LocoTurn-Ansteuerung über DCC und PC-Steuerungsprogramme

Übersicht der möglichen DCC-Befehle

LocoTurn ist ein **Schaltartikeldekoder** und reagiert auf DCC-Schaltartikel-Kommandos, wie man sie auch zur Steuerung von Weichen- und Signaldekodern verwendet (LocoTurn ignoriert DCC-Lokbefehle!).

LocoTurn nutzt standardmäßig DCC-Befehle von Adresse # 211 bis # 224, um interne Funktionen anzusteuern. Die DCC-Befehle ab # 225 umfassen im wesentlichen die Märklin-kompatiblen Kommandos. Ab # 229 beginnen die Direktfahrbefehle zu bestimmten Ports. Wieviele DCC-Adressen benötigt werden, hängt ab von der Anzahl der verwendeten Ports und ob das **#define DCC_MAERKLIN_7687_COMPATIBLE** auf 0 oder 1 gesetzt wurde.

Über das #define **DCC_OFFSET** kann ein Offset eingestellt werden, damit LocoTurn auf andere DCC-Adressbereiche reagiert. Der Offset wird auf die jeweiligen DCC-Adressen addiert. Bei manchen Zentralen muss z.B. der Roco-Offset von „-4“ hier eingestellt werden, damit LocoTurn die Adressen richtig interpretiert. Bei großen Anlagen mit mehreren Drehscheiben können die Drehscheiben auf diese Weise individuell gesteuert werden.

Sobald die Zentrale (oder ein PC-Steuerungsprogramm über die Zentrale) ein entsprechendes DCC-Kommando sendet, führt LocoTurn die jeweilige Funktion aus. Die Funktionen bestehen immer aus ein Adresse und einer Richtungsangabe (RED / GRN). RED und GRN stehen für rote/grüne Taste in manchen Zentralen. Tatsächlich wird jedoch für RED „0“ und für GRN „1“ gesendet. Pro Adresse lassen sich so 2 Funktionen steuern.

DCC-Adresse	Funktion	#define im Sketch
210, RED	DCC-only-Modus ausschalten (DS reagiert dann auch auf Drehencoder, Poti etc.)	DCCRED_ADDR | |210, GRN |DCC-only-Modus einshalten (DS reagiert nur auf DCC-Kommandos) |DCCGRN_ADDR
211, RED	Gleissperrsignal auf der Bühne (Seite ohne Haus) rot (nur bei WS281x-Board oder MobaLEDLib-Anbindung sinnvoll)	DCC_SPERRSIGNAL_180_RED_ADDR
211, GRN	Gleissperrsignal weiß (grün)	DCC_SPERRSIGNAL_180_GRN_ADDR
212, RED	Gleissperrsignal beim Haus rot (nur bei WS281x-Board oder MobaLEDLib-Anbindung sinnvoll)	DCC_SPERRSIGNAL_RED_HOUSE_ADDR
212, GRN	Gleissperrsignal weiß (grün)	DCC_SPERRSIGNAL_GRN_HOUSE_ADDR
213, RED	Warnlicht AUS: Disable flash on turntable house	DCC_DISABLE_FLASH_ADDR
213, GRN	Warnlicht EIN: Enable flash	DCC_ENABLE_FLASH_ADDR
214, RED	Sound OFF: Disable the automatic generated sound if the turntable starts/stops moving	DCC_DISABLE_SOUND_ADDR

DCC-Adresse	Funktion	#define im Sketch
214, GRN	Sound ON: Enable the automatic generated sound if the turntable starts/stops moving	DCC_ENABLE_SOUND_ADDR
215, RED	Leiser: Decrease the volume	DCC_VOLUME_DN_ADDR
215, GRN	Lauter: Increase the volume	DCC_VOLUME_UP_ADDR
216, RED	Set the sound volume to SOUND_VOLUME1 (10 by default)	DCC_VOLUME_1_ADDR
216, GRN	Set the sound volume to SOUND_VOLUME2 (20 by default)	DCC_VOLUME_2_ADDR
217, RED	Play sound 1	DCC_PLAY_SOUND1_ADDR
217, GRN	Play sound 2	DCC_PLAY_SOUND2_ADDR
218, RED	Play sound 3	DCC_PLAY_SOUND3_ADDR
218, GRN	Play sound 4	DCC_PLAY_SOUND4_ADDR
219, RED	Play sound 5	DCC_PLAY_SOUND5_ADDR
219, GRN	Play sound 6	DCC_PLAY_SOUND6_ADDR
220, RED	Continuously rotate in the positive direction, mostly CW	DCC_ROTATE_POS_DIR_ADDR
220, GRN	Continuously rotate in the negative direction (CCW)	DCC_ROTATE_NEG_DIR_ADDR
221, RED	Licht AUS: Disable the light in the machine house on the turntable	DCC_DISABLE_LIGHT_ADDR
221, GRN	Licht EIN: Enable the light	DCC_ENABLE_LIGHT_ADDR
222, RED	Set the moving speed to MOVE_SPEED1	DCC_SET_SPEED1_ADDR
222, GRN	Set the moving speed to MOVE_SPEED2	DCC_SET_SPEED2_ADDR
223, RED	Set the moving speed to MOVE_SPEED3	DCC_SET_SPEED3_ADDR
223, GRN	Set the moving speed to MOVE_SPEED4	DCC_SET_SPEED4_ADDR
224, RED	Neu ab Version 1.3: U-Turn clockwise (CW)	DCC_U_TURN_CW_ADDR
224, GRN	Neu ab Version 1.3: U-Turn counter clockwise (CCW)	DCC_U_TURN_CCW_ADDR
225, RED	Stop the turntable	DCC_STOPP_ADDR
225, GRN	Calibrate the zero position (During the calibration no other commands are accepted)	DCC_CALIBRATE_ADDR
226, RED	nicht verwendet!	DCC_RESERVE_3_ADDR
226, GRN	Reverse the turntable in DCC direction	DCC_REVERSE_TABLE_ADDR
227, RED	Turn to the next port in positive direction (CW)	DCC_STEP_POS_DIR_ADDR
227, GRN	Turn to the next port in negative direction (CCW)	DCC_STEP_NEG_DIR_ADDR
228, RED	Set movement direction to positive direction, mostly CW (compatibility Märklin turntable decoder)	DCC_POS_DIR_ADDR

DCC-Adresse	Funktion	#define im Sketch
228, GRN	Set movement direction to negative direction (CCW)	DCC_NEG_DIR_ADDR
229, RED	Move to Port #1 in DCC direction	DCC_PORT_ADDR1
229, GRN	Move to Port #2 in DCC direction	DCC_PORT_ADDR2
…	Direktfahrbefehle zu bestimmten Ports, Ergänzung je nach Anzahl	…

Märklin 7686 kompatibler Drehscheibendekoder

LocoTurn ist Adressen-kompatibel zur Märklin 7686/7687 Drehscheibensteuerung, deshalb sollten alle gängigen PC-Steuerungsprogramme LocoTurn-gesteuerte Drehscheiben ansteuern können.

Die Kommandos für eine klassische Märklin-Drehscheibe nutzen Adressen ab # 225 ff.

LocoTurn´s DCC-Adressen orientieren sich inhaltlich am Pseudo-Märklin-Standard 7686:

Befehl	DCC-Kommando	Bemerkung
Stopp Drehscheibe	225, RED	Die Drehscheibenbewegung wird gestoppt
Kalibrierung	225, GRN	Kalibriert den Null-Punkt (nicht Märklin-kompatibler Befehl)
-	226, RED	nicht verwendet
U-Turn	226, GRN	Führt eine 180°-Drehung / U-Turn in DCC-Richtung aus; diese wird mit #228 eingestellt
Step +	227, RED	Dreht die Bühne zum nächsten Port (im Uhrzeigersinn / CW)
Step -	227, GRN	Dreht die Bühne zum vorherigen Port (gegen den Uhrzeigersinn / CCW)
DCC-Richtung —>	228, RED	Ändert die DCC-Richtung auf Uhrzeigersinn / CW, ohne eine Bewegung zu starten
DCC-Richtung ←–	228, GRN	Ändert die DCC-Richtung auf gegen den Uhrzeigersinn / CCW, ohne eine Bewegung zu starten

Auswirkungen "Märklin 7686 kompatibler Drehscheibendekoder"

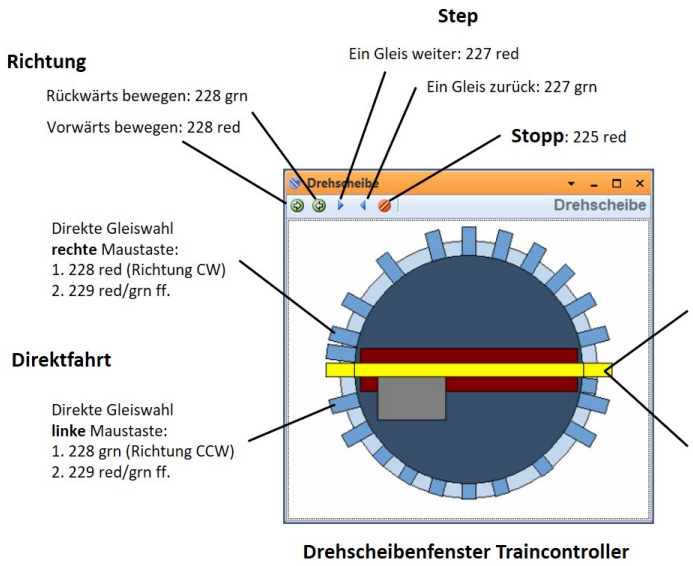
Durch Auswahl von #define **DCC_MAERKLIN_7687_COMPATIBLE** 1 wird die Bewegungslogik eines Märklin-Drehscheibendekoders verwendet. Dieser Quasi-Standard wird von vielen PC-Steuerungsprogrammen eingesetzt, z.B. von Traincontroller (TC). Diese Steuerungsprogramme senden grundsätzlich nur Direkt-Befehle für die **Ports in der ersten Drehscheiben-Hälfte**. Über die Kombination aus der DCC-Drehrichtung und dem Fahrbefehl kann jede beliebige Position der Drehscheibe angefahren werden. Die Bühne dreht immer maximal 180°. Die Drehrichtung bestimmt, ob die Seite mit oder ohne Haus am Zielport hält.

In der config.h reicht es, nur für die erste Drehscheibenhälfte die DCC-Adressen zu definieren.

In TC muss man beim Einrichten der Drehscheibe die Einstellung „Märklin Digitale Drehscheibe 7686 und Kompatible“ nutzen, d.h., die Port-/Gleisadressen folgen dem Märklin Schema: Es werden dann

nur für die aktiven Gleise aus der 1. Drehscheiben-Hälfte DCC-Adressen und -Befehle genutzt.

TC sendet bei Direktanklicken eines Ports im TC-Miniaturbildchen immer 2 Befehle: 1. Richtung, 2. Ziel-Port. Bei den Zielports sendet TC nur die DCC-Befehle für den ersten Drehscheiben-Halbkreis, auch wenn in den 2. Halbkreis gefahren werden soll.



Drehscheibenfenster Traincontroller

LocoTurn	TC	TC-Symbol	Adresse	Bemerkung
Stopp	Stopp		225 red	Stopp
Referenzfahrt	-	-	225 grn	Kalibrierung
-	-	-	226 red	-
U-Turn	U-Turn	-	226 grn	Startet Bewegung
Step +	Ein Gleis weiter		227 red	Startet Bewegung
Step -	Ein Gleis zurück		227 grn	Startet Bewegung
DCC-Richtung ←	Rückwärts bewegen		228 red	Nur Richtung
DCC-Richtung →	Vorwärts bewegen		228 grn	Nur Richtung

Da LocoTurn intern alle Ports direkt anfahren kann (auch die im 2. Halbkreis), muss aus der aktuellen Port-Position, der Drehrichtung und dem Port-Befehl umgerechnet werden, wohin LocoTurn tatsächlich fahren muss und ggf. muss man die Ports für einen Halbkreis addieren (PORT_CNT / 2). Mit diesem „Trick“ gelingt es, die Bühnenseite mit dem Drehscheiben-Haus auch in die 2. DS-Hälfte zu fahren, damit das Haus in der gleichen Position steht wie das TC-Symbol.

Immer wenn die Drehrichtung zum Zielport eine Drehung von mehr als 180° ergeben würde (also der längere Weg wegen der Drehrichtung), dann wird statt dessen die korrespondierende Position in der „kürzer“ zu erreichenden Hälfte angefahren. Durch diesen Algorithmus ergeben sich 48 eindeutig anzufahrende Positionen inklusive der korrekten Häuschenposition.

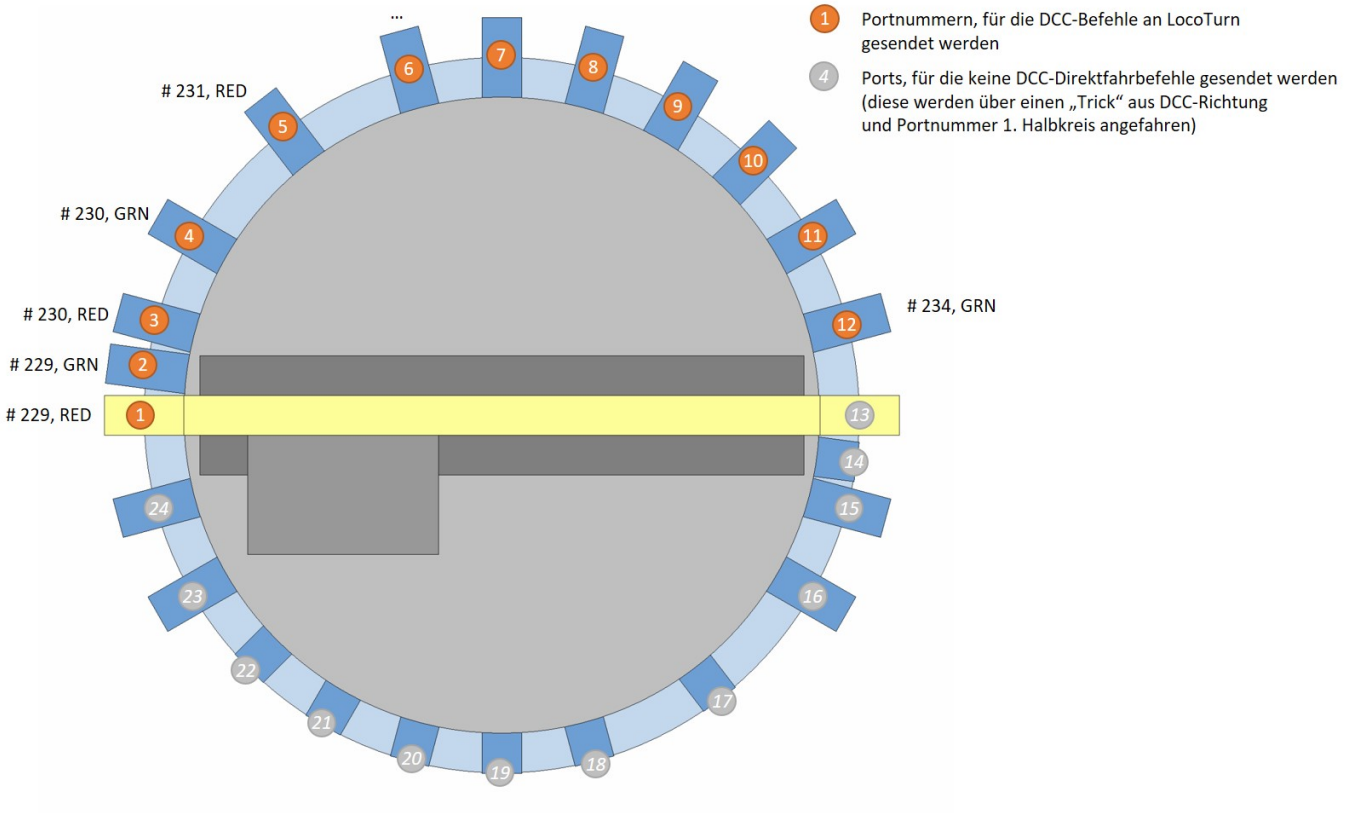
Auf diese Weise lässt sich die Drehscheibe mit TC und LocoTurn synchron steuern, sodass echte Drehscheibe und TC-Symbol auch mit der Hausseite der Bühne übereinstimmen.

Mit dieser Logik ist es in der Regel möglich, LocoTurn auch mit anderen PC-Steuerungen zu verwenden.

Das beißt sich allerdings mit der LocoTurn-Logik „kürzeste Strecke berechnen“. Die wird ausgehebelt, da aus dem Steuerungsprogramm immer 2 DCC-Befehle kommen: Richtung und Zielport. In der vorgegebenen Richtung wird zum Zielport gefahren, was u.U. nicht die kürzeste Route sein kann.

Beispiel:

- Bei 24 verwendeten Ports sendet TC nur die DCC-Befehle für Port 1 - 12.
- Port 13 - 24 werden nie gesendet, da für TC Port 1 = Port 13 ist (12 = 24 usw.).
- Die DCC-Adressen # 229, RED - 234, GRN werden verwendet (1 - 12).
- 235, RED - 240, GRN (13 - 24) werden nicht benötigt.



Durch Auswahl von #define **DCC_MAERKLIN_7687_COMPATIBLE** 0 wird die Märklinlogik deaktiviert. Dies empfiehlt sich nur, wenn die Drehscheibe nicht über ein PC-Steuerungsprogramm gesteuert werden soll. Stattdessen kann man nun auch DCC-Befehle von der Zentrale für die 2. Drehscheiben-Hälfte senden und LocoTurn wird zum gewünschten Port fahren. In diesem Fall wird immer der kürzeste Weg gewählt und die DS dreht entsprechend links- oder rechtsherum. Natürlich muss man in der config.h dann auch für alle Ports die DCC-Adressen hinterlegen.

Für U-Turns gibt es ab Software-Version 1.3 die DCC-Adresse 224; hiermit kann ein U-Turn in Uhrzeigerrichtung (RED) und gegen den Uhrzeiger (GRN) ausgelöst werden.

Trennung von Empfang und Verarbeitung von DCC-Befehlen

Der Empfang von DCC-Befehlen ist sehr zeitkritisch. Hierzu muss die Dcc.process()-Routine der DCC-Library sehr häufig aufgerufen werden, das gilt sowohl für die Einbindung in die loop() als auch in irgendwelche Unterroutinen, die im Sketch ablaufen, z.B. bei OLED-Ausgaben.

Daher werden die empfangenen DCC-Befehle zunächst in einer kleinen Tabelle gepuffert und erst kurze Zeit später von einer separaten Routine abgearbeitet.

Bei Ansteuerung von LocoTurn mit einer PC-Steuerungssoftware über eine Zentrale sendet die Software i.d.R. bei DS-Bewegungen 2 Befehle: 1. die Richtung, 2. den Zielport (oder die U-Turn-Bewegung). Diese beiden Befehle müssen zwingend empfangen werden können, damit LocoTurn sauber arbeitet.

Bei der ESU-ECOS-Zentrale wird z.B. für jeden Befehl 4 x On und nach kurzer Pause 4 x Off gesendet und danach der nächste Befehl etc. Andere Programme/Zentralen senden ihre DCC-Kommandos in unterschiedlichen Zeitabständen, z.B. alle On-Befehle quasi auf einmal ohne Pause dazwischen oder schön mit Pause nacheinander. Manche Zentralen senden die Befehle weniger häufig oder senden gar keine Off-Befehle. LocoTurn muss sicherstellen, dass mehrere Befehle quasi gleichzeitig empfangen und verarbeitet werden können.

Wenn die Dcc.process() zu selten aufgerufen wird, besteht das Risiko, dass DCC-Kommandos verpasst werden. Daher wird nach einem ersten DCC-Befehl einige 100 ms gewartet, bevor der DCC-Befehl ausgeführt wird. Das hilft, weitere DCC-Befehle sicher einzusammeln. Die Wartezeit lässt sich über #define **DCC_WAITING_TIME** definieren, sinnvolle Zeiten sind 50 – 350 ms.

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

Permanent link:
https://wiki.mobaledlib.de/anleitungen/bauanleitungen/locoturn_v10/150_locoturn_pc_steuerung?rev=1772011210

Last update: **2026/02/25 09:20**