Eingangsport mit einem Kabel an einen GND-Port (Masse) anschließen, können Sie überprüfen, ob das angeschlossene Servo/Zubehör korrekt reagiert.

I- Befehl (invertieren)

Mit dem I-Befehl wird der Eingang invertiert. Das bedeutet, dass ein Signal "Ein" (High) am Eingang dann als ein Signal "Aus" (Low) angesehen wird und umgekehrt.

R-Befehl (Reset - Zurücksetzen)

Mit dem R-Befehl können Sie den Port zurücksetzen. Der Port kann dann neu konfiguriert werden.

<enter oder X> Befehl

Die Eingabe von <enter> oder X speichert die Einstellungen eines Ports und kehrt zurück zu "Allgemeine Befehle", es kann dann ein anderer Port ausgewählt werden

```
Enter command (P/A/D/E/I/R/?): p  
  
Select port number.  
Enter value from 1 to 16: 1  
DCC 1, Acc.type 1 (S. Steady), , Not Inv.  
  
Enter command for Accessory on A1/P1  
M/I/R/T/A/D/N/enter/? : r  
  
Do you want to reset port 1? Y/N [N]: y  
Port 1 is now reset  
  
Port settings are saved!  
  
Enter command (P/A/D/E/I/R/?):
```

? - Befehl (Hilfe)

Hiermit werden die verfügbaren Eingabebefehle angezeigt

E-Befehl (Beenden)

Mit dem E-Befehl werden die Änderungen der Ports sofort gespeichert und der Betriebs-Modus gestartet. Die Konfiguration des Ports muss dann nicht erst mit <enter> oder X beendet werden

Servo-Befehle

Der Dialog zur Konfiguration eines "neuen" Servos sieht wie folgt aus:

An einem zuvor ungenutzten Port wird das Servo automatisch auf Winkel von 75Grad 'Low' und 105Grad 'High' eingestellt(Standardwerte). Ein zuvor bereits an diesem Port eingestelltes Servo behält selbstverständlich seine bereits eingestellten Winkel.

In beiden Fällen wird das Servo in die Position 'Low' gefahren.

- (min) Befehl

Der Befehl '-' reduziert den jeweiligen Winkel pro Tastendruck um 1Grad.

Hierbei muss der Winkel 'High' um mindestens 5Grad größer sein als der Winkel 'Low'.

+ (plus) Befehl

Der Befehl '+' erhöht den Winkel pro Tastendruck um 1Grad.

Hierbei muss der Winkel 'Low' mindestens 5Grad kleiner sein als der Winkel 'High'.

9-Befehl

Der Befehl 9 stellt das Servo auf 90 Grad ein. Dies ist besonders dafür gedacht, um das Servo in Mittelstellung zu bringen und dann den Servoarm entsprechend zu montieren.

C-Befehl (Umschalten)

Der Befehl C schaltet zwischen den beiden Winkeln um. Nun kann der andere Winkel mit den Befehlen plus und minus eingestellt werden.

Wenn beide Positionen eingestellt sind, können Sie das Servo testen, indem Sie den C-Befehl einige Male eingeben. Dies ist auch mit dem T-Befehl (Test)möglich, allerdings läuft dieser dann automatisch ab.

I-Befehl (Invertieren)

Je nachdem wie das Servo montiert wird, kann es erforderlich sein, die Servostellung zu invertieren (umzukehren), damit z.B die Weichenstellung (Gerade/Abzweig) mit der DCC Zentrale bzw. der Anzeige auf dem PC Steuerungsprogramm übereinstimmt. Sollte dies nicht der Fall sein, so wechseln Sie mit dem Befehl I zwischen 'nicht invertiert und 'invertiert'.

T-Befehl (Test)

Der T-Befehl ist der gleiche wie der C-Befehl. Der simulierte DCC-Zustand wird zusätzlich angezeigt, und der eingestellte Winkel wird direkt angefahren und angezeigt

S-Befehl (Geschwindigkeit)

Mit dem Befehl S können Sie die individuelle Drehgeschwindigkeit des ausgewählten Servos einstellen.

Diese kann zwischen 5 und 100 Millisekunden pro Grad eingestellt werden. Dabei sind 5 ms. sehr schnell und 100 ms. sehr langsam.

Die Standardeinstellung ist 25 ms. Diese ist auch mit dem Befehl S als Generalbefehl einstellbar.

```
Specify action (P/A/R/T/D/E/I/?): P
Specify port number
Enter value from 1 to 16: 4
Port is undefined. Specify DCC address
Set DCC address for port 4
Enter value from 1 to 2000: 32
DCC Address set to 32
Specify Accessory(1), Servo(2) or Input(3)
Enter value from 1 to 3: 2
Port 4 set as Servo port.
Servo on A32/P4 set to 75 degrees
Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: -
LOW angle: 74 degrees
Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: -
LOW angle: 73 degrees
Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: C
Servo on A32/P4 set to 105 degrees
Angle set to HIGH: 105
Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: +
HIGH angle: 106 degrees
Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: +
HIGH angle: 107 degrees
Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: N
Specify administrative note: W12
Administrative note set to W12
Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: S
Set speed for servo on A32/P4
Enter value from 5 to 100 (25): 15
Speed set to 15 ms/step
Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter:
Port settings are saved!
```


F-Befehl (Herzstück-Polarisierung)

Mit dem Befehl F können Sie automatisch einen Port bestimmen, an den Sie ein Relais zur Herzstück-Polarisierung anschließen können. Wenn Sie ein Relais wünschen, geben Sie hier eine '1' für 1Relais oder eine '2' für 2Relais ein (siehe unten).

MARDEC wählt hierbei selbstständig einen freien Port zwischen 1 und 16 aus.

Wenn bereits ein Port für die Polarisation zugewiesen wurde, fragt MARDEC, ob Sie diesen entfernen (R) oder das Relais invertieren möchten (1 für das 1.Relais, 2 für das 2.Relais).

```
Specify port number
Enter value from 1 to 16: 4
DCC 32, Servo W12, Angles 73/107, Not Inv., Speed 15, Frog port no
Angle set to HIGH: 107

Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: f
How many relays ? (0/1/2): 2

Port 15 is set as first frogpoint port for servo on A32/P4, No inversion
Port 13 is set as second frogpoint port for servo on A32/P4, No inversion
```

Das jeweilige Relais schaltet genau bei der Hälfte des Servo-Drehwinkels um. Beim Drehen des Servos auf den Winkel 'High' wird das Relais aktiviert (eingeschaltet) und beim Drehen auf den Winkel 'Low' deaktiviert (ausgeschaltet). Sollte dies falsch sein, können Sie mit der Option 'Invertierung' das Relais umgekehrt aktivieren bzw. deaktivieren.

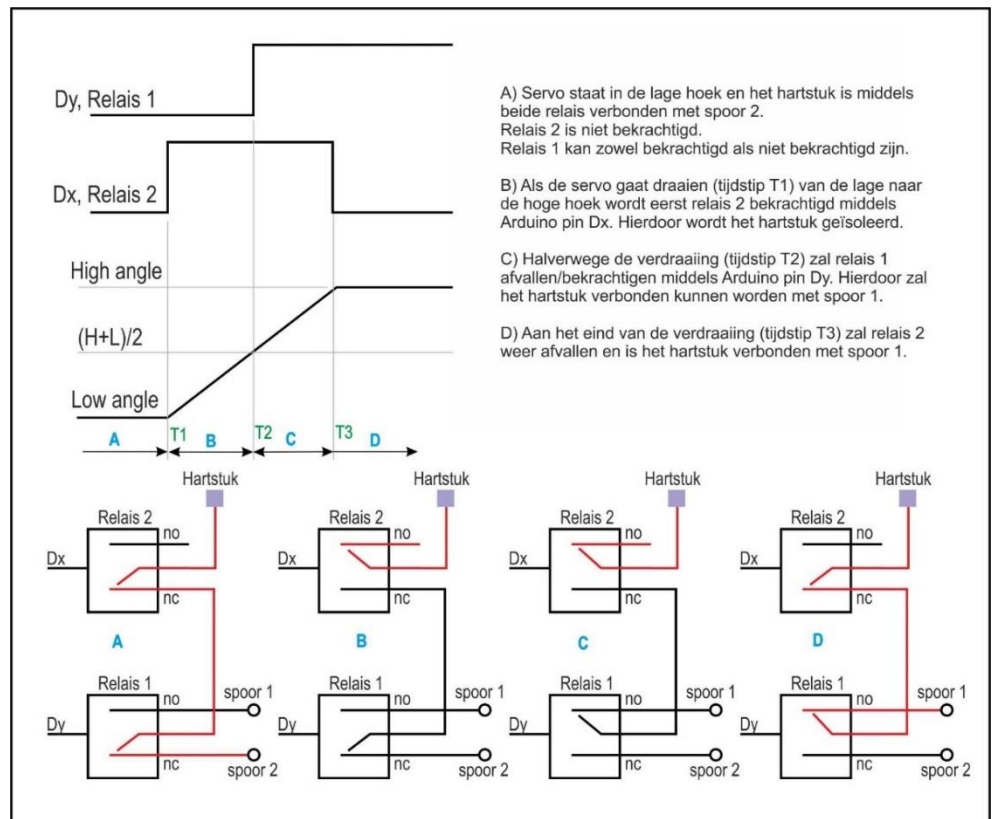
Da das Servo selbst auch über die Möglichkeit des Invertierens verfügt, können Sie die Position der Weiche und deren Herzstück-Polarisierung der Punkte mit Hilfe der Software einstellen.

```
Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: F

Remove frogpoint or Toggle Port relay inversion? R/1/2: 1
First Relay inverted
```

Sie können auch zwei Relais verwenden. Das zweite Relais schaltet dann erst, wenn die Drehung des Servos beendet wird. Für zwei Relais geben Sie eine '2' ein. Da einige Relais bei 0V anstatt bei 5V eingeschaltet sind, können Sie auch die Invertierung des zweiten Relais einstellen. Das Relaismodul, wie auf Seite 7 beschrieben, ist ein solches "0V" Relais. Siehe auch Seite 27.

Standardmäßig sind die Relais auf "nicht invertiert" gesetzt. Durch erneutes Eingeben des F-Befehls können Sie die Relais invertieren.



TIPP: Natürlich können Sie auch etwas anderes als ein Relais an den Herzstück-Port anschließen, z.B. ein Weichensignal oder eine LED, die die Position der Weiche anzeigt. Sie können ein Relais auch verwenden, um etwas anderes als die Herzstück-Polarisierung zu schalten. Im Großen und Ganzen ist dieser Port einfach ein Anschluss, der bei Umstellung der Weiche die Werte 'High' oder 'Low' annimmt.

A-Befehl (Address)

Mit dem Befehl A wird die DCC-Adresse eines Ports eingestellt. Sollte diese Adresse bereits einem anderen Port zugeordnet sein, erhalten Sie eine Warnung. Wenn Sie das bestätigen, können Sie mit einer Adresse mehrere Servos/Zubehöre gleichzeitig steuern.

N-Befehl (Note)

Mit dem N-Befehl können Sie Weichen/Servos/Zubehören eine 4-stellige Verwaltungsbezeichnung zuweisen. Diese hat keine technische Bedeutung und keinen Einfluss auf die Programmierung. Sie erhalten eine Warnung, wenn die Nummer bereits einem anderen Servo/Zubehör zugeordnet wurde.

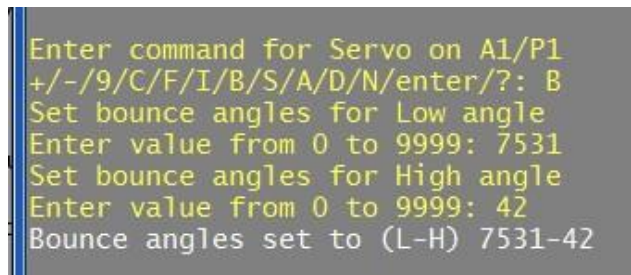
B- Befehl (Bouncing)

Mit dem B-Befehl können Sie den Servoarm am Ende der Drehung mehrmals (0-4) "nachwippen" lassen. Der Winkel, in dem sich der Arm zurückdreht, ist von 1 bis 9 Grad einstellbar. Die Winkel werden als eine Anzahl von bis zu 4 Ziffern eingegeben.

Beispiel 1: Ein Wert von 7531 bedeutet, dass das Servo zuerst 7 Grad zurück und dann wieder in die Endposition fährt. Danach fährt das Servo 5 Grad zurück und wieder in die Endposition. Danach fährt das Servo 3 Grad zurück und wieder in die Endposition. Schließlich geht das Servo 1 Grad zurück und wieder in die Endposition.

Beispiel 2: Bei einem Wert von 42 läuft das Servo zunächst um 4 Grad und danach um 2 Grad zurück, bevor es in der Endposition verbleibt

Die "Nachwipp-Winkel" sind für beide Endpositionen getrennt einstellbar



D-Befehl (Dokumentierung)

Der Befehl D zeigt eine Übersicht aller Einstellungen.

R-Befehl (Reset - Zurücksetzen)

Mit dem R-Befehl können Sie den Port zurücksetzen. Der Port kann dann neu konfiguriert werden.

<enter oder X> Befehl

Die Eingabe von <enter> oder X speichert die Einstellungen eines Ports und kehrt zurück zu "Allgemeine Befehle", es kann dann ein anderer Port ausgewählt werden

? - Befehl (Hilfe)

Hiermit werden die verfügbaren Eingabebefehle angezeigt

E-Befehl (Beenden)

Mit dem E-Befehl werden die Änderungen der Ports sofort gespeichert und der Betriebs-Modus gestartet. Die Konfiguration des Ports muss dann nicht erst mit <enter> oder X beendet werden

Zubehör Funktionen

Die DCC-Signaldaten für ein Zubehörteil können nur zwei Werte annehmen (0/1, ein/aus, hoch/niedrig, Gerade/Abzweig). Mit diesen Informationen kann MARDEC verschiedene Funktionen realisieren.

Die unten erwähnten "doppelten" Funktionen (2, 4 und 6) erfordern die Verwendung eines zweiten Anschlusses. Dies ist der sogenannte "Buddy-Port".

Welcher Port der "Buddy-Port" werden soll können Sie selbst bestimmen. MARDEC wird Sie danach fragen: *Set Buddy-Port*. Geben Sie hier den von Ihnen gewünschten Port mit einem Wert von 1 bis 16 ein.

Für einen noch nicht eingerichteten Port wird zunächst die Eingabe der gewünschten Funktion (Mode) angefordert.

Mit dem M-Befehl, kann die Funktion (Mode) eines Ports geändert werden.
Die folgenden Funktionen sind verfügbar:

Enter command (P/A/D/E/I/R/?): **P**
Select port number.
Enter value from 1 to 16: **1**
Port is undefined. First enter DCC address

Set DCC address for port 1
Enter value from 0 to 2000: **1**
[DCC Address set to 1](#)

Select port type: Accessory(A), Servo(S) or Input(I): **A**
[Port 1 set as Accessory port.](#)

Select mode for this accessory
0=Help. Enter value from 0 to 9 (9): **0**
1 Single steady, 2 Double steady
3 Single flashing, 4 Double flashing
5 Single One shot, 6 Double One shot
7 Analog PWM, 8 Flickering
9 Random On/Off

Mode 1, Single Steady

Enter value from 0 to 9: **1**
[Mode set to Single Steady](#)

Enter command for Accessory on A1/P1
M/I/R/T/A/D/N/enter/?: **M**

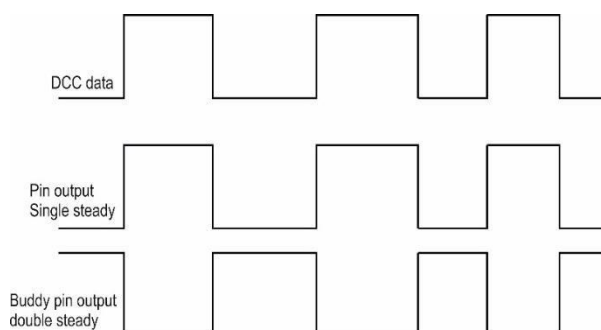
Mode 2, Double Steady

Select mode for this accessory
0=Help. Enter value from 0 to 9 (1): **2**
Set Buddy port. Enter value from 1 to 16: **2**
[Port 2 set as buddy port](#)
[Mode set to Double Steady](#)

Enter command for Accessory on A1/P1
M/I/R/T/A/D/N/enter/?: **M**

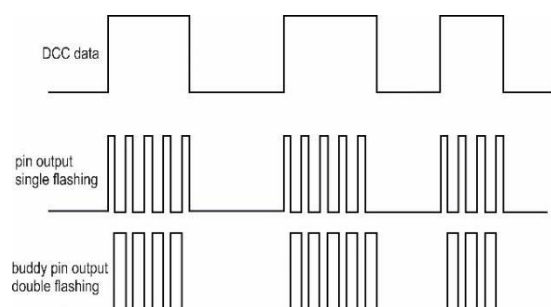
Mode 3, Single Flashing

Select mode for this accessory
0=Help. Enter value from 0 to 9 (2): **3**
[Mode set to Single Flashing](#)
Enter 'On' time in milliseconds Enter value from 5 to 30000: **200**
['On' time set to 200 msec.](#)
Enter 'Off' time in milliseconds Enter value from 5 to 30000: **400**
['Off' time set to 400 msec.](#)



Bei "Mode 1" folgt der Port dem jeweiligen DCC-Signal.

Bei "Mode 2" folgt der gewählte Port dem DCC-Signal, ein zweiter Bei "Mode 1" folgt der Port dem jeweiligen DCC-Signal.



In "Mode 3" ist der Port abwechselnd hoch und niedrig, solange das DCC-Signal 1 (hoch/ein) ist.
Die Ein- und Ausschaltzeiten sind separat einstellbar zwischen 5 und 30.000 ms. (=30 Sek.).

Enter command for Accessory on A1/P1
M/I/R/T/A/D/N/enter/?: **M**

Mode 4, Double Flashing

Select mode for this accessory

0=Help. Enter value from 0 to 9 (3): **4**

Set Buddy port. Enter value from 1 to 16: **2**

Port 2 set as buddy port Mode set to Double Flashing

Enter 'On' time in milliseconds Enter value from 5 to 30000: **500**

'On' time set to 500 msec.

Enter 'Off' time in milliseconds Enter value from 5 to 30000: **100**

'Off' time set to 100 msec.

Enter command for Accessory on A1/P1

M/I/R/T/A/D/N/enter/?: **M**

Mode 5, Single One Shot

Select mode for this accessory

0=Help. Enter value from 0 to 9 (4): **5**

Mode set to Single One shot Enter 'On' time in milliseconds Enter value from 5 to 30000: **500**

'On' time set to 500 msec.

Select trigger moment Up(U), Down(D) or Both(B): **D**

Trigger set to type Down-Pulse

Enter command for Accessory on A1/P1

M/I/R/T/A/D/N/enter/?: **M**

Mode 6, Double One Shot

Select mode for this accessory

0=Help. Enter value from 0 to 9 (5): **6**

Set Buddy port. Enter value from 1 to 16: **2**

Port 2 set as buddy port

Mode set to Double One shot

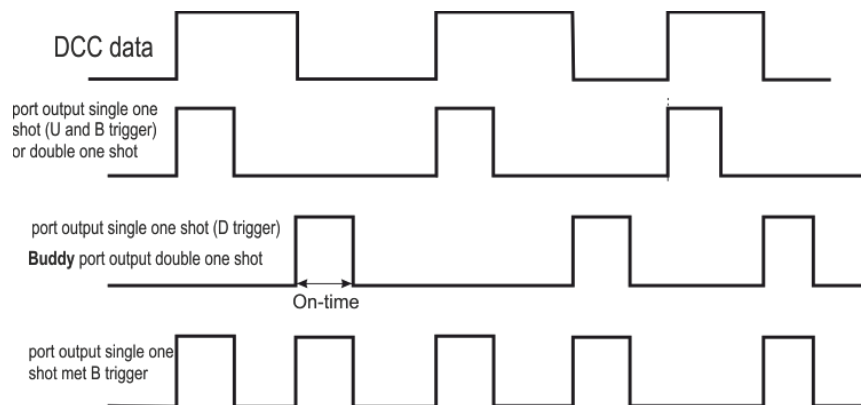
Port 2 is assigned as buddy port to accessory on A1/P1

Enter 'On' time in milliseconds

Enter value from 5 to 30000: **1000**

'On' time set to 1000 msec.

Mode 4 ist ähnlich dem "Mode 3) also einfaches Blinken, aber ein Buddy-Port erhält den entgegengesetzten Wert. Hiermit kann z.B. ein Wechselblinken realisiert werden.



Bei "**Mode 5**" wird der Port für eine wählbare Zeitspanne (5-30.000 ms) 1 (hoch/Ein), wenn das DCC-Signal 1 (hoch/Ein) wird (U- oder B-Auslösung), oder 0(niedrig/Aus) wird (D- oder B-Trigger).

Wenn die Adresse eines "Single Shot" mit der eines anderen an diesem Mardec angeschlossenen Servos oder Zubehörs übereinstimmt, wird sie am Ende des Impulses "umgewandelt", als ob ein DCC-Signal gesendet worden wäre. Beispiel: Bei der Adresse 12 sind sowohl ein einfacher Einzelschuss als auch ein Doppelblinken konfiguriert. Wenn MARDEC nun die Adresse 12 "sieht", werden sowohl der "Single Shot" als auch das Doppelblinken gestartet. Am Ende des Impulses wird das Doppelblinken wieder abgeschaltet. Damit kann z.B. ein Zubehör für kurze Zeit (max. 30 Sek.) aktiviert werden.

Der "Double One Shot" (**Mode 6**) ist ähnlich dem "Single Shot".

Beim "Double One Shot" wird der 1.Port 1 (hoch/Ein), wenn das DCC-Signal 1(Ein) wird. Der 2.Port(Buddy-Port) wird 1 (hoch/Ein), wenn das DCC-Signal 0(Aus) ist.

Mit dieser Funktion können Sie z.B. alternierende Spulen, wie die Wechselfspulen von Weichen steuern. Derartige Wechselfspulen benötigen aber mindestens 12 Volt und nehmen einen relativ großen Strom auf. Daher können Sie diese nicht direkt mit einem Arduino-Port verbinden, sondern dies erfordert eine zusätzliche Verstärkerstufe. Siehe hierzu das Beispiel im Schema auf Seite 11 (Port 16).

Für eine Weiche benötigen Sie zwei Verstärkerstufen, eine für jede Spule. Die auf Seite 10 erwähnten MOSFET-Module sind dafür geeignet.

Enter command for Accessory on A1/P1

M/I/R/T/A/D/N/enter/?: **M**

Mode 7, Analog PWM

Select mode for this accessory

0=Help. Enter value from 0 to 9 (6): **7**

Mode set to Analog PWM value

Set Minimum PWM value for port on A1/P1

Enter value from 0 to 255: **5**

Minimum PWM value set to 5

Set Maximum PWM value for port on A1/P1

Enter value from 5 to 255 (255): **200**

Maximum PWM value set to 200

Set Rise time (sec) for PWM port

Enter value from 0 to 1000: **60**

Rise time set to 60 sec.

Set Fall time (sec) for PWM port

Enter value from 0 to 1000: **30**

Fall time set to 30 sec.

Enter command for Accessory on A1/P1

M/I/R/T/A/D/N/enter/?: **M**

Mode 8, Flickering (PWM)

Select mode for this accessory

0=Help. Enter value from 0 to 9 (7): **8**

Mode set to Flickering(PWM)

Set Minimum PWM value for port on A1/P1

Enter value from 0 to 255: **2**

Minimum PWM value set to 2

Set Maximum PWM value for port on A1/P1

Enter value from 2 to 255 (255): **200**

Maximum PWM value set to 200

Enter command for Accessory on A1/P1

M/I/R/T/A/D/N/enter/?: **M**

Mode 9, Random on/off

Select mode for this accessory

0=Help. Enter value from 0 to 9 (1): **9**

Mode set to Random

Fixed on/off time? 1=No, 2=ON, 3=OFF Enter value from 1 to 3: **2**

Fixed ON time(x0,02 sec) ?

Enter value from 1 to 30000: **1000**

Fixed ON time set to 20.00 sec.

Set Minimum random time (x0.02 sec) Enter value from 1 to 30000: **500**

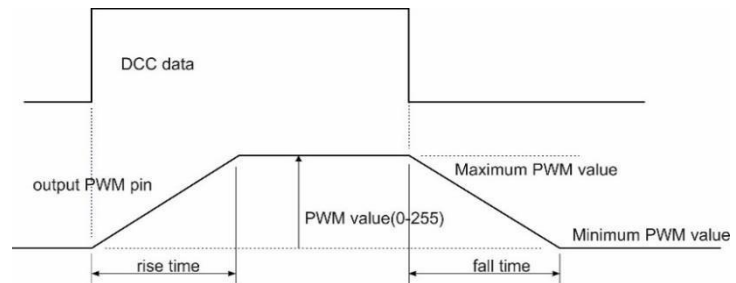
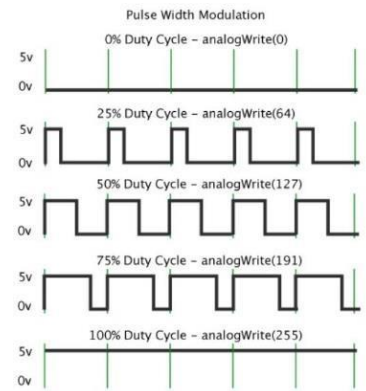
Minimum random time set to 10.00 sec.

Set Maximum random time (x0.02 sec) Enter value from 500 to 30000: **4000**

Maximum random time set to 80.00 sec.

Analog PWM (**Mode 7**): an den Ports 1,3,4 und 9 beim Arduino Uno, sowie an den Ports 3 bis 10 beim Arduino Mega kann der Arduino eine Blockspannung einstellen, deren Pulsbreite (Tastverhältnis Ein-Aus) von 0 bis 100% variieren kann.

Dies wird durch 'Schreiben' eines Wertes von 0 bis 255 bei diesem Port realisiert. Ist an einen solchen Port z.B. eine LED angeschlossen und der PWM-Wert wird entsprechend variiert, kann diese LED gedimmt werden. Dies kann z.B. für eine variable Schienenbeleuchtung in Tag-/Nachtzyklen verwendet werden.



Für diese Funktion können folgende Werte eingestellt werden:

- der maximale PWM-Wert (Mindestwert - 255)
- der minimale PWM-Wert (0 - Maximalwert)
- die Anstiegszeit (0-1000 sec)
- die Fallzeit (0-1000 sec)

Mit 3 derartigen Zubehör-Ports können Sie z.B. einen RGB-LED-Streifen steuern. Allerdings braucht auch ein LED-Streifen eine Verstärkerstufe in Form eines MOSFET-Transistors.

Flickering PWM (**Mode 8**): hiermit können Sie eine angeschlossene LED mit einer einstellbaren minimalen und maximalen Helligkeit flackern lassen.

Alle 20 msec. wird ein Zufallswert für die Helligkeit bestimmt. Durch das Flackern von drei LEDs in verschiedenen Farben können Sie z.B. Feuer simulieren.

Es kann aber auch für eine Schweißlichtsimulation verwendet werden.

Für "Flickering" wird ebenfalls PWM verwendet, so dass diese Funktion nur an den Ports 1, 3, 4 und 9 beim Arduino Uno, bzw. den Ports 3-10 beim Arduino Mega verwendet werden kann.

Random (Zufall **Mode 9**): hier wird ein angeschlossenes Zubehör (z.B. eine LED) kontinuierlich ein- und ausgeschaltet, solange das DCC-Signal 1(hoch) ist. Für jede Ein- und Ausschaltperiode wird eine zufällige Zeit festgelegt. Die minimale und maximale Zeit kann in Schritten von 20msec zwischen 20msec. und 600 Sekunden eingestellt werden. Es kann eine feste Ein- bzw. Ausschaltzeit oder eine völlig zufällige Zeit gewählt werden.

Wenn eine feste Einschaltzeit festgelegt ist, ist die Ausschaltzeit zufällig. Wenn eine feste Ausschaltzeit festgelegt ist, ist die Einschaltzeit zufällig.

Die Zufallsfunktion kann zum Ein- und Ausschalten z.B. der Innenbeleuchtung von Häusern oder der Straßenbeleuchtung verwendet werden.

Mit sehr kurzen Zeiten kann das Flackern einer Lampe simuliert

Zubehör-Befehle

? - Befehl (Hilfe)

Hiermit werden die verfügbaren Eingabebefehle angezeigt

```
Enter command for Accessory on A22/P2
M/I/R/T/A/D/N/enter/?: ?

I Invert port
T Test port
M Modify mode
A DCC-Address
D Display all
N admin. Note
R Reset port
Enter/X Update Port
```

M-Befehl (Modus)

Mit dem M-Befehl können Sie eine andere Funktion für den ausgewählten Port wählen. Es wird nach den spezifischen Einstellungen der neuen Funktion gefragt.

Mit derselben Funktion können Sie auch die Anschlusseinstellungen ändern oder, im Falle einer Doppelfunktion, den Buddy-Anschluss ändern.

Der Befehl M hat eine eigene Hilfefunktion:

```
Enter command for Accessory on A2/P14
M/I/R/T/A/D/N/E/enter/?: m

Select mode for this accessory
0=Help. Enter value from 0 to 9 (1): 0
1 Single steady, 2 Double steady
3 Single flashing, 4 Double flashing
5 Single One shot, 6 Double One shot
7 Analog PWM, 8 Flickering
9 Random On/Off
```

D-Befehl (Dokumentierung)

Der Befehl D zeigt eine Übersicht aller Einstellungen

N-Befehl (Note)

Mit dem N-Befehl können Sie Weichen/Servos/Zubehören eine 4-stellige Verwaltungsbezeichnung zuweisen. Diese hat keine technische Bedeutung und keinen Einfluss auf die Programmierung.

Sie erhalten eine Warnung, wenn die Nummer bereits einem anderen Servo/Zubehör zugeordnet wurde.

R-Befehl (Reset - Zurücksetzen)

Mit dem R-Befehl können Sie den Port zurücksetzen.

Der Port kann dann neu konfiguriert werden.

```
Enter command (P/A/D/E/I/R/?): p

Select port number.
Enter value from 1 to 16: 1
DCC 1, Acc.type 1 (S. Steady), , Not Inv.

Enter command for Accessory on A1/P1
M/I/R/T/A/D/N/enter/?: r

Do you want to reset port 1? Y/N [N]: y
Port 1 is now reset

Port settings are saved!

Enter command (P/A/D/E/I/R/?):
```

<enter oder X> Befehl

Die Eingabe von <enter> oder X speichert die Einstellungen eines Ports und kehrt zurück zu

"Allgemeine Befehle", es kann dann ein anderer Port ausgewählt werden

E-Befehl (Beenden)

Mit dem E-Befehl werden die Änderungen der Ports sofort gespeichert und der Betriebs-Modus gestartet.

Die Konfiguration des Ports muss dann nicht erst mit <enter> oder X beendet werden

T-Befehl (Test)

Der T-Befehl ist für alle Zusatzfunktionen identisch.

Durch wiederholte Eingabe von 'T' wird das DCC-Signal simuliert und abwechselnd auf 0 und 1 gesetzt.

Damit kann das angeschlossene Zubehör auf seine vorgesehene Funktion getestet werden.

Die Eingabe von < enter> beendet den Test und das Zubehörmenü wird wieder angezeigt.

Der Befehl E kann auch eingegeben werden, um direkt in den Betriebszustand zu gelangen.

```
Enter command for Accessory on A2/P14
M/I/R/T/A/D/N/E/enter/?: t
Press 'E' or 'Enter' to stop testing or 'T' for on/off
DCC value: 1
DCC value: 0
DCC value: 1
DCC value: 0

Enter command for Accessory on A2/P14
M/I/R/T/A/D/N/E/enter/?:
```

A-Befehl (Adresse)

Mit dem Befehl A wird die DCC-Adresse eines Ports eingestellt. Sollte diese Adresse bereits einem anderen Port zugeordnet sein, erhalten Sie eine Warnung. Wenn Sie das bestätigen, können Sie mit einer Adresse mehrere Servos/Zubehöre gleichzeitig steuern.

I-Befehl (Inversion)

Mit dem I-Befehl können Sie die Ausgabe eines Ports umkehren, wenn Sie dies wünschen.

Die folgende Tabelle zeigt, was Inversion pro Modus bedeutet

Mode 1		Single Steady	
	DCC	Port	
Not Inv	L	L	
	H	H	
Inv.	L	H	
	H	L	

Mode 2		Double Steady	
	DCC	Port	Buddy
Not Inv	L	L	H
	H	H	L
Inv.	L	H	L
	H	L	H

Mode 3		Single Flashing	
	DCC	Port on	Port off
Not Inv	L	L	L
	H	H: on time	L: off time
Inv.	L	L: off time	H: on time
	H	H	H

Mode 4		Double Flashing			
	DCC	Port on	Port off	Buddy on	Buddy off
Not Inv	L	L	L	L	L
	H	H: on time	L: off time	L: off time	H: on time
Inv.	L	L: off time	H: on time	H: on time	L: off time
	H	H	H	H	H

Mode 5		Single One Shot		
	DCC	Port (U)	Port (D)	Port (B)
Not Inv	L > H	H pulse	L	H pulse
	H > L	L	H pulse	H pulse
Inv.	L > H	L pulse	H pulse	L pulse
	H > L	H	L pulse	L pulse

Mode 6		Double One Shot	
	DCC	Port	Buddy
Not Inv	L > H	H pulse	L
	H > L	L	H pulse
Inv.	L > H	L pulse	H pulse
	H > L	H pulse	L pulse

Mode 7		Analog PWM	
	DCC	Port	
Not Inv	L > H	rising	
	H > L	falling	
Inv.	L > H	falling	
	H > L	rising	

Mode 8		Flickering PWM	
	DCC	Port	
Not Inv	L	L	
	H	flickering	
Inv.	L	flickering	
	H	H	

Mode 9		Random on/off		
	DCC	Fixed On	Fixed off	No Fixed
Not Inv	L	L	L	L
	H	Fix: H/Ra: L	Fix: L/Ra: H	Ra. H-L
Inv.	L	H	H	H
	H	Fix: L/Ra: H	Fix: H/Ra: L	Ra H-L

		Servo	
		DCC	Angle
Not Inv		L	Low
		H	High
Inv.		L	High
		H	Low

Frog port		
	Angle	1-Relay
Not Inv	Low	L
	High	H
Inv.	Low	H
	High	L

Frog port		
	Servo	2-Relay
Not Inv	Rotates	H
	Stopped	L
Inv.	Rotates	L
	Stopped	H

H = Hoog, High, Hoch, Haute = 5 Volt

L = Laag, Low, Niedrig, Bas = 0 Volt

Verschiedenes

Inbetriebnahme

Beim Start des Arduino blinkt die Status-LED dreimal kurz auf. Dies zeigt Ihnen an, dass der Arduino gestartet wird. Der Arduino wird auch neu gestartet, wenn Sie Putty im Konfigurationsstatus starten. Ab MARDEC 3.0 und höher gibt es keinen Neustart beim Umschalten zwischen Konfigurations- und Betriebsmodus.

Einrichtung von Servos

Gehen Sie wie folgt vor, um ein neues Servo einzurichten:

verbinden Sie das Servo mit dem Arduino und stellen Sie den Winkel mit dem 9-Befehl auf 90 Grad

montieren Sie nun das Servo unter der Anlage. Bringen Sie die Weichenzungen ungefähr in Mittelstellung zwischen den Gleisen

verwenden Sie den Befehl C, um die 1. Endlage zu erreichen

verwenden Sie hierbei die Befehle + und -, um den Winkel zu korrigieren

verwenden Sie nun den Befehl C, um die 2. Endlage zu erreichen

passen Sie diese mit + und - ebenfalls an

mit Herzstück-Polarisierung: überprüfen Sie die Verbindung zwischen Herzstück und Schiene. Differiert diese, dann invertieren Sie das Relais.

Überprüfen Sie die Displayanzeige Ihrer Zentrale bei dem entsprechenden DCC-Signal. Weicht diese von der realen Weichenstellung ab, dann invertieren Sie das Servo.

Schließen Sie mit <enter> ab, um die Einstellungen zu speichern.

Beenden

Das Bedienfeld (Putty) kann jederzeit geschlossen werden. Es gibt dafür keinen separaten Befehl.

Benutzen Sie also Alt-F4 oder klicken Sie auf das bekannte Kreuz oben rechts im Fenster.

Stellen Sie sicher, dass Sie sich im allgemeinen Befehlsmodus befinden, damit die letzten Änderungen für einen Port gespeichert sind.

Protokollierung

Der gesamte Konfigurationsdialog wird in folgender Datei protokolliert:

Meine Dokumente/Mardec/MARDEC_<Datum>_<Zeit>.log.

Es ist zu beachten, dass <Datum>_<Zeit> die Zeit des Schließens ist.

Dokumentieren

Wenn Sie die Konfiguration abgeschlossen haben, ist es ratsam, sämtliche Einstellungen in einem Dokument festzuhalten.

Dazu gibt es die Verknüpfung "Document Mardec" im Ordner MARDEC des Windows-Startmenüs.

Dadurch wird ein MS Excel-Dokument geöffnet, das Sie ausfüllen können. (Benötigt MS Excel auf Ihrem PC).

Arduino Nano und Mega2560

Um MARDEC auf einen **Arduino Nano** hochzuladen, müssen Sie die Arduino IDE verwenden. Der Standard-Bootloader ist zu groß, so dass nicht genügend Speicherplatz für MARDEC zur Verfügung steht, Sie können den Bootloader jedoch ersetzen.

Siehe Dokument "fixing the bootloader" auf www.arcomora.com/download/

Danach können Sie die Arduino IDE verwenden, um die .INO-Datei hochzuladen.

Um MARDEC auf einen **Arduino Mega2560** hochzuladen, verwenden Sie das 'Upload Tool' im Windows-Startmenü oder auf dem Desktop.

Für den Arduino Mega können Sie das DCC-Shield verwenden.

An den MEGA-Ports 11 bis 16 des Shields sind die Pins 54 bis 59 des MEGA angeschlossen.

MARDEC steuert dann die Pins 54 bis 59 auf einem MEGA so, als wären dies die Ports 11 bis 16 auf dem UNO.

So können Sie die Schraubklemmen 11 bis 16 einfach so verwenden, wie an einem UNO.

Konfigurieren weiterer Decoder

Nach der Installation der ArCoMoRa Software wird das Decoderprogramm beim ersten Start von MARDEC automatisch auf den angeschlossenen Arduino geladen.

Für alle weiteren Arduinos/DCCNext gibt es hierfür ein anderes Verfahren .

Starten Sie die Verknüpfung '**Upload Tool**' im Windows Startmenü oder auf dem Desktop.

Bei der Verwendung des Arduino MEGA2560 müssen Sie grundsätzlich diese Methode wählen, weil bei der Standardinstallation wird immer das Upload zu einem Arduino UNO durchgeführt wird.

Dieses 'Upload Tool' sowie eine Anleitung hierfür finden Sie im Windows-Startmenü im Arcomora-Ordner.

Anpassen des Com-Ports

Der Windows Com-Port kann sich beim Wiedereinschalten eines Arduino ändern.

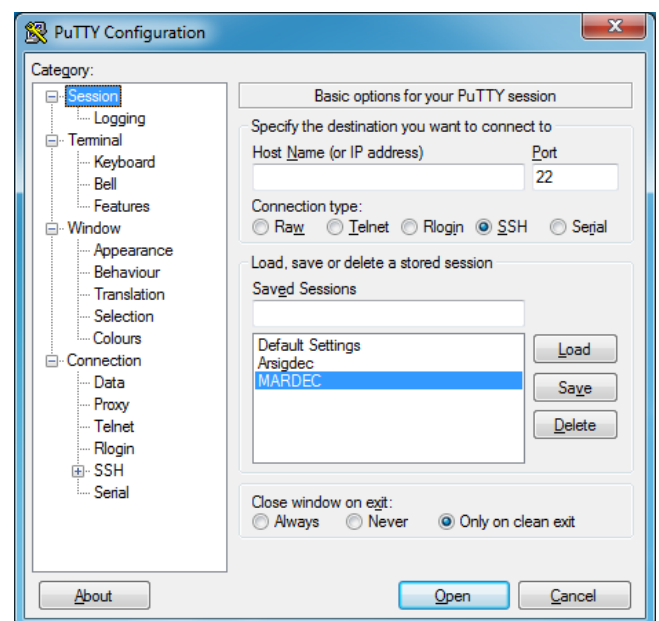
Mit dem Tool '**Change COM port**' können Sie den Com Port einfach ändern.

Diese finden Sie auch im Windows-Startmenü im ArCoMoRa Ordner.

Konfigurationsbildschirm für die Einstellungen (Putty)

Sie können das Putty Bedienfeld wie folgt einstellen:

1. Klicken Sie auf die Verknüpfung 'Configure Putty' im Windows-Startmenü im Arcomora-Ordner.
2. Die nebenstehende Anzeige erscheint:
3. Klicken Sie hier auf MARDEC und dann auf Load.
4. Die Sitzungsprotokollierung ermöglicht es Ihnen, die Protokollierungsoptionen anzupassen.
5. Hinweis: Wenn Sie den Dateinamen ändern, können die Konfigurationssitzungen nicht mehr gespeichert werden!
6. Mit Window Appearance/Behaviour/Colours können Sie auch das Aussehen ändern.
7. Bei Bedarf können Sie hier auch den COM-Port ändern.
8. Ändern Sie keine der anderen Einstellungen!
9. Wählen Sie Session und klicken Sie auf Save, um die Einstellungen erneut zu speichern.
10. Klicken Sie auf Öffnen, um die Systemsteuerung wieder zu öffnen.



Relaismodule

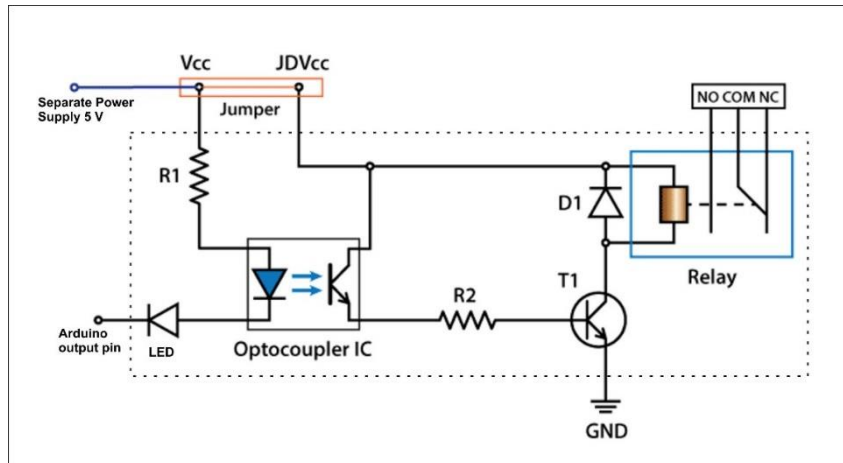
Wenn Sie die Relaismodule mit Optokoppler wie auf Seite 7 erwähnt verwenden, ist es wichtig zu wissen, dass diese 'LOW' aktiv sind.

Das bedeutet, dass das Relais mit Spannung versorgt wird, wenn die Spannung am Arduino-Anschluss 0 Volt beträgt.

Es gibt einen Standard-Jumper über die Ports Vcc und JDVcc.

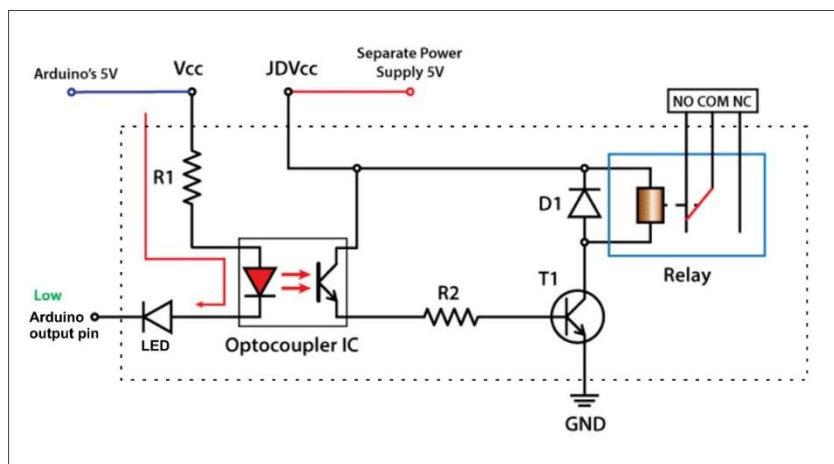
Sie haben nun zwei Möglichkeiten:

1) Lassen Sie den Jumper in seiner Position und schließen Sie den Vcc/JDVcc an eine externe Stromversorgung an.



2) Entfernen Sie den Jumper und:

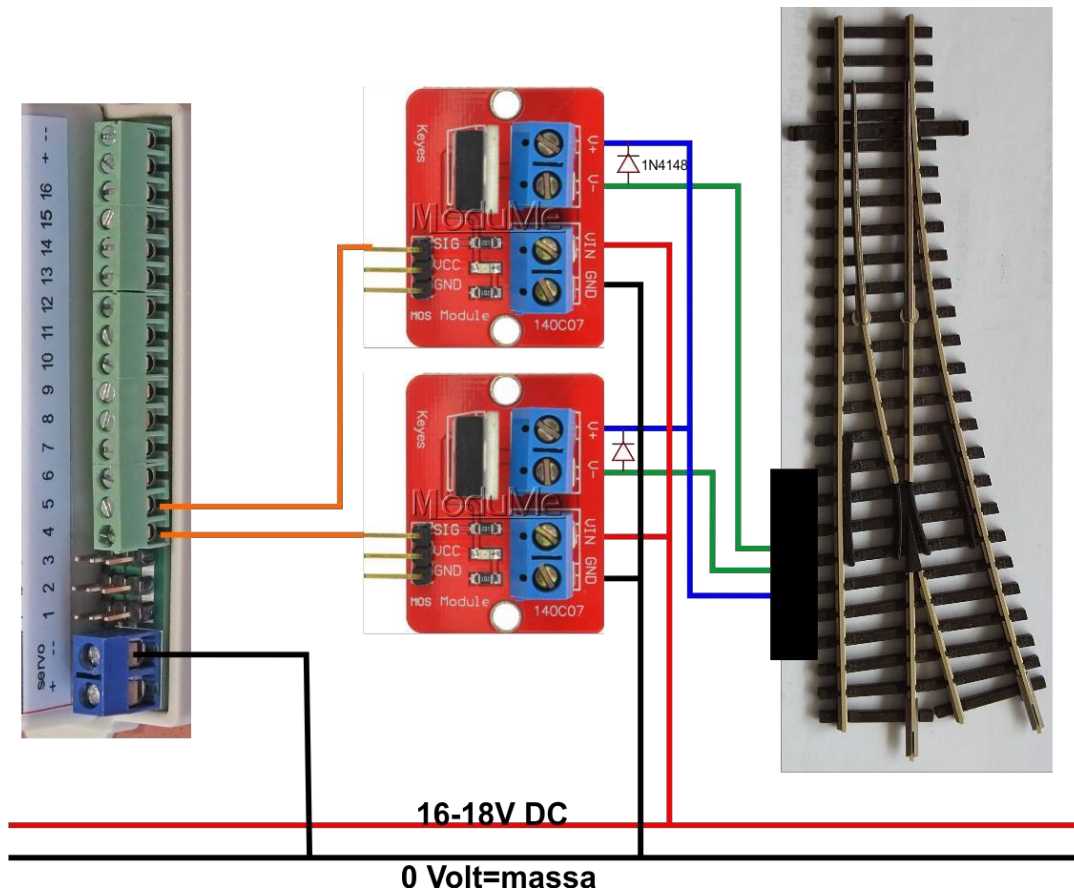
- schließen Sie Vcc an einen Arduino 5V Ausgang an
- schließen Sie JDVcc an eine externe Stromversorgung an



In diesem Bild sehen Sie, dass das Relais eingeschaltet ist, wenn der Arduino-Anschluss 'LOW' ist.

Lassen Sie auf keinen Fall den Jumper in Position und verbinden Sie Vcc/JDVcc mit dem 5V Pin des Arduino. In diesem Fall belastet auch der Spulenstrom den Arduino und das ist eine schlechte Lösung.

Anschluss eines MOSFET-Verstärkers für Weichen mit Spulenantrieben



Hier wird der DCCNext über das DCC-Shield mit der gleichen Spannung wie die MosFet Verstärker versorgt. (rote und schwarze Leitung)

Verbinden Sie den GND-Anschluss mit der Masse Ringleitung.

Hinweis: Die verwendeten Arduino-Ports hängen von der aktuellen Konfiguration ab.

Die hier gezeigten Anschlüsse sind nur ein Beispiel.

Konfiguration:

```
Enter command (P/A/D/E/I/R/?): P
Select port number.
Enter value from 1 to 16: 4
Port is undefined. First enter DCC address

Set DCC address for port 4
Enter value from 0 to 2000: 4
DCC Address set to 4

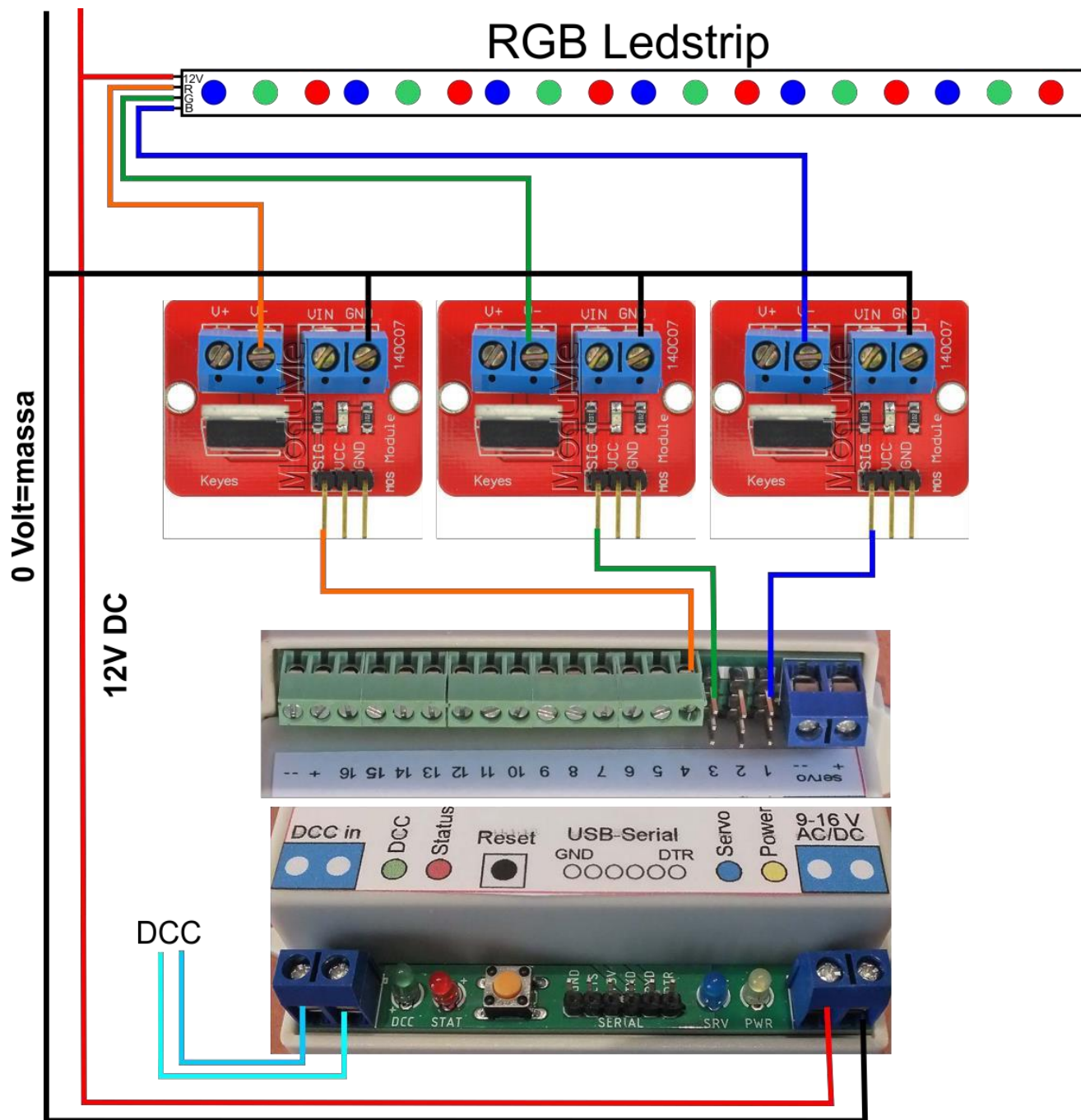
Select port type: Accessory(A), Servo(S) or Input(I): A
Port 4 set as Accessory port.

Select mode for this accessory
0=Help. Enter value from 0 to 9: 6
Mode set to Double One shot

Port 5 is assigned as buddy port to accessory on A4/P4
Enter 'On' time in milliseconds
Enter value from 5 to 30000: 250
'On' time set to 250 msec.

Enter command for Accessory on A4/P4
L/H/R/F/I/M/T/A/D/N/enter/?
Port settings are saved!
```

Verbindung von RGB-LED-Streifen mit MOSFET-Verstärkern

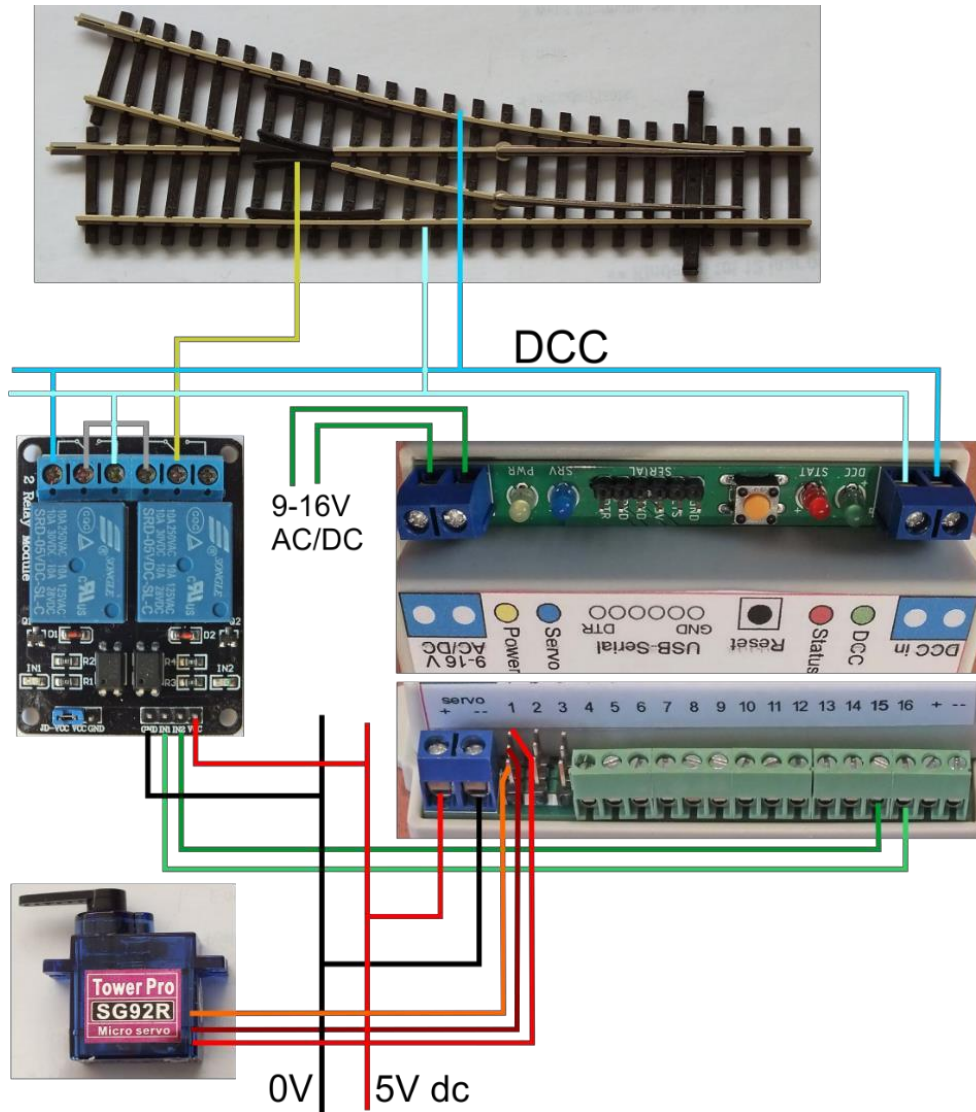


Hinweis: Die verwendeten DCCNext-Ports hängen von der aktuellen Konfiguration ab. Die hier gezeigten Anschlüsse sind nur ein Beispiel.

Konfiguration:

```
Port 1: DCC 12, Acc.type 7 (PWM), , Not Inv., PWM 5/150, Time R/F(sec) 800/800
Port 2: not used
Port 3: DCC 12, Acc.type 7 (PWM), , Not Inv., PWM 5/150, Time R/F(sec) 800/800
Port 4: DCC 12, Acc.type 7 (PWM), , Not Inv., PWM 5/150, Time R/F(sec) 800/800
Port 5: not used
```


Verbindung einer Weiche mit Doppelrelais und Servo

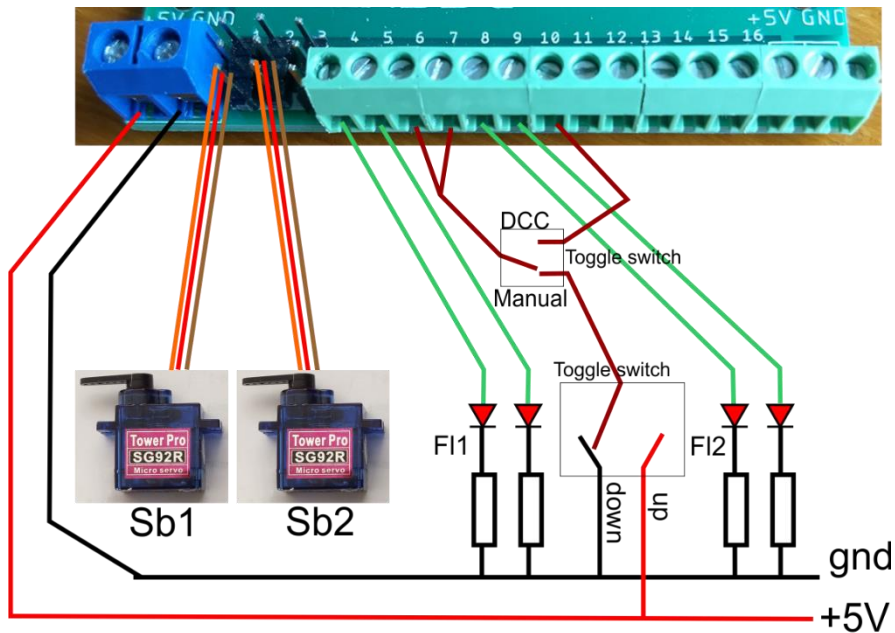


Hinweis: Die verwendeten Arduino-Ports hängen von der aktuellen Konfiguration ab.
Die hier gezeigten Anschlüsse sind nur ein Beispiel.

```
Enter command (P/A/D/E/I/R/?): p
Select port number.
Enter value from 1 to 16: 1
DCC 1, Servo Sb1, Angles 55/127, Not Inv., Speed 75, Frog port no, Bounce (L/H): 31/42
Angle set to HIGH: 127

Enter command for Servo on A1/P1
+/-/9/C/F/I/B/S/T/R/A/D/N/enter/?: F
How many relays? Enter value from 0 to 2: 2
Set Frog port. Enter value from 1 to 16: 15
Port 15 set as first frog port
Set second Frog port. Enter value from 1 to 16: 16
Port 16 set as second frog port
```

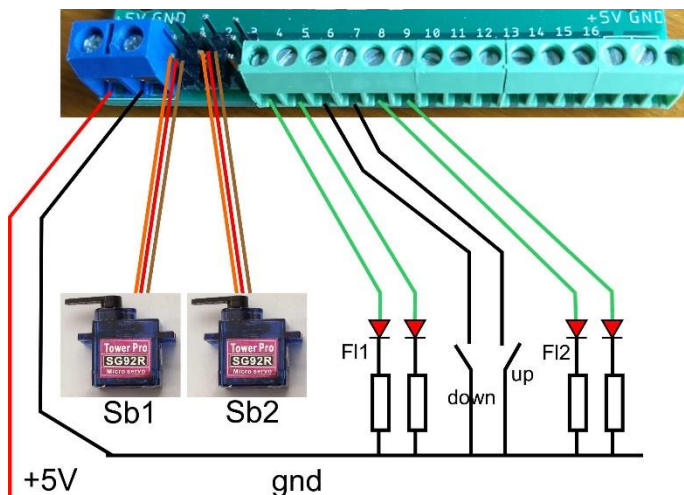
Bahnübergang



Railway crossing controlled by DCC (single steady on port 10)
or manual with two toggle switches.

```

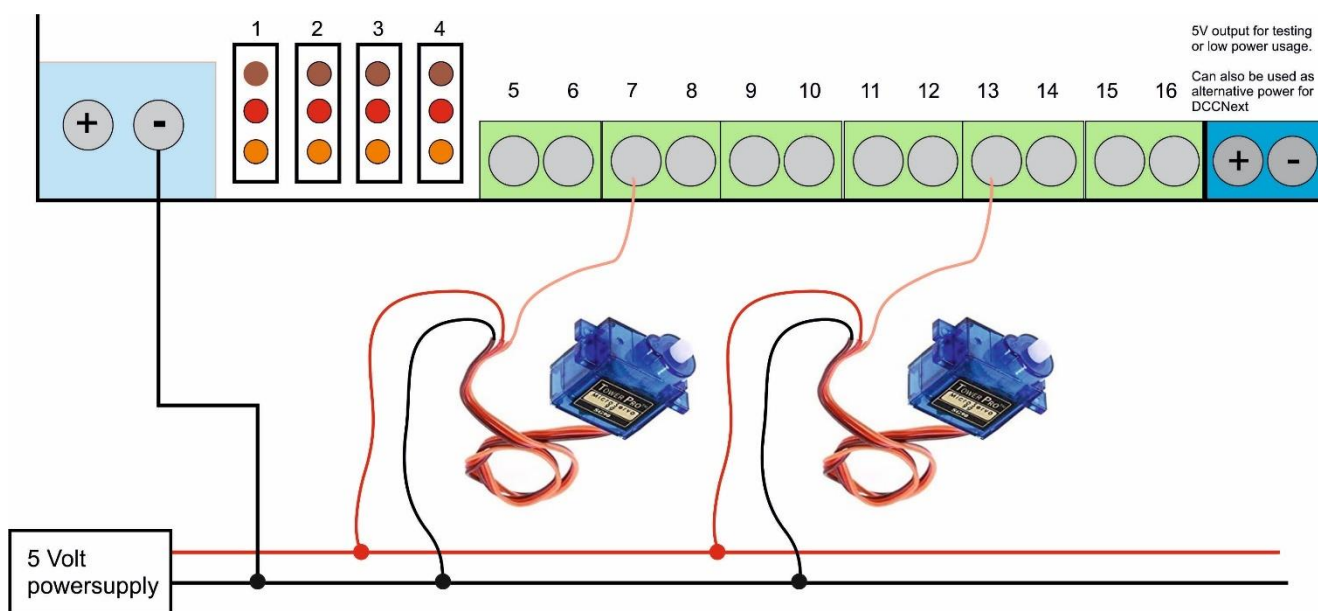
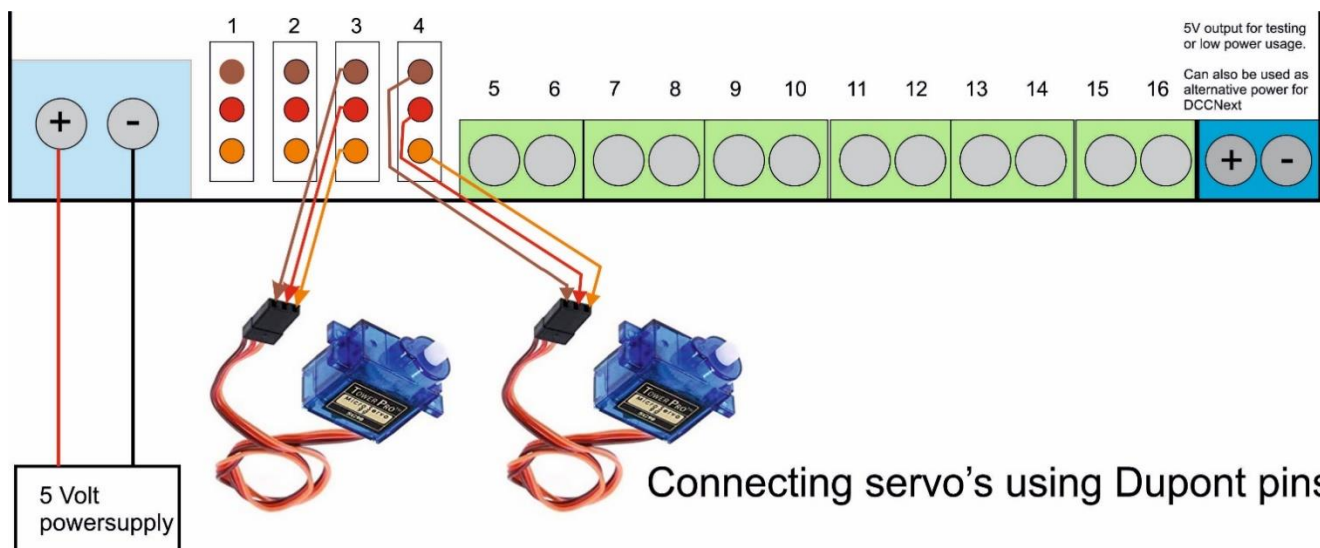
Port 1: DCC 1, Servo Sb1, Angles 55/127, Not Inv., Speed 75, Frog port no, Bounce (L/H): 31/42
Port 2: DCC 1, Servo Sb2, Angles 53/131, Not Inv., Speed 75, Frog port no, Bounce (L/H): 42/31
Port 3: not configured
Port 4: DCC 4, Acc.type 4 (D. Flashing), FI1, Not Inv., Buddy 5, Time(ms) 300/300
Port 5: Buddy for port 4
Port 6: DCC 4, Input down, Trigger: D, Second addr/delay: 1/3.0 sec, Third addr/delay: none
Port 7: DCC 1, Input up, Trigger: U, Second addr/delay: 4/7.0 sec, Third addr/delay: none
Port 8: DCC 4, Acc.type 4 (D. Flashing), FI2, Not Inv., Buddy 9, Time(ms) 300/300
Port 9: Buddy for port 8
Port 10: DCC 2, Acc.type 1 (S. Steady), , Not Inv.
Port 11: not configured
Port 12: not configured
    
```



Railway crossing controlled manual with 2 switches. No DCC required

```

Port 1: DCC 1, Servo Sb1, Angles 55/127, Not Inv., Speed 75, Frog port no, Bounce (L/H): 31/42
Port 2: DCC 1, Servo Sb2, Angles 53/131, Not Inv., Speed 75, Frog port no, Bounce (L/H): 42/31
Port 3: not configured
Port 4: DCC 4, Acc.type 4 (D. Flashing), FI1, Not Inv., Buddy 5, Time(ms) 300/300
Port 5: Buddy for port 4
Port 6: DCC 4, Input down, Trigger: D, Second addr/delay: 1/3.0 sec, Third addr/delay: none
Port 7: DCC 1, Input up, Trigger: D, Second addr/delay: 4/7.0 sec, Third addr/delay: none
Port 8: DCC 4, Acc.type 4 (D. Flashing), FI2, Not Inv., Buddy 9, Time(ms) 300/300
Port 9: Buddy for port 8
Port 10: not configured
Port 11: not configured
Port 12: not configured
    
```



Version 6 der Versionshinweise

Neu:

- Das 'Nachwippen' der Servos am Ende der Drehung. Das Servo kann bis zu 4 Mal mit Winkeln von 1 bis 9 Grad wippen. Dies kann für beide Winkel unterschiedlich sein.
 - Die Testoption für Zubehör wurde geändert. Anstatt das Zubehör mehrmals zu aktivieren, kann das Zubehör mit der T-Taste ausgeschaltet werden. Dies simuliert das DCC-Signal.
 - Es können wieder Buddy und Point Ports ausgewählt werden.
 - Die Menüstruktur für den Wechsel von Zubehör wurde vereinfacht. Hinzugekommen ist der Befehl E, der eine Kombination aus <enter> (=speichern) und dem allgemeinen Befehl E (=Betriebsmodus starten) ist.
 - Der "Single Shot" kann auch durch einen Down und/oder Up Puls aktiviert werden.
 - Die beiden bisherigen Zufallsfunktionen werden zu einer zusammengefassten Zufallsfunktion mit Ein- und Ausschaltzeiten von 20ms. bis 600 sec. zusammengeführt.
- Es ist auch möglich, eine feste Ein-/Ausschaltzeit festzulegen.
- Ein Eingang hat jetzt auch eine Invertierungsoption.
 - Ein einzelner Port kann nun auch mit dem Befehl R zurückgesetzt werden. Es ist also nicht mehr notwendig, die Adresse auf 0 zu setzen.
 - Zurück zum Konfigurationszustand kann nun auch mit dem P-Befehl erfolgen.
 - Der Schnellkonfigurationsmodus. So können Sie schnell eine Änderung für einen Port vornehmen.
- Mit 'P' wird die Konfiguration gestartet und die Portnummer sofort abgefragt.
- Nach der Änderung können Sie die Änderungen sofort mit 'E' speichern und in den Betriebsmodus zurückkehren.

Fehlerbehebung

Die Servos können auch an den Anschlüssen 12 bis 16 verwendet werden.

Neu in Version 7

In dieser Version wurde die DCC_library von Mynabay durch die NMRA-Bibliothek ersetzt.

Mit Dank an Karst Drenth (Drenth Design & Consultancy)

Durch diese Änderung wurde das Problem mit Railcom gelöst