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 gelben 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“

Bild Prog-Gen einfügen

Bild Goto Aktivierung einfügen

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.

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.

Nach der Eingabe der Adresse muss der Typ als „Grün“ oder „Rot“ definiert werden

Mit Zeile 9 kann bei Bedarf eine LED, z.B. die Signalbeleuchtung, geschaltet werden.

https://wiki.mobaledlib.de/playground/playground

ALTER TEXT 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, die im Folgenden erläutert wird.

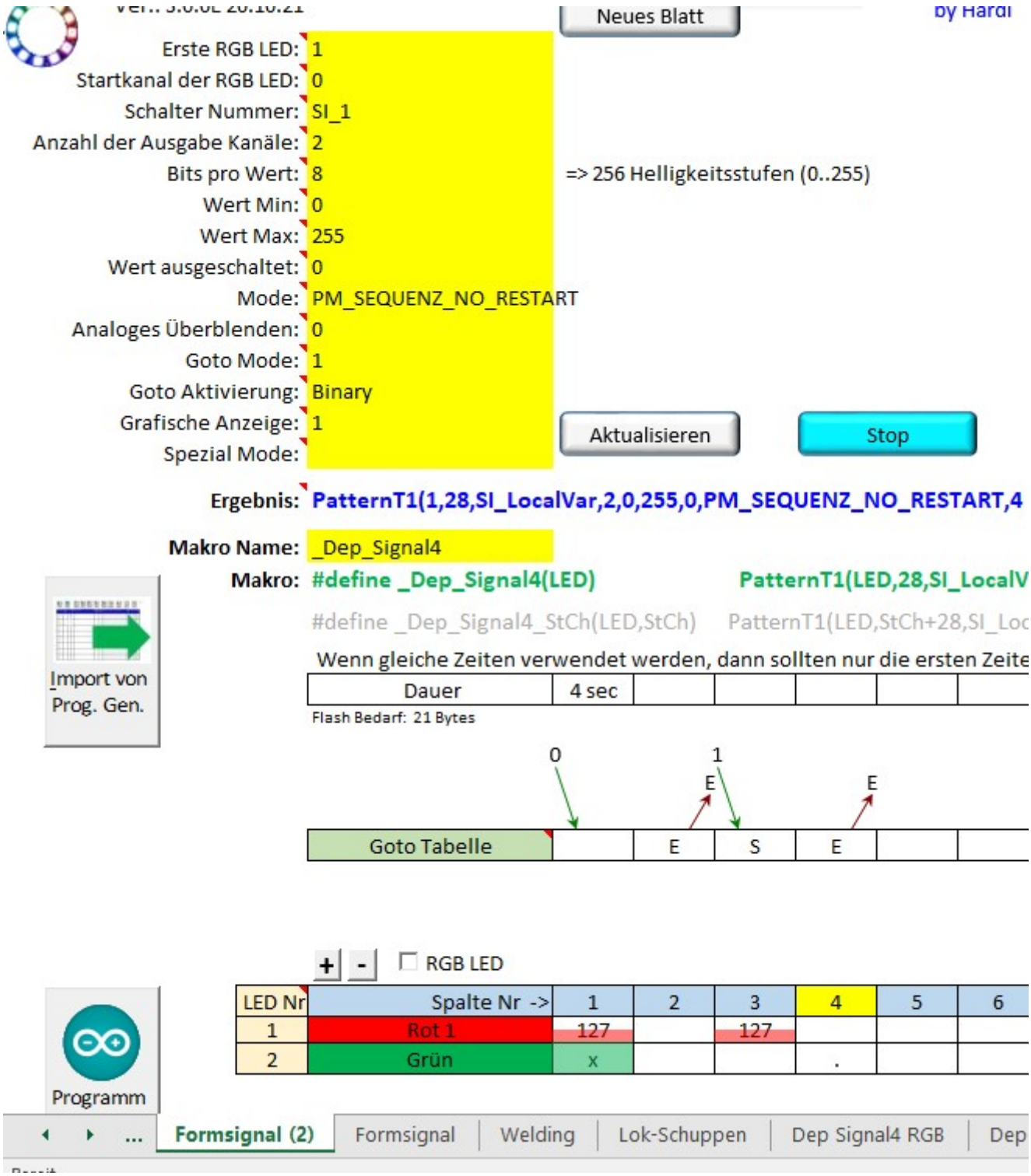
Benötigte Teie:

- 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 Schaltfläche **Neues Blatt** eine Kopie anlegen. Die Einstellungen übernehmen und einen neuen Namen vergeben, hier „Formsignal (2)“. Das Beispiel wird als Grundlage verwendet.

Nun in das neue angelegte Blatt wechseln.



Für die Steuerung des Steppers brauchen wir zwei Steuerbefehle:

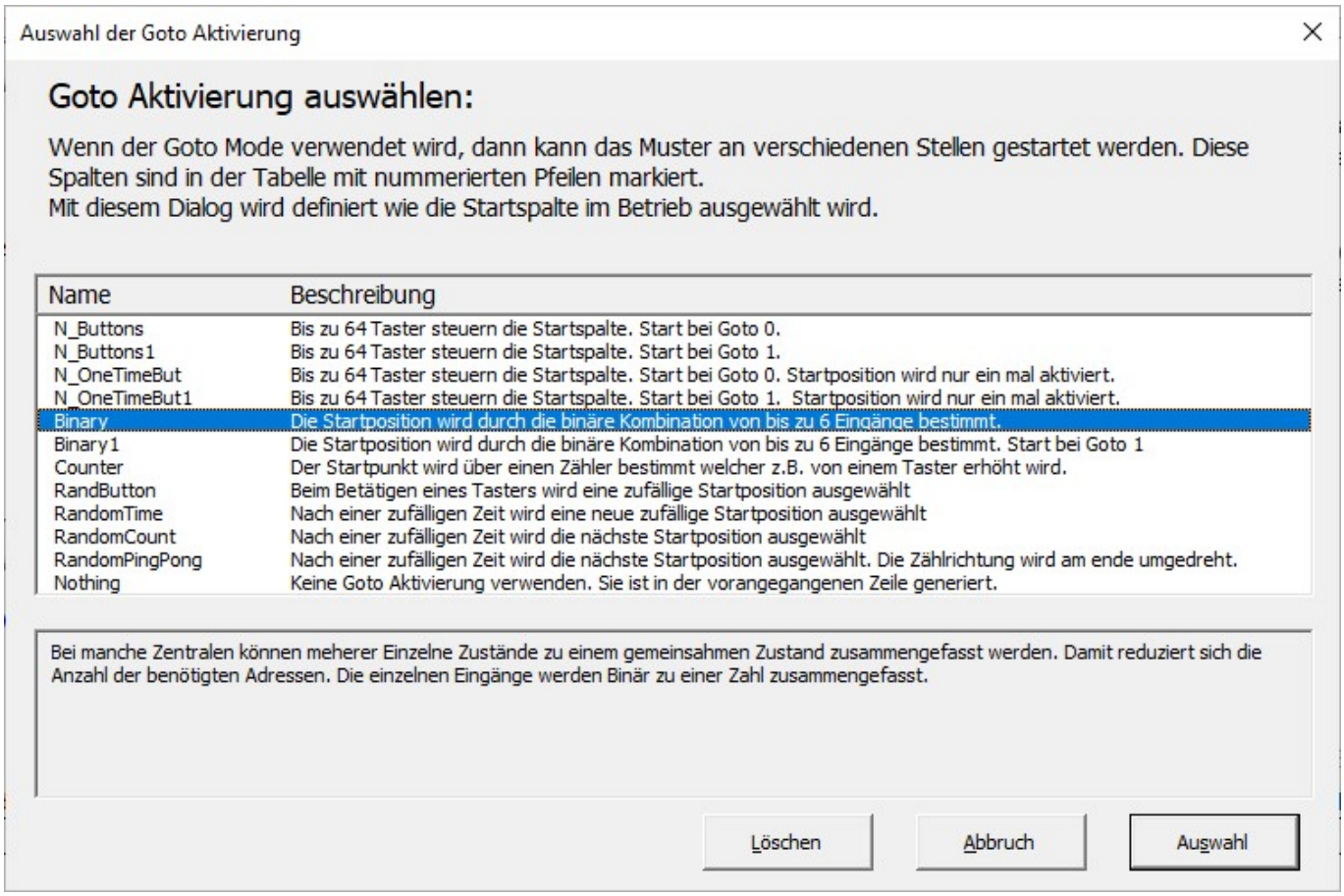
- Stepper ein-/ausschalten – über den roten Kanal (LED Nr 1) mit dem Wert 127
- Drehrichtung des Steppers – über den grünen Kanal (LED Nr 2), Wert 255 eine Drehrichtung, Wert 0 entgegengesetzt

Der blaue Kanal, das wäre dann die LED Nr 3 wenn die Anzahl der Ausgabekanäle auf „3“ erhöht wird, kann zur Steuerung einer LED verwendet werden, z.B. der Signalbeleuchtung. Diese Funktion nutze ich im Prog-Gen, nicht hier im Pattern-Config.

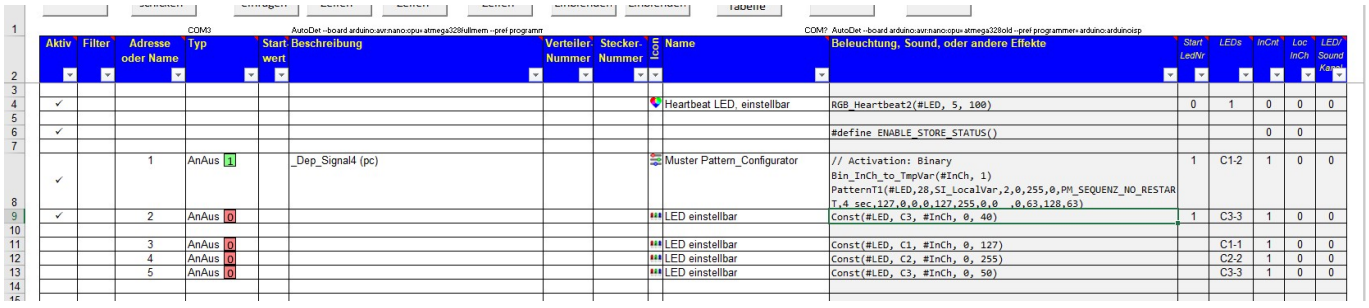
Daher den Wert für die **Ausgabekanäle auf „2“** einstellen. In der unteren Tabelle könne die Werte für die anderen Ausgabekanäle gelöscht werden.

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

- Anzahl der Ausgabe Kanäle: 2
- Bits pro Wert: 8
- Mode: „PM_SEQUENZ_NO_RESTART“ (Flanken getriggerter einmalige Sequenz. Kein Neustart während der Laufzeit)
- Analoges Überblenden: „0“ (schaltet das Überblenden ab)
- Goto Activierung: „Binary“ (Bild einfügen, mit dieser Einstellung kann dann im Prog-Gen der Ein/Aus-Schalter gewählt werden, um ein mehrfaches Bewegen der Signalflügel in die selbe Richtung auszuschließen.)



Über die Taste **„Programm Generator“** die Daten zum Prog-Gen schicken. Im Prog-Gen die Funktion an der gewünschten Stelle einfügen.



Für Testzwecke habe ich ein neues Blatt angelegt.

- Zunächst nur die Zeilen **4** (Heartbeat) und 11 bis 13 aktivieren und zum Arduino schicken.

Diese Zeilen sind zur erstmaligen Einstellung des Signals vorgesehen.
- 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 Schalten der Signalbeleuchtung. Diese Funktion hat keinen Einfluss auf den Stepper und muss nicht genutzt werden.

Nach der Grundeinstellung des Signals können die Zeilen 11 bis 13 deaktiviert, die Zeilen 6, 8, 9 aktiviert und zum Arduino geschickt werden.

- 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 definierte 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 in der Spalte „Start Wert“ definierten Zustand.*)
- Zeile **8** beinhaltet die vom Pattern-Config erzeugte Funktion. Nach der Eingabe der Adresse muss der Typ als „**Ein/Aus**“ definiert werden, nicht als Taster!, um eine mehrfache Bewegung der Flügel des Signals in eine Richtung auszuschließen.
- Mit Zeile **9** kann bei Bedarf eine LED, z.B. die Signalbeleuchtung, geschaltet werden.