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

:

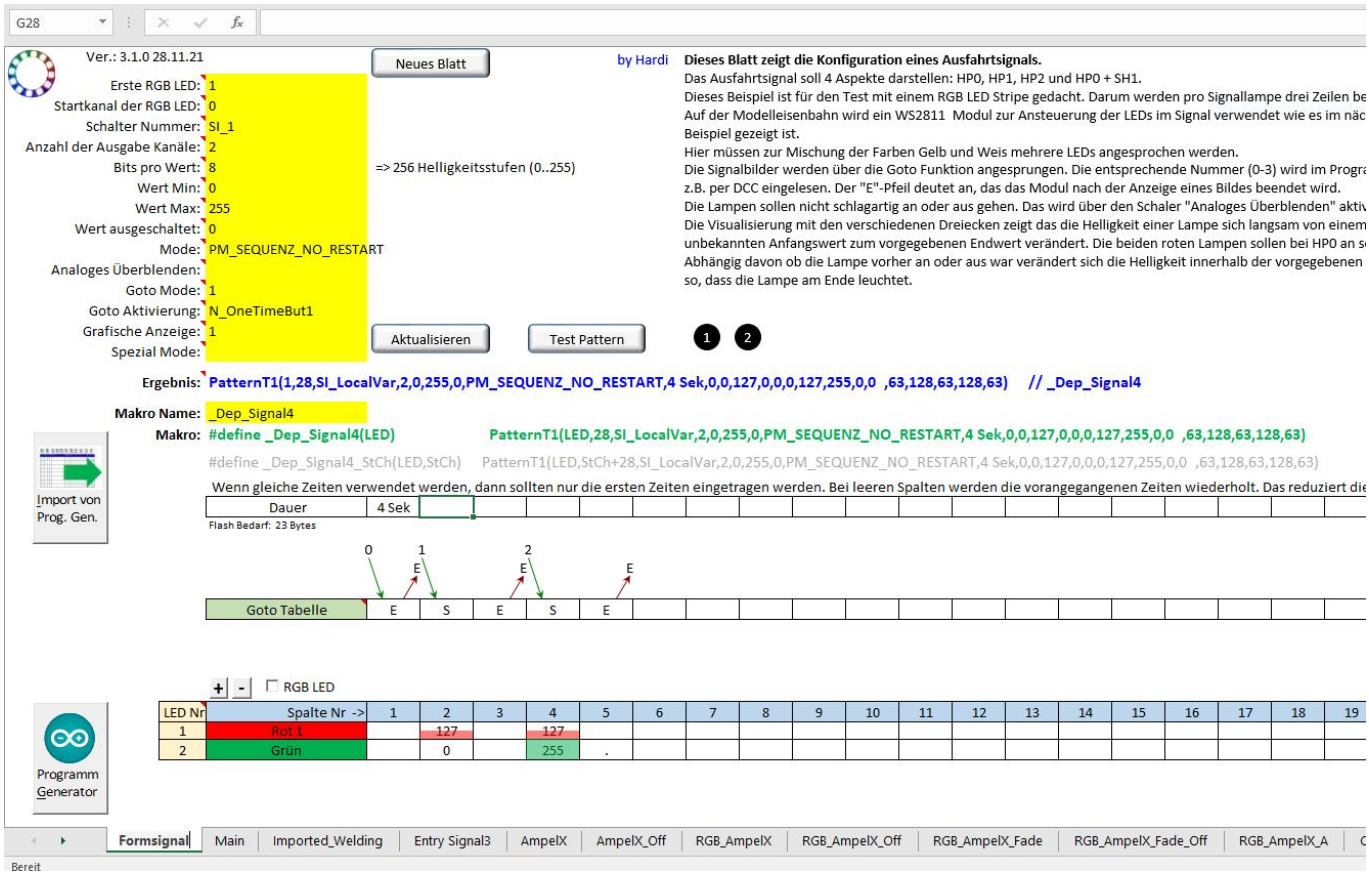
- Hauptplatine über USB- Kabel mit PC verbunden
- Stromversorgung für Steppermotor
- Formsignal mit eingebautem Mini-Steppermotor
- Bestückte Stepperplatine mit einer (550) oder drei Anschlussmöglichkeiten (551) für Stepper-Motoren

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:



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

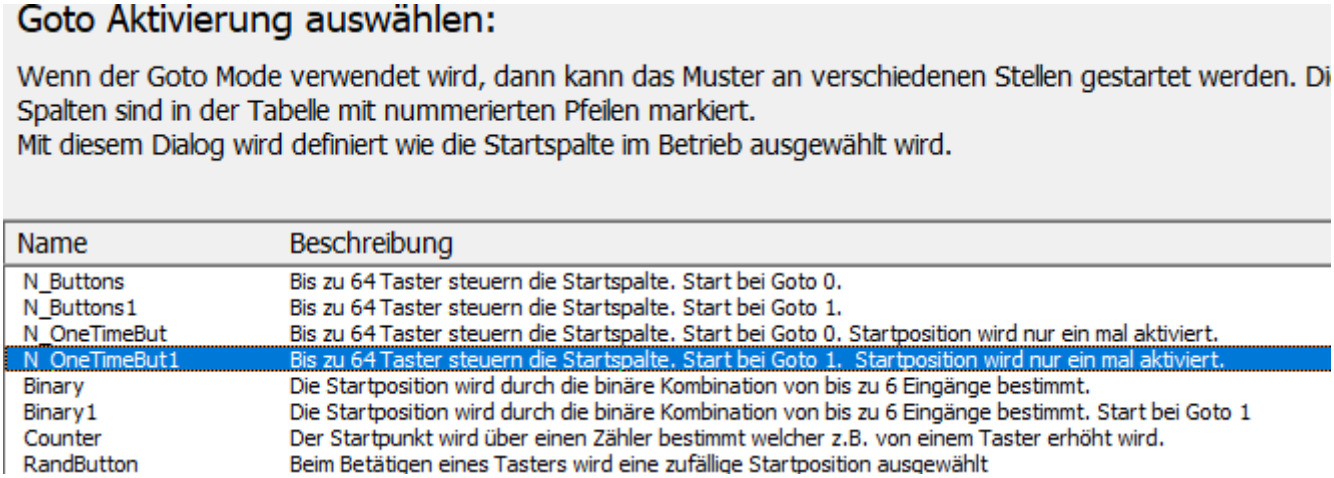
Anmerkung: Stimmen später im Prog-Gen die Tasten GRÜN oder ROT nicht mit der Einstellung des Signals überein, kann man hier einfach Grün in der zweiten Spalte auf 255 (oder x entspricht 255) und in der vierten Spalte auf 0 (oder . entspricht 0) setzen. Anschließend erneut zum Prog-Gen senden.

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 könne die Werte für die anderen Ausgabekanäle gelöscht werden.

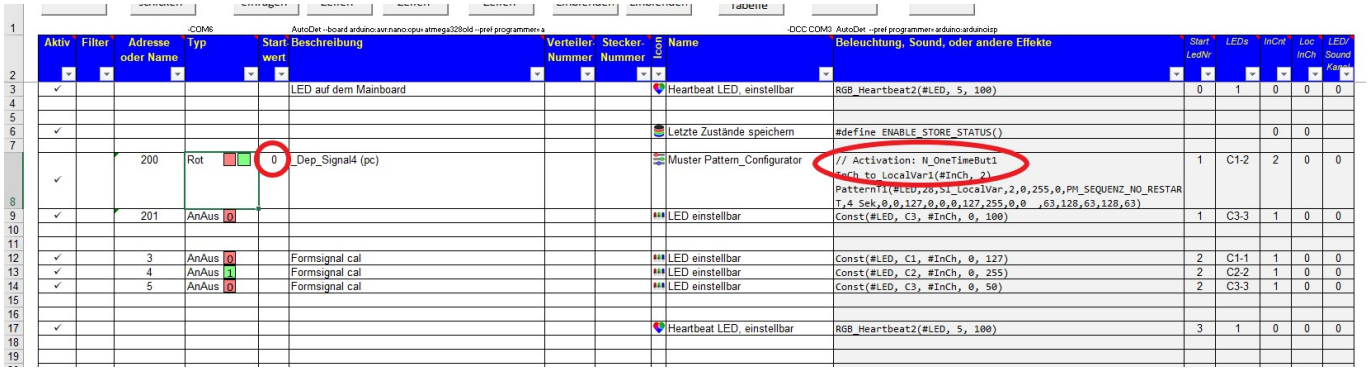
Für die anderen Einstellungen sind folgende Werte einzutragen bzw. zu ändern:

- 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 Activierung: „N_OneTimeBut1“



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.



	Aktiv	Filter	Adresse oder Name	Typ	Start-wert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Icon	Name	Beleuchtung, Sound, oder andere Effekte	Start LedNr	LEDs	InCnt	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																
6	✓									Letzte Zustände speichern	#define ENABLE_STORE_STATUS()			0	0	
7																
8	✓		200	Rot	0	_Dep_Signal4 (pc)				Muster Pattern_Configurator	// Activation: N_OneTimeBut1 InCh_to_LocalVar1(#InCh, 2) Pattern11(#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)	1	C1-2	2	0	0
9	✓		201	AnAus	0					LED einstellbar	Const(#LED, C3, #InCh, 0, 100)	1	C3-3	1	0	0
10																
11																
12	✓		3	AnAus	0	Formsignal cal				LED einstellbar	Const(#LED, C1, #InCh, 0, 127)	2	C1-1	1	0	0
13	✓		4	AnAus	1	Formsignal cal				LED einstellbar	Const(#LED, C2, #InCh, 0, 255)	2	C2-2	1	0	0
14	✓		5	AnAus	0	Formsignal cal				LED einstellbar	Const(#LED, C3, #InCh, 0, 50)	2	C3-3	1	0	0
15																
16																
17	✓									Heartbeat LED, einstellbar	RGB_Heartbeat2(#LED, 5, 100)	3	1	0	0	0
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.

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. __

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

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

Permanent link:
https://wiki.mobaledlib.de/anleitungen/stepper/signale?rev=1640105663

Last update: **2021/12/21 17:54**