

# Ansteuerung von Formsignalen mit Ministeppern

## Formsignale mit dem Mini-Stepper bewegen

Quelle: MLL Stammtisch Mai 2021

Tipps zu den Platinen, zum Einbau des Steppers und erste grundsätzliche Überlegungen zur Steuerung mit der MLL sind vom Anfang des Videos bis Minute 32 zu finden. Ab Minute 33 wird die Steuerung des zweibegriffigen Formsignals vorgeführt.

**Nachträglich hat sich jedoch leider herausgestellt, dass die Einstellungen nicht oder nicht zuverlässig unkontrollierte Bewegungen des Signals beim Einschalten oder Restart verhindern. Gerade bei der Nutzung von Programmen wie ROCRAIL ist im automatischen oder halbautomatischen Betrieb ein definierter Anfangszustand sinnvoll.**

Die Einstellungen wurden daher gegenüber der Stammtischversion geändert.

## Benötigte Teile

:

- Hauptplatine über USB- Kabel mit PC verbunden
- Stromversorgung für Steppermotor
- Formsignal mit eingebautem Mini-Steppermotor
- Bestückte Stepperplatine mit einer (550) oder drei Anschlussmöglichkeiten (551) für Stepper-Motoren

## Ablauf:

Den

## Pattern-Configurator

aufrufen und, wenn bisher noch nicht durchgeführt, aus den Beispielen die Signale laden.

Das „Dep Signal4“ Beispiel aufrufen und über die Excel-Funktionen von diesem Beispiel eine Kopie anlegen. Im Bild ist die Kopie unter „Formsignal“ abgespeichert. Das Beispiel wird als Grundlage verwendet.

Nun in das Blatt „Formsignal“ wechseln und die folgenden Werte eintragen:

Ver.: 3.1.0 28.11.21

Erste RGB LED: 1  
Startkanal der RGB LED: 0  
Schalter Nummer: SI\_1  
Anzahl der Ausgabe Kanäle: 2  
Bits pro Wert: 8 => 256 Helligkeitsstufen (0..255)  
Wert Min: 0  
Wert Max: 255  
Wert ausgeschaltet: 0  
Mode: PM\_SEQUENZ\_NO\_RESTART  
Analoges Überblenden:  
Goto Mode: 1  
Goto Aktivierung: N\_OneTimeBut1  
Grafische Anzeige: 1  
Spezial Mode:

Neues Blatt

by Hardi

Dieses Blatt zeigt die Konfiguration eines Ausfahrtsignals.  
Das Ausfahrtsignal soll 4 Aspekte darstellen: HP0, HP1, HP2 und HP0 + SH1.  
Dieses Beispiel ist für den Test mit einem RGB LED Stripe gedacht. Darum werden pro Signallampe drei Zeilen bei  
Auf der Modelleisenbahn wird ein WS2811 Modul zur Ansteuerung der LEDs im Signal verwendet wie es im nächsten  
Beispiel gezeigt ist.  
Hier müssen zur Mischung der Farben Gelb und Weiß mehrere LEDs angesprochen werden.  
Die Signalfelder werden über die Goto Funktion angesprochen. Die entsprechende Nummer (0-3) wird im Programm  
z.B. per DCC eingelesen. Der "E"-Pfeil deutet an, dass das Modul nach der Anzeige eines Bildes beendet wird.  
Die Lampen sollen nicht schlagartig an oder aus gehen. Das wird über den Schalter "Analoges Überblenden" aktiv.  
Die Visualisierung mit den verschiedenen Dreiecken zeigt das die Helligkeit einer Lampe sich langsam von einem  
unbekannten Anfangswert zum vorgegebenen Endwert verändert. Die beiden roten Lampen sollen bei HP0 an  
Abhängig davon ob die Lampe vorher an oder aus war verändert sich die Helligkeit innerhalb der vorgegebenen  
so, dass die Lampe am Ende leuchtet.

Aktualisieren Test Pattern

1 2

Ergebnis: `PatternT1(1,28,SI_LocalVar,2,0,255,0,PM_SEQUENZ_NO_RESTART,4 Sek,0,0,127,0,0,0,127,255,0,0 ,63,128,63,128,63) // _Dep_Signal4`

Makro Name: `_Dep_Signal4`

Makro: `#define _Dep_Signal4(LED) PatternT1(LED,28,SI_LocalVar,2,0,255,0,PM_SEQUENZ_NO_RESTART,4 Sek,0,0,127,0,0,0,127,255,0,0 ,63,128,63,128,63)`

#define \_Dep\_Signal4\_StCh(LED,StCh) PatternT1(LED,StCh+28,SI\_LocalVar,2,0,255,0,PM\_SEQUENZ\_NO\_RESTART,4 Sek,0,0,127,0,0,0,127,255,0,0 ,63,128,63,128,63)

Wenn gleiche Zeiten verwendet werden, dann sollten nur die ersten Zeiten eingetragen werden. Bei leeren Spalten werden die vorangegangenen Zeiten wiederholt. Das reduziert die

Dauer: 4 Sek

Flash Bedarf: 23 Bytes

0 1 2 E E S E S E

Goto Tabelle

RGB LED

LED Nr	Spalte Nr ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	Rot 1		127		127															
2	Grün		0		255	.														

Program Generator

Formsignal Main Imported\_Welding Entry Signal3 AmpelX AmpelX\_Off RGB\_AmpelX RGB\_AmpelX\_Off RGB\_AmpelX\_Fade RGB\_AmpelX\_Fade\_Off RGB\_AmpelX\_A C

Bereit

Für die Steuerung des Steppers brauchen wir zwei Steuerbefehle:

- Stepper ein-/ausschalten – über den roten Kanal mit dem Wert 127
- Drehrichtung des Steppers – über den grünen Kanal, Wert 255 eine Drehrichtung, Wert 0 entgegengesetzt

**Anmerkung:** Stimmen später im Prog-Gen die Tasten GRÜN oder ROT nicht mit der Einstellung des Signals überein, kann man hier einfach Grün in der zweiten Spalte auf 255 (oder x entspricht 255) und in der vierten Spalte auf 0 (oder . entspricht 0) setzen. Anschließend erneut zum Prog-Gen senden.

Der blaue Kanal kann zur Steuerung einer LED verwendet werden, z.B. der Signalbeleuchtung. Diese Funktion nutze ich im Prog-Gen, nicht hier im Pattern-Config.

Den Wert für die Ausgabekanäle auf „2“ stellen. In der unteren Tabelle können die Werte für die anderen Ausgabekanäle gelöscht werden.

Für die anderen Einstellungen sind folgende Werte einzutragen bzw. zu ändern:

- Bits pro Wert: „8“
- Mode: „PM\_SEQUENZ\_NO\_RESTART“ (Flanken getriggerte einmalige Sequenz. Kein Neustart während der Laufzeit)
- Analoges Überblenden: „0“ (schaltet das Überblenden ab)
- Goto Aktivierung: „N\_OneTimeBut1“

## Goto Aktivierung auswählen:

Wenn der Goto Mode verwendet wird, dann kann das Muster an verschiedenen Stellen gestartet werden. Die Spalten sind in der Tabelle mit nummerierten Pfeilen markiert.

Mit diesem Dialog wird definiert wie die Startspalte im Betrieb ausgewählt wird.

Name	Beschreibung
N_Buttons	Bis zu 64 Taster steuern die Startspalte. Start bei Goto 0.
N_Buttons1	Bis zu 64 Taster steuern die Startspalte. Start bei Goto 1.
N_OneTimeBut	Bis zu 64 Taster steuern die Startspalte. Start bei Goto 0. Startposition wird nur ein mal aktiviert.
N_OneTimeBut1	Bis zu 64 Taster steuern die Startspalte. Start bei Goto 1. Startposition wird nur ein mal aktiviert.
Binary	Die Startposition wird durch die binäre Kombination von bis zu 6 Eingänge bestimmt.
Binary1	Die Startposition wird durch die binäre Kombination von bis zu 6 Eingänge bestimmt. Start bei Goto 1
Counter	Der Startpunkt wird über einen Zähler bestimmt welcher z.B. von einem Taster erhöht wird.
RandButton	Beim Betätigen eines Tasters wird eine zufällige Startposition ausgewählt

Über die Taste „Programm Generator“ die Daten nun zum Prog-Gen schicken.

Im

## Prog-Gen

die Funktion an der gewünschten Stelle einfügen. Hier wurde ein neues Blatt zu Testzwecke angelegt.

1	Anforderung: board arduino zur nano cpu atmega200old - prel programmieren																				
2	aktiv	filter	Adresse oder Name	COM6	Typ	Start wert	Beschreibung	Vorteiler Nummer	Stecker Nummer	Icon	Name	Beleuchtung, Sound, oder andere Effekte					Start LedNr	LEDs	InCh	Loc InCh	LED Sound Kanal
3	✓						LED auf dem Mainboard			♥	Heartbeat LED, einstellbar	RGB_Heartbeat2(#LED, 5, 100)	0	1	0	0	0				
4																					
5	✓									📦	Letzte Zustände speichern	#define ENABLE_STORE_STATUS()					0	0			
6										⚙️	Muster Pattern_Configurator	// Activation: N_OneTimeBut1 Setch to LocalVar1(#InCh, 2) PatternT1(#LED, 28, SI_LocalVar, 2, 0, 255, 0, PM_SEQUENZ_NO_RESTAR T, 4 Sek, 0, 0, 127, 0, 0, 127, 255, 0, 0, 63, 128, 63, 128, 63) Const(#LED, C3, #InCh, 0, 100)	1	C1-2	2	0	0				
7	✓		200	Rot	🟢	0	_Dep_Signal4 (pc)														
8																					
9	✓		201	AnAus	🔴					⚙️	LED einstellbar	Const(#LED, C3, #InCh, 0, 100)	1	C3-3	1	0	0				
10																					
11	✓		3	AnAus	🔴		Formsignal cal			⚙️	LED einstellbar	Const(#LED, C1, #InCh, 0, 127)	2	C1-1	1	0	0				
12	✓		4	AnAus	🟢		Formsignal cal			⚙️	LED einstellbar	Const(#LED, C2, #InCh, 0, 255)	2	C2-2	1	0	0				
13	✓		5	AnAus	🔴		Formsignal cal			⚙️	LED einstellbar	Const(#LED, C3, #InCh, 0, 50)	2	C3-3	1	0	0				
14																					
15																					
16										♥	Heartbeat LED, einstellbar	RGB_Heartbeat2(#LED, 5, 100)	3	1	0	0	0				
17																					
18																					
19																					

Zunächst nur die Zeilen 3, 16 (Heartbeat) und 11 bis 13 aktivieren und zum Arduino schicken. Diese Zeilen dienen zur erstmaligen Einstellung des Signals. Zeile 11 ist der rote Kanal, der den Stepper über „ROT/GRÜN“ ein- bzw. ausschaltet. Über Zeile 12 kann die Drehrichtung bestimmt werden. Eingeschaltet (GRÜN) wird der Wert 255 gesendet, ausgeschaltet (ROT) der Wert 0 und damit die entgegengesetzte Drehrichtung.

Zeile 13 dient zum Schalter der Signalbeleuchtung. Diese Funktion hat keinen Einfluss auf den Stepper und muss nicht genutzt werden.

Nach der Einstellung des Signals können die Zeilen 11 bis 13 deaktiviert und die Zeilen 6, 8, 9 aktiviert werden.

Zeile 8 beinhaltet die im Pattern-Configurator erzeugte und importierte Funktion.

**ACHTUNG:** Mit der Zeile 6 wird der letzte Zustand gespeichert „#define ENABLE\_STORE\_STATUS()“. Damit wird erreicht, dass beim nächsten Einschalten der Anlage, bei einem Reset des Nano oder nach Unterbrechung der Stromversorgung der letzte Zustand wieder eingenommen wird.

(Die letzten Zustände bei Signalen oder anderen per DCC, Selectrix oder CAN gesteuerten Funktionen werden gespeichert und beim nächsten Start wieder aktiviert. Wenn der Modus nicht aktiviert ist, dann sind die entsprechenden Funktionen abgeschaltet bzw. beginnen mit dem in der Spalte „Start Wert“ definierten Zustand.)

Diese Funktion funktioniert hier **NICHT!** Wenn sie im Prog-Gen bereits vorhanden ist für andere Funktionen genutzt wird, muss sie durch eine „0“ in der Zeile „Startwert“ für die Signalfunktion abgeschaltet werden.

---

From:

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

Permanent link:

<https://wiki.mobaledlib.de/anleitungen/stepper/signale?rev=1640105868>

Last update: **2021/12/21 17:57**

