

DCC-EX v 5.0.9

03a - Erweiterungen - MEGA 2560



Tests wurden unter Windows 11 durchgeführt

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

Wichtig	2
Eingänge / Ausgänge des Arduino MEGA 2560	3
Händische Einrichtung	4
Ausgang z.B. an PIN 22 = Vpin 22 des MEGA's	5
Eingang z.B. an PIN 24 = Vpin 24 des MEGA's	7
Einrichtung in der Datei „mySetup.h“	9

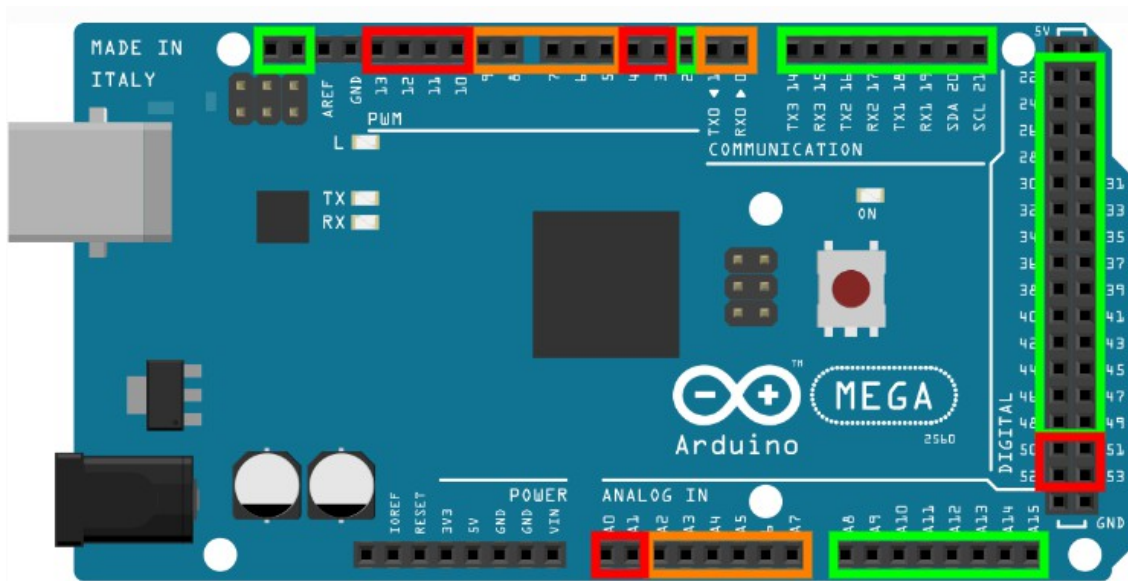
Wichtig

- **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.



Eingänge / Ausgänge des Arduino MEGA 2560



nicht zu benutzende Pin's

Pin's	Details
A0 - A1	Analoge I/O-Pin's für die Strommessung auf der Haupt- und Programmspur
3 - 4	Digitale I/O-Pin's, die häufig von Motor-Shields oder Ethernet-Shields verwendet werden
10 - 13	Digitale I/O-Pin's, die häufig von Motor-Shields verwendet werden
50 - 53	Digitale I/O-Pin's, die für spezielle Systemdienstprogramme reserviert sind

Pin's, die mit Vorsicht zu verwenden sind

Pin's	Details
A2 - A7	Analoge I/O-Pin's sind standardmäßig verfügbar, können aber für zusätzliche Strommessung verwendet werden
Tx0/Rx0	Kann für WiFi- oder Bluetooth-Adapter verwendet werden, koexistiert jedoch mit Onboard-USB und muss für Software-Uploads getrennt werden
5 - 9	Digitale I/O-Pin's sind standardmäßig verfügbar, aber einige Motor-Shields oder andere Peripheriegeräte verwenden diese

zu verwendende Pin's

Pin's	Details
SDA/SCL (20/21)	I2C - Anschluss zur Verwendung mit I2C - Displays, I/O-Expander usw., nicht für allgemeine I/O
SDA/SCL (in der Nähe von USB)	I2C - Anschluss zur Verwendung mit I2C - Displays, I/O-Expander usw., nicht für allgemeine I/O
Tx1/Rx1/Tx2/Rx2/Tx3/Rx3	Zusätzliche serielle Schnittstellen, nicht für allgemeine I/O
A8 - A15	Analoge I/O-Pin's zur freien Verwendung
2	Digitaler I/O-Pin's zur freien Verwendung
22 - 49	Digitaler I/O-Pin's zur freien Verwendung

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 dann alles einwandfrei.

Der bessere Weg ist jedoch, die Einstellungen in der Datei

„myAutomation.h“

und

„mySetup.h“

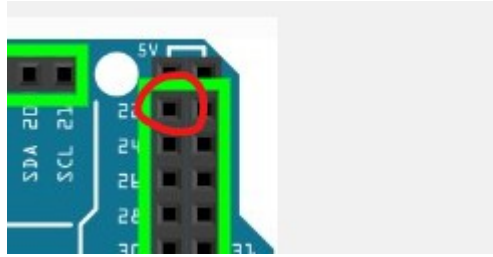
vorzunehmen.

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.

Jetzt aber zunächst zur manuellen Installation auf der nächsten Seite:

Ausgang z.B. an PIN 22 = Vpin 22 des MEGA's



- Befehl: **<Z id Vpin iflag>**

- **id** = 0 - 32767 (DCC - Adresse)
- **Vpin** = siehe „zu verwendende Pin's“
- **iflag** =

Bit 0	0 =	forward operation (ACTIVE=HIGH / INACTIVE=LOW)
	1 =	inverted operation (ACTIVE=LOW / INACTIVE=HIGH)
Bit 1	0 =	state of pin restored on power-up to either ACTIVE or INACTIVE depending on state before power-down.
	1 =	state of pin set on power-up, or when first created, to either ACTIVE or INACTIVE depending on IFLAG, bit 2
Bit 2	0 =	state of pin set to INACTIVE upon power-up or when first created
	1 =	state of pin set to ACTIVE upon power-up or when first created

- **<Z 22 22 0>** (Hier für die DCC Adresse 22 an Vpin 22
(die Adresse kann zwischen 0 - 32767 liegen))

Rückmeldung: **<O>** = ok - **<X>** = Fehler aufgetreten

- **<Z 22 1>** Led „EIN“
Rückmeldung: **<Y 22 1>**

- **<Z 22 0>** Led „AUS“
Rückmeldung: **<Y 22 0>**

- **<Z>** Listet alle definierten Ausgangs Pins auf
Rückmeldung: z.B. **<Z 22 22 0 1>**
für jeden definierten Ausgang **<Z id Vpin iflag state>**
wenn kein Ausgang definiert ist **<X>**

Einstellung bei RailControl, Nutzung des Ausgangs als Zubehörartikel		<div> Led an Vpin 22 </div> <div> <div>Basisdaten</div> <div>Position</div> </div> <div> Name: <input type="text" value="Led an Vpin 22"/> </div> <div> Adresse: <input type="text" value="22"/> <input type="button" value="-"/> <input type="button" value="+"/> </div> <div> Schaltzeit (ms): <input type="text" value="250"/> </div> <div> Invertiert: <input type="checkbox"/> </div> <div> Typ: <input type="text" value="standard"/> </div>
---	--	---

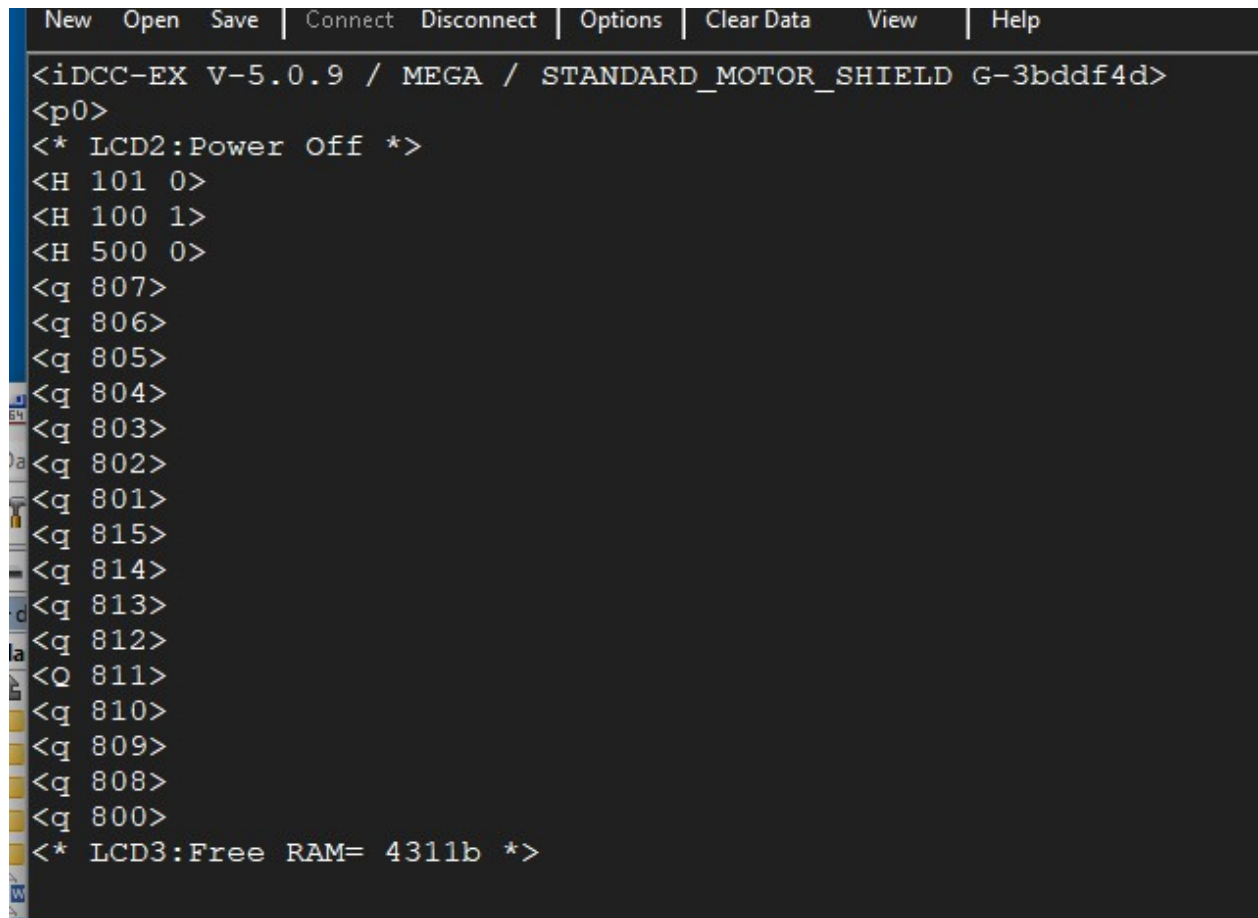
- **<Z id>** Löscht die Definition der Ausgabe-ID
Rückmeldung: **<0>** = ok - **<X>** = Fehler aufgetreten

Wenn der Arduino als solcher gesteuert wird, aktualisiert und speichert er den Zustand jedes Ausganges im EEPROM, so dass alle Zustände auch ohne Strom erhalten bleiben.

Eine Liste der aktuellen Zustände jedes Ausgabepins wird von der **EX-Command-Station** generiert, wenn der Statusbefehl aufgerufen wird. Dies bietet eine effiziente Möglichkeit, den Zustand aller Ausgänge zu initialisieren, die von einer separaten Schnittstelle oder einem GUI-Programm überwacht oder gesteuert werden.

- <s>

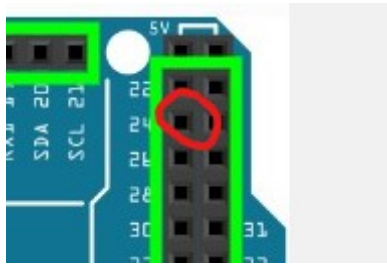
Rückmeldung: bei mir



The screenshot shows the EX-Command-Station software interface. The menu bar includes New, Open, Save, Connect, Disconnect, Options, Clear Data, View, and Help. The main window displays the following text:

```
<iDCC-EX V-5.0.9 / MEGA / STANDARD_MOTOR_SHIELD G-3bddf4d>
<p0>
<* LCD2:Power Off *>
<H 101 0>
<H 100 1>
<H 500 0>
<q 807>
<q 806>
<q 805>
<q 804>
<q 803>
<q 802>
<q 801>
<q 815>
<q 814>
<q 813>
<q 812>
<Q 811>
<q 810>
<q 809>
<q 808>
<q 800>
<* LCD3:Free RAM= 4311b *>
```


Eingang z.B. an PIN 24 = Vpin 24 des MEGA's



- **Befehl: <S id Vpin pullup>**
 - **id** = 0 - 32767 (DCC - Adresse)
 - **Vpin** = siehe „zu verwendende Pin’s“
 - **pullup** =

1	Interner Pull-up-Widerstand für PIN verwenden (ACTIVE = LOW)
0	interner Pull-up-Widerstand für PIN nicht verwenden (ACTIVE = HIGH).

Erstellt einen neuen Eingang mit angegebener PIN und PULLUP, wenn der Eingang bereits vorhanden ist, wird er mit der angegebenen PIN und PULLUP aktualisiert.

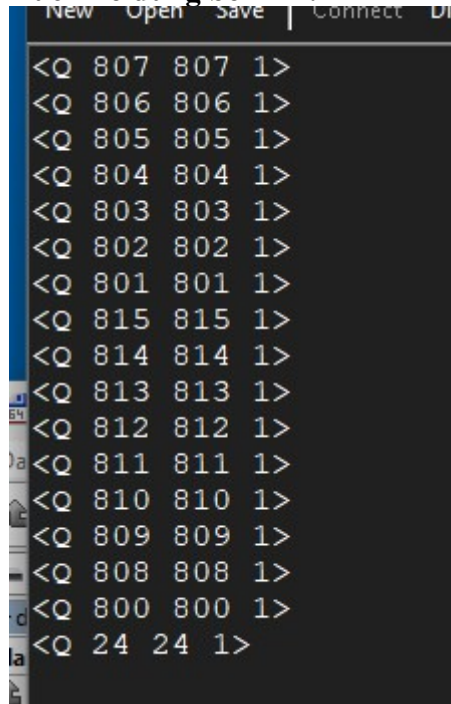
- **<S 24 24 0>** Hier für die DCC Adresse 24 an Vpin 24
(die Adresse kann zwischen 0 - 32767 liegen)

Rückmeldung: $\langle 0 \rangle = \text{ok}$ - $\langle X \rangle = \text{Fehler aufgetreten}$

<p>Einstellung bei RailControl, Nutzung als Rückmelder</p>	
<p>Taster = „gedrückt“</p>	
<p>Taster = „nicht gedrückt“</p>	

- **<S> Listet alle definierten Eingänge auf**

Rückmeldung bei mir:



```

New  Open  Save  Connect  Dis
<Q 807 807 1>
<Q 806 806 1>
<Q 805 805 1>
<Q 804 804 1>
<Q 803 803 1>
<Q 802 802 1>
<Q 801 801 1>
<Q 815 815 1>
<Q 814 814 1>
<Q 813 813 1>
<Q 812 812 1>
<Q 811 811 1>
<Q 810 810 1>
<Q 809 809 1>
<Q 808 808 1>
<Q 800 800 1>
<Q 24 24 1>

```

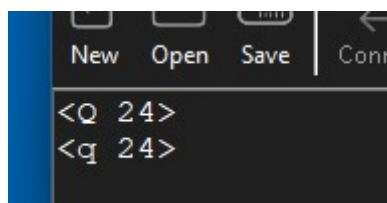
- **<S id> Löscht die Definition des Eingangs**

Nachdem alle Eingänge ordnungsgemäß definiert wurden, verwendet man den Befehl **<E>** (Großbuchstabe E), um die zuvor gemachten Definitionen dauerhaft im EEPROM zu speichern.

Wenn später Änderungen/ Ergänzungen/ Löschungen an den Eingängen vorgenommen werden, muß der Befehl **<E>** in Großbuchstaben abermals aufgerufen werden, um diese neuen Definitionen im EEPROM dauerhaft zu aktualisieren.

Es können auch alle im EEPROM gespeicherten Einstellungen (zu Weichen, Sensoren und Ausgängen) gelöscht werden, indem der Befehl **<e>** (Kleinbuchstabe e) aufgerufen wird. (Es gibt dann KEIN zurück mehr)

In einem Terminalprogramm kann man den Zustand des Eingangs testen.



```

New  Open  Save  Conn
<Q 24>
<q 24>

```

<Q id>	für den Übergang des Eingangs vom INACTIVE - Zustand in den ACTIVE - Zustand (d.h. der Eingang wurde ausgelöst)
<q id>	für den Übergang des Eingangs vom ACTIVE - Zustand in den INACTIVE - Zustand (d.h. der Eingang wird nicht mehr ausgelöst)

<Q> = Listet den Status aller definierten Sensoren auf

Einrichtung in der Datei „mySetup.h“

Weitere Beschreibung siehe unter:

- **Datei:** 03d - Erweiterungen - NANO
- **Überschrift:** Einrichtung in der Datei „myAutomation.h“ / „mySetup.h“

1. Eingänge

- **Absatz:** Befehle für Eingänge

```
85 // Vpin 855 ist kein Digital Anschluss, kann somit nicht benutzt  
86  
87 // MEGA - Eingänge  
88 SETUP("<S 24 24 1>"); // Vpin 24 = PIN 24 des MEGA  
89  
90 // MEGA - Ausgänge
```

Auszug aus meiner Datei: mySetup.h - Eingang auf MEGA PIN 24

1. Ausgänge:

- **Absatz:** Befehle für Ausgänge

```
SETUP("<S 24 24 1>"); // Vpin 24 = PIN 24 des MEGA  
  
// MEGA - Ausgänge  
SETUP("<Z 22 22 0>"); // Vpin 22 = PIN 22 des MEGA
```

Auszug aus meiner Datei: mySetup.h - Ausgang an MEGA PIN 22