

Servosteuerung und Herzstückpolarisation mit Selectrix

Beschreibung für die Servos Makros

Beschreibung der Selectrix Ansteuerung für die Servoplatine und einer Herzstückpolarisierung für Weichen im Zweileitersystem.

Alle erforderlichen Makros wurden mit dem Pattern-Konfigurator erstellt.

Für die Servos

Macroname: Servo_rot

Makroname: Servo_gruen

Makroname: Servo_blau

Für die Herzstückpolarisierung

Makroname: Herz_1

Makroname: Herz_2

Makroname: Herz_3

Im Prog_Generator der MobaLedLib wird die gleiche Selectrix Adresse für den Servo **und** für die dazu gehörende Herzstückpolarisierung verwendet. z.B:

70	3	AnAus	3	Servo rot 1. Servoplatine	1	2	// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	5	C1-1
70	4	AnAus	4	Servo gruen 1. Servoplatine	1	2	// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	5	C2-2
70	5	AnAus	5	Servo blau 1. Servoplatine	1	2	// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	5	C3-3
70	3	AnAus	3	Herz_1 für die erste Weiche			// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	6	C1-2
70	4	AnAus	4	Herz_2 für die zweite Weiche			// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	6	C3-4
70	5	AnAus	5	Herz_3 für die dritte Weiche			// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	7	C2-3
71	3	AnAus	3	Servo rot 2. Servoplatine	1	4	// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	8	C1-1
71	5	AnAus	5	Servo gruen 2. Servoplatine	1	4	// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	8	C2-2
71	7	AnAus	7	Servo blau 2. Servoplatine	1	4	// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	8	C3-3
71	3	AnAus	3	Herz_1 für die vierte Weiche			// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	9	C1-2
71	5	AnAus	5	Herz_2 für die fünfte Weiche			// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	9	C3-4
71	7	AnAus	7	Herz_3 für die sechste Weiche			// Activation: BinaryBin_InCh_to_TapVar(#InCh, 1)P	10	C2-3
71.1-71.2	1	AnAus	1 2		1	3	EntrySignal3Bin(#LED, #InCh, 128)	11	C1-3
				Sicherung des letzten Zustandes			#define ENABLE_STORE_STATUS		



Erste RGB LED:	7
Startkanal der RGB LED:	0
Schalter Nummer:	SI_1
Anzahl der Ausgabe Kanäle:	1
Bits pro Wert:	8
Wert Min:	0
Wert Max:	255
Wert ausgeschaltet:	0
Mode:	0
Analoges Überblenden:	
Goto Mode:	1
Goto Aktivierung:	Binary
Grafische Anzeige:	1
Spezial Mode:	

Parameter des Pattern-Configurator Ein Auszug aus der o. g. Tabelle

Erste RGB LED:	Der Wert „7“ hat keine Bedeutung, dieser Wert wird beim Übertragen in den Prog. Generator nicht verwendet
Startkanal der RGB LED: 0	
Schalter Nummer:	SI_1 wird im Goto Modus nicht benutzt. SI = Special Input, 1 = immer an.
Bits pro Wert 8	

verwendet werden. (Wert Max /Min werden dann nicht mehr verwendet) |

Wert Max: 255	Wird verwendet, wenn „Bits pro Wert“ nicht 8 ist. Dann kann über diesen Wert bestimmt werden, wie der maximale Bit Wert interpretiert wird.
---------------	---

Drei Zeilen, jede für einen Servo auf der Servoplatine. Vollständige Darstellung eines Makros

```
// Activation: Binary Bin_InCh_to_TmpVar(#InCh, 1)  
PatternT1(#LED,128,SI_LocalVar,2,0,255,0,0,400 ms,4,2 ,0,0,63,128,0,63)
```

Damit der Status der Weichenpositionen festgehalten wird, kommt wird z.B. am Ende der Konfiguration folgende Zeile eingefügt

```
#define ENABLE_STORE_STATUS
```

„**#define ENABLE_STORE_STATUS**“ muss nur einmal, irgendwo in der Konfiguration stehen.

Beschreibung für die Herzstückpolarisierungs Makros

Diese Herzstückpolarisierung wurde bei Peco Weichen integriert, dabei wurden beide



Umschaltkontakte der Relais verwendet.



Die Mittelschlüsse 1 werden mit dem Herzstück verbunden , Anschluss 5 und 4 gehen zu den Aussenschienen und die Anschlüsse 3 und 2 gehen zu den Zungen , der festsehende und der bewegliche Teil einer Zunge sind jeweils verbunden aber vom Herzstück getrennt.

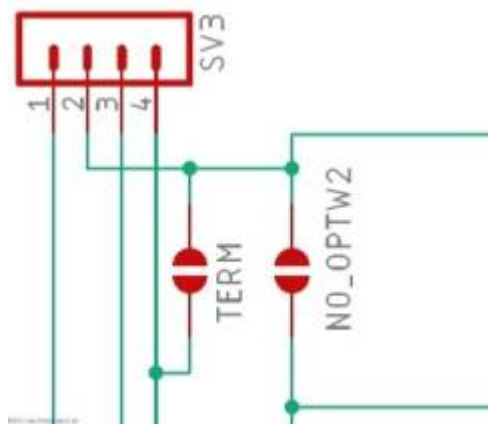
Bemerkung zur Hardware

Die Verbindung TERM auf der Rückseite der Platine muss geschlossen sein damit der nächste Steckplatz funktioniert.

Dieser Lötjumper bleibt nur dann offen, wenn eine zweite Platine neben der ersten über die U-Verbinder SV3 bzw. SV 4 angeschlossen ist. Mit SV3 geht es zur nächsten Platine Eingang SV4. Die Verbindung TERM wird dann nur bei der letzten Platine verbunden.

TERM wird auch geschlossen, wenn die Relais Platinen über einen Wannenstecker der Verteilerplatine angeschlossen werden.

Dies gilt nur mit dieser Modifikation Vers. 1.2. Bei der Version 1.1 hat TERM keine Auswirkung!



Der Pattern Configurator

Anzahl der Ausgabe Kanäle: 2

Bits pro Wert: 1

Die Relaisplatine hat zwei WS2811 für die Relaisansteuerung.

Es werden jeweils zwei Kanäle eines WS2811 für ein Relais verwendet, siehe Schaltplan.

Kanal Rot und Grün des ersten WS2811 für das Relais 1.

Kanal Blau des ersten WS2811 und Kanal Rot des zweiten WS2811 für das Relais 2

Kanal Grün und Kanal Blau des zweiten WS2811 für das Relais 3.

Bei den Startkanälen wird für das erste Relais eine **0** eingetragen.

Startkanal der RGB LED: **0**

Das zweite Relais bekommt eine **2**, weil dies vom blauen Kanal des ersten WS2811 und vom roten Kanal des zweiten WS2811 gesteuert wird.

Das dritte Relais wird vom zweiten- und dritten Kanal des 2. WS2811 gesteuert und bekommt eine **1**. Somit ist hier der Wert nach #Led, **128** gefolgt von **130** und dann **129**.

```
// Activation: Binary Bin_InCh_to_TmpVar(#InCh, 1)
PatternT5(#LED,128,SI_LocalVar,2,0,255,0,0,1500 ms,400 ms,400 ms,1400 ms,400
ms,4,2 ,0,0,63,128,0,63) // Activation: Binary Bin_InCh_to_TmpVar(#InCh, 1)
PatternT5(#LED,130,SI_LocalVar,2,0,255,0,0,1500 ms,400 ms,400 ms,1400 ms,400
ms,4,2 ,0,0,63,128,0,63)
```

```

  \
  // Activation: Binary Bin_InCh_to_TmpVar(#InCh, 1)
  PatternT5(#LED,129,SI_LocalVar,2,0,255,0,0,1500 ms,400 ms,400 ms,1400 ms,400
  ms,4,2 ,0,0,63,128,0,63)
  \
  
```

Die Schaltdauer ist mit 400 ms und die Wartezeit, bevor geschaltet wird, ist hier mit 1500 ms angegeben.

Dies muss für die eigene Weiche angepasst werden, je nach Weg und Geschwindigkeit der Zunge.

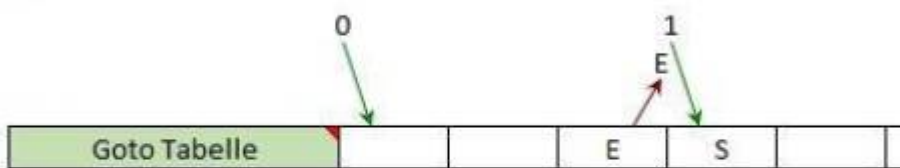
```

#define Herz_1_StCh(LED,StCh) PatternT5(LED,StCh+128,SI_LocalVar,2,0,255,0,0,1500 ms,400 ms,400 ms,1400 ms,400
  
```

Wenn gleiche Zeiten verwendet werden, dann sollten nur die ersten

Dauer	1500 ms	400 ms	400 ms	1400 ms	400 ms
-------	---------	--------	--------	---------	--------

Flash Bedarf: 23 Bytes



D Nr	Spalte Nr ->	1	2	3	4	5
1	Rot	0	255	0	0	0
2	Grün	0	0	0	0	255

Servo Grün | Servo Blau | Herzstückpoli_1 | Herzstückpoli_2

Jedes Makro wird wiederum in den Programm Generator in die gewünschte Zeile, entsprechend der Verkabelung der Verteilerplatte, übertragen.

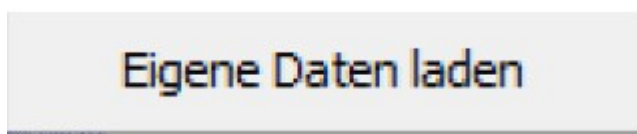
70	3	AnAus	3	Herz_1 für die erste Weiche	// Activation: BinaryBin_InCh_to_TmpVar(#InCh, 1)	6	C1-2
70	4	AnAus	4	Herz_2 für die zweite Weiche	// Activation: BinaryBin_InCh_to_TmpVar(#InCh, 1)	6	C3-4
70	5	AnAus	5	Herz_3 für die dritte Weiche	// Activation: BinaryBin_InCh_to_TmpVar(#InCh, 1)	7	C2-3

An dieser Tabelle erkennt man die unterschiedlichen LED Nummern, hier 6, 6, 7

Übertragen der Dateien

1. Möglichkeit

Die Makros sind in dem File Servo und Herzstück Makros.MLL_pcf gespeichert und können in dem Pfad ... Dokumente, MyPattern_Config_Example geladen werden. Im Pattern Configurator auf den Kreis



..., drücken und mit dieses file integrieren. Dann kann jedes einzelne Makro in den Programm Generator übertragen werden.


2. Möglichkeit

Das File Prog_Gen_Data_Servo Master.MLL_pgf in den Pfad... Dokumente, Prog_Generator_Data



Laden aus Datei

kopieren und mit unter Optionen, Dateien einen neuen Reiter erstellen.



Kopiere von Seite zu Seite

Mit unter Optionen können dann die gewünschten Zeilen des neuen Reiters in ein eigenes Layout übertragen werden.

Autor und Danksagung

Vielen Dank an Hardi für die zahlreichen Information. a.hein 20. Juni 2021

From:

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

Permanent link:

https://wiki.mobaledlib.de/anleitungen/selectrix/servo_und_herzstueck?rev=1624364937

Last update: **2021/06/22 13:28**

