

# Grundlegende Philosophie von LocoTurn

## Einstellungen der Drehscheiben-Parameter

Der Sketch umfasst zahlreiche Einstellparameter, um die Drehscheiben-Funktionen konfigurieren zu können. Diese Parameter werden meist über sogenannte `#defines` festgelegt. Die `#defines` sind nichts anderes als eine Art Textersetzung und werden während des Kompilervorgangs berücksichtigt. `#define LICHT 1` bedeutet nichts anderes, als dass überall, wo LICHT im Sketch steht, eine „1“ eingebaut wird. `#defines` sind immer GROSS geschrieben. Zwischen dem Parameternamen und dem Wert kommt kein „=“-Zeichen (auch wenn dies hier im Manual zur Verbesserung der Lesbarkeit manchmal anders steht)!

Die wesentlichen benutzerindividuellen Parameter stehen in der `Turntable_Config.h` Datei bzw. im entsprechenden Reiter der Arduino-IDE. Sie beschreiben die jeweilige individuelle Drehscheibe und die verwendeten Steuerungseinstellungen. Im ino-File stehen weitere `#defines`. Generell gibt es `#defines`, die echte Benutzer-Parameter sind, um den Sketch zu beeinflussen, und Sketch-interne `defines`, die eher zur Vereinfachung des Programmablaufs dienen.

Im Hauptsketch (ino) stehen zudem die default-Werte für diese Benutzerparameter (falls mal einer in der `Turntable_Config.h` vergessen wurde) und die Sketch-internen `defines`!

Bei einer neuen Programmversion kopiert man einfach das „`Turntable.ino`“-File in das Verzeichnis „`Turntable`“ (oder als Text in den entsprechenden Reiter). Die Konfiguration im „`Turntable_Config`“ bitte nicht übernehmen, sonst überschreibt ihr eure Drehscheibeneinstellungen! Ausnahme: neue `#defines` für neue Funktionen müssen ggf. in der „`Turntable_Config.h`“ eingebaut werden.

Die meisten Parametereinstellungen müssen direkt im Programm bzw. im Config-Reiter gemacht werden, eine spätere Einstellungen im lfd. Betrieb (z.B. über Dekoder-CVs, wie man das von anderen Dekodern kennt) ist i.d.R. nicht möglich. Für Änderungen der Parameter muss man diese im Sketch anpassen und den Code erneut in den Arduino Nano laden.

Der Grund, warum keine CVs verwendet werden konnten, ist schlicht und ergreifend der begrenzte Speicherplatz eines Nano.

## Konfigurationsdatei

Der Arduino-Sketch beinhaltet das eigentliche Programm (`Turntable.ino`-File), eine Konfigurationsdatei (eigener Reiter „`Turntable_Config.h`“) sowie mehrere modifizierte Bibliotheken (Libraries).

```
Turntable Diagnose.cpp Diagnose.h Dprintf.h NmraDcc.cpp NmraDcc.h Smart_JQ6500_Serial.h Turntable_Config.h
1 //-----
2 //
3 //          LocoTurn
4 //          Turn table control
5 //          Version V1.0
6 //          by Hardi & domapi
7 //          22.01.2023
8 //-----
9
```

Bei einer neuen Programmversion, d.h. einer neuen „Turntable.ino“, kann man die eigene Konfiguration übernehmen. Diese muss nur in Ausnahmefällen angepasst werden.

Bevor die Steuerung verwendet werden kann, muss man zunächst in der Turntable\_Config.h-Datei die eigene Drehscheibe definieren (Anzahl und Lage der Ports, Drehgeschwindigkeit, Bühnengleispolarisierung etc.) und das Verhalten der verschiedenen Steuerungsmöglichkeiten einstellen (Beleuchtungseffekte, Sound-Lautstärke etc.).

Ein Beispiel zur Konfiguration einer Fleischmann-Drehscheibe und einem 400 Step-Stepper-Motor findet sich hier am Ende der Seite: [Parameter zur Einstellung der Drehscheibe](#)

## Definition von Ports/Gleisabgängen

- Eine Drehscheibe (DS) hat eine bestimmte Anzahl von Gleisabgängen, im Sketch Ports genannt. Diese sind über ein #define PORT\_CNT festgelegt.
- Es gibt 3 Arten von Ports:
  - **aktive** Ports können tatsächlich zum Befahren der Drehscheibe genutzt werden, Loks können die Bühne befahren und wieder abfahren; d.h. es ist ein Gleis angeschlossen
  - **passive (blinde)** Ports, an denen ein kurzes Gleisstück angeschlossen ist (vgl. typische Fleischmann-Drehscheibe), das aber keine Verbindung zur Anlage hat
  - **inaktive** Ports, an denen kein Gleis angeschlossen ist, auch kein kurzes Stück wie z.B. bei Fleischmann-DS üblich
- Bei den meisten Drehscheiben liegen die Gleisabgänge in einem bestimmten Raster und sind gleichverteilt über einen kompletten Vollkreis. Bei einer klassischen Fleischmann-Drehscheibe gibt es z.B. im 7,5°-Raster Gleisabgänge, also in Summe 48 Ports ( $360^\circ / 7,5^\circ$ ). Eine kleine Fleischmann Drehscheibe hat z.B. 24 Ports im 15°-Raster.
- Bei der Fleischmann-DS liegt gegenüber einem Auffahrgleis entweder ein Gleisabgang (= aktiver Port) oder ein passiver, „blinder“ Port (ist mit kurzen Schienenstücken versehen, man kann aber kein Gleis anschließen und folglich nicht auf-/abfahren). Von daher liegen benutzbare / anfahrbare Ports immer gegenüber. Das muss so sein, damit man U-Turns machen kann.
- Es gibt immer eine geradzahlige Anzahl Ports.
- Die kleinste Anzahl Ports ist 2: Ein Gleisabgang plus der gegenüberliegende Port; damit kann man dann aber wirklich nur eine Lok wenden 😬.



muss man ggf. Programmteile ausschalten, damit zusätzliche DEBUG-print-Ausgaben aktiviert werden können!

- Einrichten der Steuerung + Test
  1. DCC abschalten, spart Speicher
  2. seriellen Monitor einschalten (Ausgabe + Eingabe)
- Einsatz auf der Anlage
  1. DCC einschalten
  2. keine Verwendung des seriellen Monitors, da sinnlos (es ist i.d.R. kein PC angeschlossen)
- Einsatz auf der Anlage ohne Poti; Steuerung nur mit Encoder und DCC
- Einsatz auf der Anlage ohne Poti und ohne Encoder; Steuerung nur mit DCC

Die Einsatzszenarien lassen sich über die #defines in der Turntable\_Config.h einstellen: [Parameter zur Einstellung der Drehscheibe](#)

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

Permanent link:  
[https://wiki.mobaledlib.de/anleitungen/bauanleitungen/locoturn\\_v10/150\\_locoturn\\_philosophie?rev=1727942195](https://wiki.mobaledlib.de/anleitungen/bauanleitungen/locoturn_v10/150_locoturn_philosophie?rev=1727942195)

Last update: **2024/10/03 07:56**

