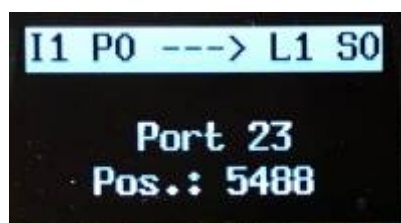


# Drehscheibenbewegungen

Man kann über das #define **PARTLY\_USED\_PORTS** auswählen, ob alle Ports anfahrbar sind oder nur die aktiven und passiven, also die mit Gleisanschlüssen bzw. deren Gegenüber.

Standardmäßig - außer bei DCC-Befehlen - nimmt die DS immer den kürzesten Weg zum nächsten anzufahrenden Port (**DIRECTION\_CHANGE\_POSSIBLE** = 0). Am Beispiel der Uhr: aktueller Port ist 3 Uhr, wir wollen auf 10 Uhr fahren. Der kürzere Weg ist entgegen dem Uhrzeigersinn. Die Drehscheibe wird rückwärts gedreht.

Im Hauptbildschirm ist in der Mitte der aktuelle Port ersichtlich und darunter die Position des StepperMotors:



Die oberste Zeile ist eine Statuszeile, sie stellt dar:

- **Initialisierungsstatus I** ja/nein (1/0), zeigt, ob mindestens in einer Richtung die Positionen finegetuned wurden.
- **Polarisierungsrelais P** on/off (1/0)
- **DCC-Richtung** <--- --> (nur wenn DCC verwendet wird, d.h. #define USE\_DCC = 1)
- **Licht L** on/off (1/0)
- **Sound S** on/off (1/0)

Drehscheibenbewegungen können über mehrere Wege ausgelöst werden:

## Drehencoder

- Es sind 2 unterschiedliche Steuerungslogiken für den Drehencoder definiert, die über **ENCODER\_LOGIC** eingestellt werden:
  - **ENCODER\_LOGIC** = 1 Drehencoder-Drehen während des Drehscheiben-Stillstands ermöglicht Portauswahl, Bewegung wird erst gestartet, wenn Encoder-Taste gedrückt wird, während der Bewegung kann die anzufahrende Portnummer nicht mehr geändert werden, Taste während Bewegung stoppt die Bewegung
  - **ENCODER\_LOGIC** = 0 Encoder-Drehen startet sofort die Bewegung, während einer Bewegung kann weiter am Encoder gedreht und ein anderer Port selektiert werden. Drücken der Encoder-Taste während einer Bewegung stoppt die Drehscheibe.
- Beim Mehrfach-Drehen des Drehencoders behält die DS die einmal eingeschlagene Richtung bei, wenn **DIRECTION\_CHANGE\_POSSIBLE** = 0 ist, auch wenn es andersherum kürzer wäre. Dies verhindert abrupten Richtungswechsel in der Bewegungsphase.
- Bei **DIRECTION\_CHANGE\_POSSIBLE** = 1 ist ein Richtungswechsel während einer Bewegung erlaubt.

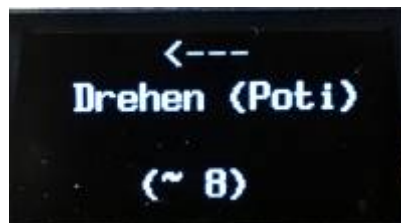
- Bei Notaus (Drücken des Drehencoders während einer Bewegung) erscheint die Portnummer im OLED-Display mit einer vorangestellten Tilde „~“, d.h. die DS steht *ungefähr* in der Nähe von Port x.
- Während einer Bühnen-Fahrt wird in der untersten Zeile angezeigt, wo sich die Hausseite der Bühne gerade *ungefähr* befindet.



## Poti

Das Potentiometer oder kurz Poti rastet in der Mittelstellung ein.

- Dreht man es nach rechts, bewegt sich die Drehscheibe im Uhrzeigersinn, dreht man es nach links, bewegt sie sich gegen den Uhrzeigersinn.
- Je weiter man das Poti aufdreht, desto schneller dreht sich die Bühne. Der aktuelle Port wird in der untersten Zeile dargestellt.



- Zurück in Mittelstellung stoppt sie wieder. Die Portnummer erscheint im OLED-Display mit einer vorangestellten Tilde „~“, d.h. die DS steht *ungefähr* in der Nähe von Port x.



Falls kein Poti verwendet wird und keines angeschlossen ist, muss dass `#define USE_POTI` auf 0 gesetzt werden, sonst kann es sein, dass die Drehscheibe unkontrolliert herumschleicht.

## DCC

Bei Ansteuerung der Drehscheibe über DCC entscheidet die DCC-Richtung über die Drehrichtung der Bühne.

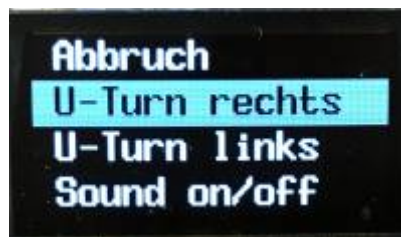
Von der Zentrale bzw. einer entsprechenden PC-Modellbahnsteuerungssoftware (z.B. Traincontroller)

wird zunächst die Richtung als DCC-Kommando gesendet und anschließend die Portposition bzw. Durchführung eines U-Turns. Die DCC-Richtung wird in der oberen Statuszeile angezeigt (Pfeil) und gespeichert. Sendet die Zentrale nur einen Fahrbefehl, wird die gespeicherte Richtung verwendet.

Weitere Infos über DCC-Anbindung sind hier zu finden: [LocoTurn-Ansteuerung über DCC und PC-Steuerungsprogramme](#)

## Menü

Über das Menü ([Menüsteuerung](#)) kann eine 180°-Drehung CW / U-Turn rechts oder CCW / U-Turn links ausgelöst werden.



Weitere Bewegungen über das Menü sind nicht möglich.

## Serieller Monitor

Im seriellen Monitor der Arduino-IDE können diverse Drehscheibenbewegungen ausgelöst werden: [Kommandos für den seriellen Monitor](#)

Der serielle Monitor in der Arduino-IDE muss auf 9600 Baud eingestellt sein.



Im echten Betrieb auf der Anlage sollten alle Ein- und Ausgaben über den seriellen Monitor abgeschaltet werden; es ist schließlich i.d.R. kein PC angeschlossen. Für Testzwecke und beim Einrichten der Drehscheibe ist der serielle Monitor jedoch sehr hilfreich, insbesondere für Debugging.

## Typischer Bewegungsablauf

Der Bewegungsablauf (mit eingeschaltetem Sound) ist folgendermaßen:

1. Warnleuchte und Sound (Hupe) starten
2. Signale blenden auf rot über, bzw. werden bei MLL-Betrieb auf rot gestellt
3. Motor läuft an, ein entsprechendes Anlaufgeräusch ertönt
4. Drehscheibe beschleunigt sanft und startet die Drehung, das typische Fahrgeräusch wird abgespielt
5. Während der Drehung wird am OLED der ungefähre Port angezeigt, wo sich die Bühne gerade

befindet

6. Am Zielport bremst die Bühne ab und bleibt stehen
7. Der Bewegungs-Sound stoppt
8. Die Signale überblenden entsprechend der Bühnenstellung auf weiß bzw. rot und die Warnleuchte wird ausgeschaltet
9. Es ertönt ein erneutes Hupen

From:  
<https://wiki.mobaledlib.de/> - **MobaLedLib Wiki**

Permanent link:  
[https://wiki.mobaledlib.de/anleitungen/bauanleitungen/locoturn\\_v10/150\\_locoturn\\_bewegungen?rev=1679638559](https://wiki.mobaledlib.de/anleitungen/bauanleitungen/locoturn_v10/150_locoturn_bewegungen?rev=1679638559)

Last update: **2023/03/24 07:16**

