# 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 Kompiliervorgangs 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!

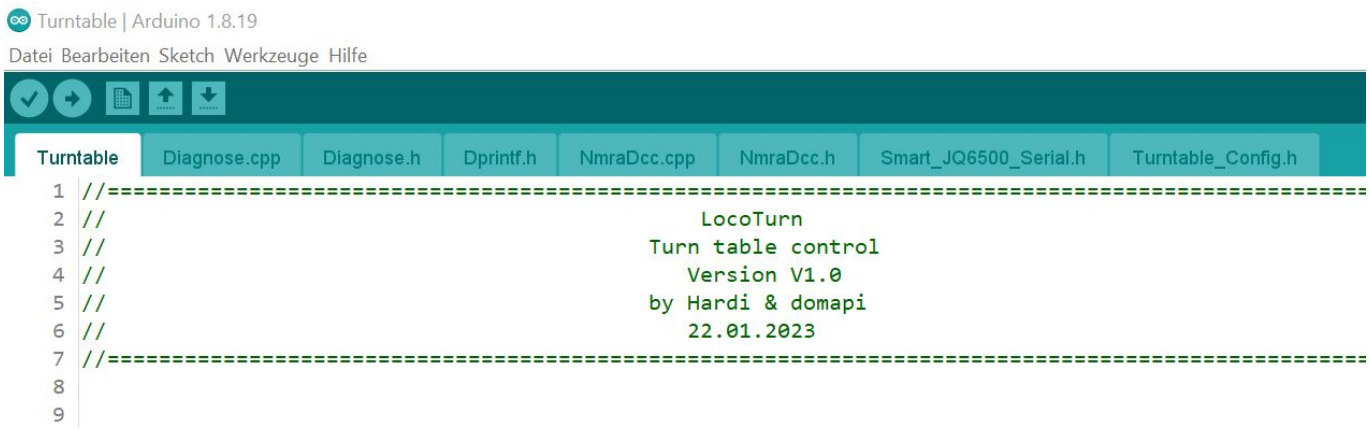
Bei einer neuen Programmversion kopiert man das ino-file in den Reiter „Turntable“. Die Konfiguration im „Turntable_Config“ bitte nicht übernehmen, sonst überschreibt ihr eure Drehscheibeneinstellungen! Ausnahme: neue #defines für neue Funktionen müssen ggf. in der „Turntable_Config.h“ eingebaut werden.

Die meisten Parametereinstellungen müssen direkt im Programm 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).




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

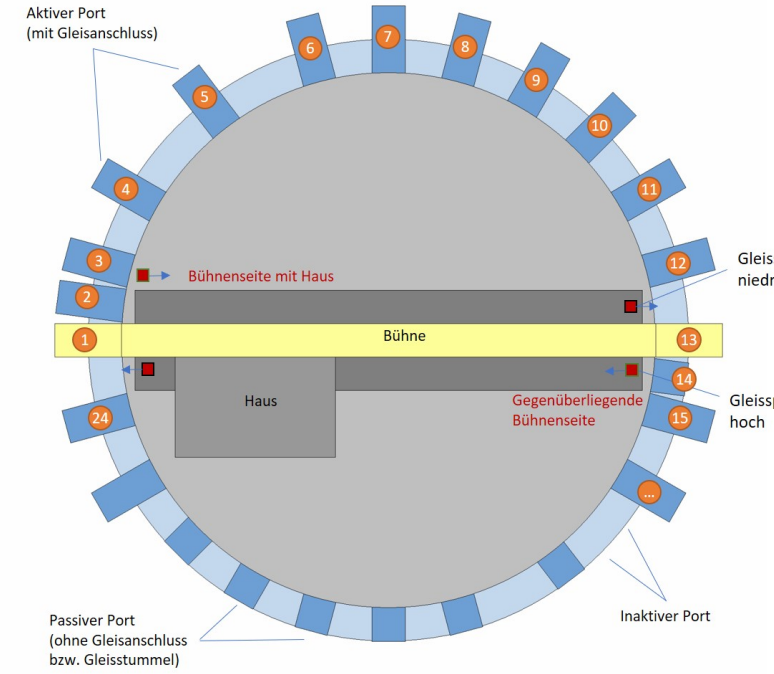
# Definition von Ports/Gleisabgängen

- 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° / 7,5°).
- 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).
- Es gibt immer eine geradzahlige Anzahl Ports.
- Inaktive Ports können von der Steuerung übersprungen werden. Sie werden dann nicht angefahren, wenn die DS mit dem Drehencoder, DCC oder mit den normalen Befehlen im seriellen Arduino-Monitor gesteuert wird! Nur mit dem Poti oder mit speziellen Befehlen im seriellen Monitor sind beliebige Zwischenpositionen anfahrbar.

- Intern werden die typischen 48 Ports von 0 - 47 nummeriert, im Display oder im seriellen Monitor wird 1 - 48 angezeigt. Die Steuerung arbeitet also intern mit Gleisabgängen/Ports von 0 - (x-1), wobei x der Gesamtzahl an Gleisabgängen entspricht. Im Display werden jedoch Werte zwischen 1 und x angezeigt.
- Beim Einrichten der Scheibe muss man sich überlegen, ob wirklich jeder einzelne Gleisabgang angefahren werden soll oder nur die benutzten Ports, an denen Gleise (aktiv) oder kurze Gleisstücke (passiv) angeschlossen sind. Dazu gibt in der Konfiguration entsprechende Parameter.
- Zum ersten Testen empfiehlt es sich, z.B. 12 Ports zu nutzen. Das entspricht einer analogen Uhr und die Uhrzeit-Positionen sollten jedem vertraut sein.

**Fleischmann/Märklin Drehscheibe**

- 48 Ports im 7.5° Raster
- Jeder aktive Port hat ein passives (blindes) Gegenüber (dort sind dann kleine Gleisstücke/ -stummel angebracht)
- Inaktive/unbenutzte Ports sind wahlweise nicht anfahrbar (nur per Poti)
- Im Beispiel rechts werden nur 24 Ports benutzt
- Auf der DS stehen 4 Lichtsignale, 2 jeweils nach außen gerichtet, 2 nach innen.
- Die nach innen gerichteten (hohen) Signale verhindern die Abfahrt von der Bühne,
- Die nach außen gerichteten (niedrigen) Gleissperrsignale die Auffahrt auf die Bühne
- Bei Bewegung der DS sind alle Signale auf rot.
- Bei Stillstand der DS sind nur die Signale auf weiß (= grün), wo das Bühnenende auf einen gültigen Gleisabgang/Port zeigt, an dem also ein echtes Gleis angeschlossen ist.
- Bei passiven Gleisabgängen/Ports ohne Gleis wird rot angezeigt
- Die zwei Signale am jeweiligen Bühnenende werden paarweise parallel zusammengeschaltet.
- Es gibt bei der FLM-Drehscheibe eine Seite mit Haus und eine ohne.




**Exkurs:**
Ein gültiger Port muss immer >= 0 und < PORT_CNT sein; die OLED-Anzeige addiert +1 dazu. Intern wird auch mit negativen Ports und Ports >= PORT_CNT gerechnet, um auch Nullpunkt-Durchgänge in der richtigen Drehrichtung verarbeiten zu können. Im Sketch werden dann Steps für eine komplette Umdrehung addiert/subtrahiert, um z.B. eine Bewegung von der 1 Uhr auf die 11 Uhr Position hinzubekommen, oder umgekehrt.

# Typische Einsatzszenarien

Der Sketch ist sehr speicherintensiv und nutzt den Nano-Speicher quasi komplett aus. Beim Testen 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_philosopie?rev=1676466919**

Last update: **2023/02/15 14:15**