

Ansteuerung der Verbraucher per Tastendruck oder einem Digitalsystem (z.B. Selectrix)

Inhaltsverzeichnis

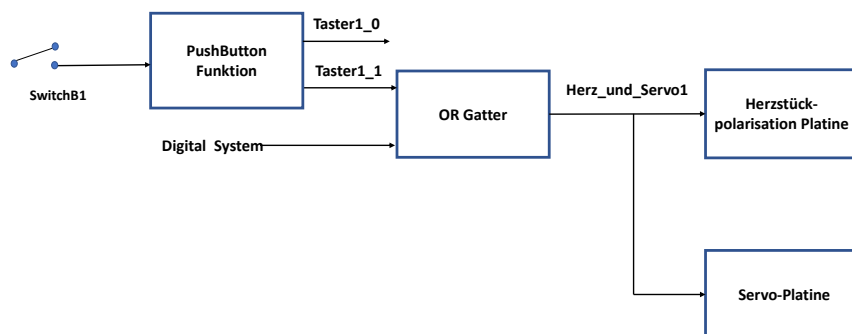
1	Darstellung der PushButton Funktion.....	1
2	Variablen	1
3	Parameter der PushButton Funktion	2
4	Ansteuerung der TastenLED	2
5	Einbindung des Digital Systems.....	3
6	Besonderheiten	4
7	Anwendung für eine Weiche und ein Entkupplungsgeis.....	6
8	Eine Wechselschaltung.....	8

1 Darstellung der PushButton Funktion

Funktionsweise der PushButton Funktion anhand eines Beispiels für die Herzstück-Polarisierung und gleichzeitige Servo-Ansteuerung

Info: Für die PushButton Funktion bitte den Expertenmodus aktivieren

☒ Expertenmodus



SwitchB (B= Board) bedeutet, dass hier ein Taster der PushButton Platine oder ein Taster am Anlagenrand verwendet wird . SwitchB1 für den ersten- ,SwitchB2 für den zweiten Taster usw.

2 Variablen

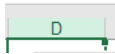
Bei dieser Push Button Aktion werden drei Namen verwendet

- SwitchB1** ist der Name für den ersten-, SwitchB2 für den zweiten Taster usw. SwitchB (B für Board) ist der Name für alle Tasten . Die 1 steht für die erste 2 für die zweite Taste
- Taster_A** Diese Push Button Funktion erzeugt die Variablen Taster_A_0 und Taster_A_1
- Slot_X** Slot_X, hier für 10 "Verbraucher" Slot_1 bis Slot_10, wobei in diesem Beispiel für Slot_1 **Herz_und_Servo1** und für Slot_2 **Herz_und_Servo2** verwendet wird

Wichtig ist, dass die Variablen erst definiert werden müssen, bevor diese verwendet werden.

3 Parameter der PushButton Funktion

Der erste Taster wird in Spalte D (Channel oder Name) der Excel Tabelle **SwitchB1** genannt.



SwitchB1

Die gewählte Funktion in Spalte K : *PushButton_0_1(#InCh, Taster_A_0, 1, 1, 0, 0, 0)* erzeugt zwei Variablen eine 0 und eine 1.

Bemerkung: Es gibt auch Funktionen, die mehrere Variablen erzeugen

Der Name der Variablen vor der 0 bzw.1 steht in der Zeile unterhalb von SwitchB1.

SwitchB1
Taster_A_1

Der Name der Variable ist in diesem Beispiel Taster_A. Als Name könnte auch gleich der Name des Verbrauchers sein. z.B. PushB_Andreaskreuz.

Die Variable *Taster_A_0* ist bei ausgeschalteter Funktion und *Taster_A_1* bei eingeschalteter Funktion **aktiv**.

Beim Auswählen dieser PushButton Funktion werden folgende Parameter eingetragen

Taster_A_0	Zielvariable 1
1	Zustände rotieren [0 / 1]
1	Verwende Zustand 0 beim rotieren [0 / 1]
0	Abschalten durch langes drücken [0 / 1]
0	Optionale Zähler Parameter
0	Abschaltzeit

Einige Erklärungen der Parameter

Parameter **Zielvariable 1**: Name der o.g. gewählte Variablen **Taster_A** mit der Endung **_0**

Parameter Zustände rotieren: Wenn dies auf 1 steht, kann mit einem weiteren Druck auf den Button zum nächsten Zustand gewechselt bzw. weiter geschaltet werden.

4 Ansteuerung der TastenLED

Für die LED neben dem Taster auf der Tasterplatine oder der LED, die in einem Taster eingebaut ist, wird die Routine : *Const(#LED, C1, #InCh, 20, 150)* verwendet.

LED Kanal Auswahl	
<input checked="" type="radio"/> Alle	<input checked="" type="radio"/> 1 / Rot
<input type="radio"/> 1&2 / Gelb	<input type="radio"/> 2 / Grün
<input type="radio"/> 2&3 / Zyan	<input type="radio"/> 3 / Blau
20	Helligkeit wenn deaktiv
150	Helligkeit wenn aktiv
1	LED Kanal

Es wird **ROT** (C1-1) für die erste, **Grün** (C2-2) für die zweite und **Blau** (C3-3) für die dritte TasterLED genommen. Für die vierte TasterLED beginnt es wieder mit **Rot**.(C1-1). Das sind jeweils die drei Kanäle des WS2811 Chips.

Die Helligkeiten können von 0 bis 255 angegeben werden.

Der LED Kanal für die TasterLEDs ist **1**

In diesem Beispiel ist bei Aktivierung die Helligkeit 150 und bei Deaktivierung 20. Somit zeigt die Helligkeit der TasterLED , ob der Verbraucher aktiviert ist.

Beispiel für die ersten 5 Taster der PushButton Platine

SwitchB1			Herz_und_Servo1	1	1	PushButton_0_1(#InCh, Taster_A_0, 1, 1, 0, 0, 0)		
Taster_A_1			Taster LED dauer an			Const(#LED, C1, #InCh, 10, 150)	1-0	C1-1
SwitchB2			Herz_und_Servo2	1	2	PushButton_0_1(#InCh, Taster_B_0, 1, 1, 0, 0, 0)		
Taster_B_1						Const(#LED, C2, #InCh, 10, 150)	1-0	C2-2
SwitchB3			Baustellenlicht	1	3	PushButton_0_1(#InCh, Taster_C_0, 1, 1, 0, 0, 0)		
Taster_C_1						Const(#LED, C3, #InCh, 20, 150)	1-0	C3-3
SwitchB4			Gaslicht	1	4	PushButton_0_1(#InCh, Taster_D_0, 1, 1, 0, 0, 0)		
Taster_D_1						Const(#LED, C1, #InCh, 20, 150)	1-1	C1-1
SwitchB5			RGB Ampel	1	5	PushButton_0_1(#InCh, Taster_E_0, 1, 1, 0, 0, 0)		
Taster_E_1						Const(#LED, C2, #InCh, 20, 150)	1-1	C2-2

5 Einbindung des Digitalsystems

Damit die Verbraucher auch mit dem Digitalsystem aktiviert werden können, wird eine Logic Funktion verwendet.

Info: Für die Logic Funktion bitte den Expertenmodus aktivieren

z.B. **Logic(Slot_3, #InCh OR Taster_C_1)** für

Baustellenlicht
Andreaskreuz
House_Room_warm

Dabei ist die erste Variable der Channel /Name (Spalte D) für die Verbraucheraktion

In diesem Beispiel wurde als dritte Variable **Slot_3** gewählt.

Für die zweite Variable wird der Channel/ Name für die dritte TasterLED, hier TasterC_1 genommen. #InCh steht hierbei als Bezeichner für die Adresse aus der Spalte D und E.

In den Zeilen mit **Herz_und_Servo1** und **Herz_und_Servo2** bzw. **Slot_3 bis Slot_10** werden in den Spalten K die gewünschten Verbraucher letztendlich ausgewählt.

In diesem Beispiel wird bei Selectrix mit der Adresse 10 und mit dem Bit 3 geschaltet.

10	3	AnAus 3	Selectrix/ Taster-03	Logic(Slot_3, #InCh OR Taster_C_1)
----	---	----------------	----------------------	------------------------------------

Adressen des Digitalsystems für alle Verbraucher

10	1	AnAus 1	Selectrix/ Taster-01	Herz_und_Servo1			Logische Verknüpfung	Logic(Herz_und_Servo1, #InCh OR Taster_A_1)
10	2	AnAus 2	Selectrix/ Taster-02	Herz_und_Servo2			Logische Verknüpfung	Logic(Herz_und_Servo2, #InCh OR Taster_B_1)
10	3	AnAus 3	Selectrix/ Taster-03	Herz_und_Servo3			Logische Verknüpfung	Logic(Herz_und_Servo3, #InCh OR Taster_C_1)
10	4	AnAus 4	Selectrix/ Taster-04	Baustelle Andreas Haus			Logische Verknüpfung	Logic(Slot_4, #InCh OR Taster_D_1)
10	5	AnAus 5	Selectrix/ Taster-05	Weiche			Logische Verknüpfung	Logic(Slot_5, #InCh OR Taster_E_1)
10	6	Tast 6	Selectrix/ Taster-06	Entkuppler			Logische Verknüpfung	Logic(Slot_6, #InCh OR Taster_F_1)
10	7	AnAus 7	Selectrix/ Taster-07	Strassenleuchten			Logische Verknüpfung	Logic(Slot_7, #InCh OR Taster_G_1)
10	8	AnAus 8	Selectrix/ Taster-08	MOS			Logische Verknüpfung	Logic(Slot_8, #InCh OR Taster_H_1)
11	1	AnAus 1	Selectrix/ Taster-09	Haus 1 und Haus 2			Logische Verknüpfung	Logic(Slot_9, #InCh OR Taster_I_1)
11	2	AnAus 2	Selectrix/ Taster-10	LED Signal			Logische Verknüpfung	Logic(Slot_10, #InCh OR Taster_J_1)

Sollen **mehrere** Verbraucher mit **einer** Taste- bzw. Digitaleingabe aktiviert werden, wird die Variable **mehrmals** eingetragen.

Slot_3			Baustellenlicht		3	ConstrWarnLightRGB6(#LED, #InCh, 5, 255, 100 ms, 0 ms, 500 ms)	0-4
Slot_3			Andreas		3	AndreaskrRGB(#LED, #InCh)	0-10
Slot_3			House_warm		3	House(#LED, #InCh, 1, 1, ROOM_WARM_W)	0-12

6 Besonderheiten

Herz_und_Servo1			Herz_1 (pc)		1	// Activation: BinaryBin_InCh_to_TmpVar(#InCh, 1)PatternT1(#LED,12	0-1	C1-2
Herz_und_Servo2			Herz_2 (pc)		2	// Activation: BinaryBin_InCh_to_TmpVar(#InCh, 1)PatternT1(#LED,13	0-1	C3-4
Herz_und_Servo1			Servo_rot (pc)		2	// Activation: BinaryBin_InCh_to_TmpVar(#InCh, 1)PatternT1(#LED,28	0-3	C1-1
Herz_und_Servo2			Servo_gruen (pc)		1	// Activation: Binary Bin_InCh_to_TmpVar(#InCh, 1) PatternT1(#LED,29,SI_LocalVar,1,0,255,0,0,10 sek,20,0,200,0 ,0,63,128,63)	0-3	C2-2

Hier wird für eine Herzstück Polarisierung und gleichzeitiger Servoansteuerung eine mit dem Pattern Konfigurator erstellte Funktion verwendet.

https://wiki.mobaledlib.de/anleitungen/selectrix/servo_und_herzstueck#servosteuerung_und_herzstueckpolarisation_mit_selectrix

Zwei Relais werden alternativ geschaltet, verwendbar z.B. für einen Weichenspulenantrieb

Slot_7			Relay 1 und 2 (pc)		7	// Activation: BinaryBin_InCh_to_TmpVar(#InCh, 1)PatternT1	0-32	C1-2
Slot_8			Relay 3 und 4 (pc)		7	// Activation: BinaryBin_InCh_to_TmpVar(#InCh, 1)PatternT1	0-32	C3-4
Slot_9			Relay 5 und 6 (pc)		7	// Activation: BinaryBin_InCh_to_TmpVar(#InCh, 1)PatternT1	0-33	C2-3

Mit der **Const** Funktion kann auch ein einzelnes Relais geschaltet werden.

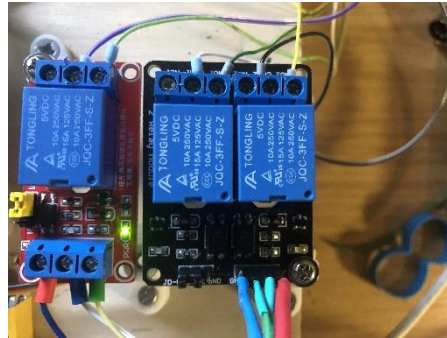
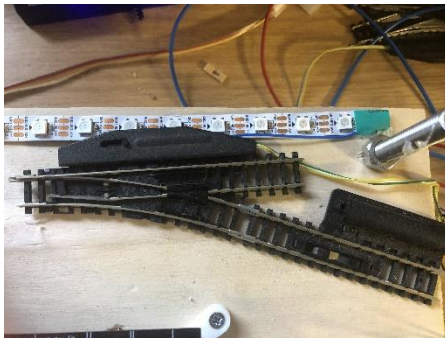
Slot_10			Relais 7		7	Const(#LED, C1, #InCh, 0, 255)
---------	--	--	----------	--	---	--------------------------------

Mit dieser Programmierung können die Verbraucher entweder mit dem Digitalsystem **ODER** mit den Tastern aktiviert werden. Wird ein Verbraucher mit der Taste aktiviert, kann das Digitalsystem diesen leider **nicht** wieder ausschalten.

7 Anwendung für eine Weiche und ein Entkupplungsgeis

Mit der Push Button Platine soll eine Weiche und ein Entkupplungsgeis geschaltet werden

Zur Ansteuerung werden s.g. Channel Relay Module verwendet



Zur Weichenansteuerung wird ein **Zwei Channel Relay Module** zum Umschalten der Weiche, und für das Entkupplungsgeis wird ein **Ein Channel Relay Module** verwendet.

Dabei wird bei jedem Relais der Schließer, bezeichnet als NO (normally open) benutzt.

Ein WS2811 mit den drei Ausgängen R G B steuert die Module.

Die Ausgänge R G des WS2811 gehen jeweils an den IN1 bzw. IN2 Eingang des **Zwei Channel Relay Modules**. Der dritte Ausgang B des WS2811 wird mit dem Eingang IN des **Ein Channel Relay Modules** verbunden.

An die Mittenkontakte der Relais wird eine Leitung der Versorgungsspannung angeschlossen. Die zweite Leitung der Versorgungsspannung geht direkt an die Weiche bzw. an das Entkupplungsgeis.

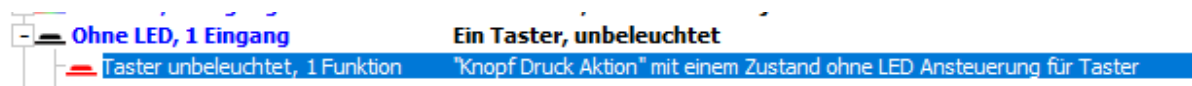
Bei Minitrix wird das weiße Kabel mit den Mittenkontakten verbunden. Bei Märklin das gelbe Kabel.

Die Schaltleitungen der Weiche, bei Minitrix das grüne und das gelbe Kabel, werden jeweils an den Schließer NO (normally open) des Relais angeschlossen. Bei Märklin die blauen Kabel.

Mit dieser Push Button Platine Routine wird zuerst der Taster bestimmt.

SwitchB5		Weiche	1	4	Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, Taster_E_0, 1, 1, 0, 0, 0)
Taster_E_1					LED einstellbar	Const(#LED, C2, #InCh, 20, 150)

Wahl der Push Button Funktion hier: *PushButton_0_1(#InCh, Taster_E_0, 1, 1, 0, 0, 0)*

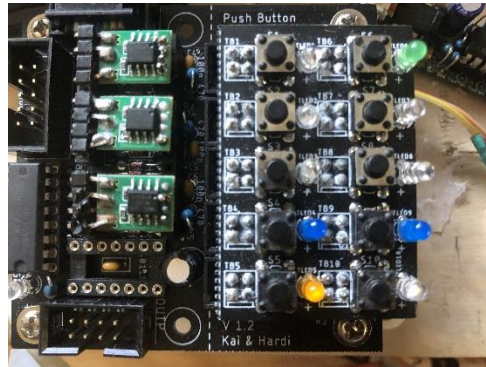
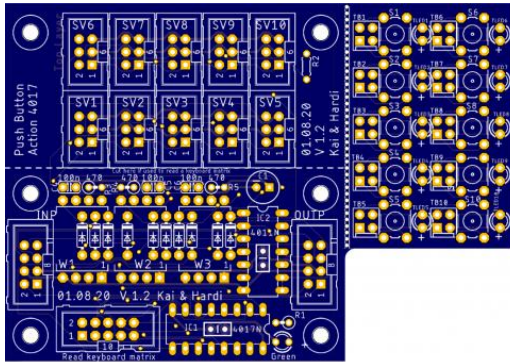


Taster_E_0	Zielvariable 1
1	Zustände rotieren [0 / 1]
1	Verwende Zustand 0 beim rotieren [0 / 1]
0	Abschalten durch langes drücken [0 / 1]
0	Optionale Zähler Parameter
0	Abschaltzeit

Zu beachten ist hier die Zielvariable , diese ist hier **Taster_E_0**

Mit dieser Push Button Funktion wird die kleine Huckepack Platine für Tastern und LEDs verwendet.

Diese kleine Platine ist ein Anhängsel der 300 PushButton Platine. Im Bild rechte Seite



Bei Verwendung von z.B. Tastern mit RGBLEDs muss eine andere PushButton Routine angewandt werden.

Damit die Relais auch mit dem Digitalsystem aktiviert werden sollen gibt es eine

Logische Verknüpfung für das Digitalsystem

10	5	AnAus	5	Selectrix/ Taster-05	Weiche		1	Logische Verknüpfung	Logic(Slot_5, #InCh OR Taster_E_1)
10	6	Tast		Selectrix/ Taster-06	Entkuppler		1	Logische Verknüpfung	Logic(Slot_6, #InCh OR Taster_F_1)

Das Entkupplungsgeis wird als Taster definiert, damit mit jedem Tastendruck die Aktion ausgeführt wird.

Und zum Schluss die Verbraucher hier eben die Weiche und das Entkupplungsgeis.

Slot_5			Weiche		4		Muster Pattern_Configurator	// Activation: BinaryBin_InCh to TmpVar(#InCh, 1)
Slot_6			Entkuppler		4		Mono-Flop	MonoFlop(Entkuppler, #InCh, 500)
Entkuppler							LED einstellbar	Const(#LED, C3, #InCh, 0, 255)

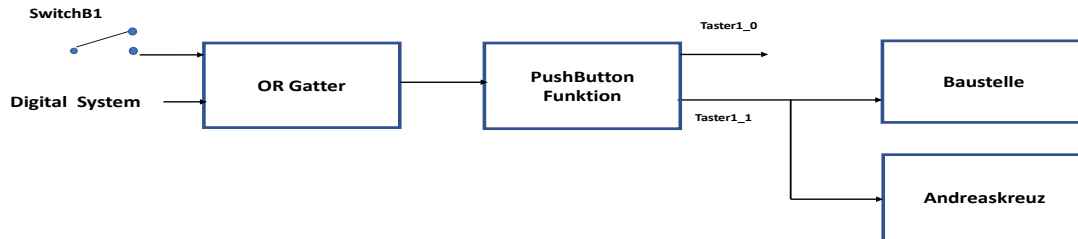
Die Ansteuerung der Weiche wurde ein im Pattern Generator erstelle Funktion übernommen

Zur Ansteuerung des Entkupplungsgeis wird ein Mono Flop mit einer selbst gewählten Variablen *Entkuppler* und eine *Const LED-Funktion* verwendet.

8 Eine Wechselschaltung

Mit der „Wechselschaltung“ kann man den Verbraucher per **Taster einschalten** und dann mit dem Digitalsystem wieder **ausschalten** und **umgekehrt**, wobei das Bit im Selectrix System **leider** gesetzt bleibt. Es bleibt aber die Frage, ist eine Wechselschaltung notwendig. Reicht es nicht die Verbraucher per Software oder eben per Taster ein und auszuschalten.

Bei dieser sog. Wechselschaltung wird die PushButton Funktion vom OR Gate aktiviert.



Mit der Wechselschaltung kann man per **Taster einschalten** und dann mit dem Digitalsystem wieder **ausschalten** und **umgekehrt**, wobei das Bit im Selectrix System **leider** gesetzt bleibt.

Das ist nicht so schön, weil auch dann in der Modellbahn Software gezeigt wird, dass dieser Verbraucher noch aktiv ist, obwohl dieser per Taster schon ausgeschaltet wurde.

Für eine optimale Lösung **müsste** die MLL den Zustand auf den Bus zurückschreiben können. Dies ist derzeit noch nicht in der MLL integriert.

Die am Anfang beschriebene Funktionen für die Wechselschaltung neu angepasst

Für die „Wechselschalter“ Ansteuerung wurde beim Schaltertyp anstatt AnAus- **AnAus 3** die Tast- **Tast** Funktion gewählt. Dadurch kann man mit der MLL Taste den Servo in die eine Richtung und mit dem Digitalsystem in die andere Richtung bewegen. Das Selectrix Digital Bit kann mit der MLL Taste **nicht** geändert werden, somit würde innerhalb der Modellbahnsoftware dann z.B. bei einer Weiche **falsch** angezeigt werden.

Hier die **Logische Verknüpfung** $Sx_or_Tasterx\ OR\ SwitchBx$ (x) steht für die jeweiligen Nummern

10	1	Tast		Selectrix/ Taster-01	Herz_und_Servo1			Logische Verknüpfung	Logic(Sx_or_Taster1, #InCh OR SwitchB1)
10	2	Tast		Selectrix/ Taster-02	Herz_und_Servo2			Logische Verknüpfung	Logic(Sx_or_Taster2, #InCh OR SwitchB2)
10	3	Tast		Selectrix/ Taster-03	Herz_und_Servo3			Logische Verknüpfung	Logic(Sx_or_Taster3, #InCh OR SwitchB3)
10	4	Tast		Selectrix/ Taster-04	Baustelle Andreas Haus			Logische Verknüpfung	Logic(SX_or_Taster4, #InCh OR SwitchB4)
10	5	Tast		Selectrix/ Taster-05	Weiche			Logische Verknüpfung	Logic(SX_or_Taster5, #InCh OR SwitchB5)
10	6	Tast		Selectrix/ Taster-06	Entkuppler			Logische Verknüpfung	Logic(SX_or_Taster6, #InCh OR SwitchB6)
10	7	Tast		Selectrix/ Taster-07	Strassenleuchten			Logische Verknüpfung	Logic(SX_or_Taster7, #InCh OR SwitchB7)
10	8	Tast		Selectrix/ Taster-08	MOS			Logische Verknüpfung	Logic(SX_or_Taster8, #InCh OR SwitchB8)
11	1	Tast		Selectrix/ Taster-09	Haus 1 und Haus 2			Logische Verknüpfung	Logic(SX_or_Taster9, #InCh OR SwitchB9)
11	2	Tast		Selectrix/ Taster-10	Mix			Logische Verknüpfung	Logic(SX_or_Taster10, #InCh OR SwitchB10)

Push Button Funktion mit separater LED

SwitchB1										LED einstellbar	Const(#LED, C1, #InCh, 10, 255)
Sx_or_Taster1			Herz_und_Servo1	Wanne 1 und 2	1	2				Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, Herz_und_Servo1_0
SwitchB2										LED einstellbar	Const(#LED, C2, #InCh, 10, 150)
Sx_or_Taster2			Herz_und_Servo2	Wanne 1 und 2	1	2				Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, Herz_und_Servo2_0
SwitchB3										LED einstellbar	Const(#LED, C3, #InCh, 20, 150)
Sx_or_Taster3			Herz_und_Servo3	Wanne 1 und 2	1	2				Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, Herz_und_Servo3_0

SwitchB4										LED einstellbar	Const(#LED, C1, #InCh, 10, 150)
SX_or_Taster4			Baustelle, Andreas, House	Wanne 3 Taster 4	1	3				Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, Baustelle1_0, 1, 1,
SwitchB5										LED einstellbar	Const(#LED, C2, #InCh, 10, 150)
SX_or_Taster5			Weiche	Wanne 4 Taster 5	1	4				Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, Weiche1_0, 1, 1, 0,
SwitchB6										LED einstellbar	Const(#LED, C3, #InCh, 30, 150)
SX_or_Taster6			Entkuppelgleis	Wanne 4 Taster 6	1	4				Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, Entkupplungsgleis1_0
SwitchB7										LED einstellbar	Const(#LED, C1, #InCh, 30, 150)
SX_or_Taster7			Strassenbeleuchtung	Wanne 5 Taster 7	1	5				Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, Strassenleuchten1_0,
SwitchB8										LED einstellbar	Const(#LED, C2, #InCh, 10, 150)
SX_or_Taster8			MOS	Wanne 6 Taster 8	1	6				Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, MOS1_0, 1, 1, 0, 0,
SwitchB9										LED einstellbar	Const(#LED, C3, #InCh, 1, 250)
SX_or_Taster9			Haus 1 und Haus 2	Wanne 7 Taster 9	1	7				Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, Kleine Haeuser1_0, 1
SwitchB10										LED einstellbar	Const(#LED, C1, #InCh, 10, 150)
SX_or_Taster10			Fire , Ampel, Kamin, Gaslight	Wanne 8 Taster 10	1	8				Taster unbeleuchtet, 1 Funk	PushButton_0_1(#InCh, MIX1_0, 1, 1, 0, 0,

Verbraucher , hier die Herzstück Polarisation und die Servo Ansteuerung

Herz_und_Servo1_1			Herz _1 (pc)		1					Muster Pattern_Configurator	// Activation: BinaryBin_InCh_to_TmpVar
Herz_und_Servo2_1			Herz _2 (pc)	Taster	1					Muster Pattern_Configurator	// Activation: BinaryBin_InCh_to_TmpVar
Herz_und_Servo3_1			Herz _3 (pc)	Taster	1					Muster Pattern_Configurator	// Activation: BinaryBin_InCh_to_TmpVar
Herz_und_Servo1_1			Servo_rot (pc)		2					Muster Pattern_Configurator	// Activation: BinaryBin_InCh_to_TmpVar
Herz_und_Servo2_1			Servo_gruen (pc)	Taster	2					Muster Pattern_Configurator	// Activation: BinaryBin_InCh_to_TmpVar
Herz_und_Servo3_1			Servo_blau (pc)	Taster	2					Muster Pattern_Configurator	// Activation: BinaryBin_InCh_to_TmpVar

Baustelle1_1			Baustellenlicht	Taster 4	3					Baustellenlicht 6x	ConstrWarnLightRGB6(#LED, #InCh, 5, 255, 1
Baustelle1_1			Andreas	Taster 4	3					Andreaskreuz RGB	AndreaskrRGB(#LED, #InCh)
Baustelle1_1			House_warm	Taster 4	3					Belebtes Haus	House(#LED, #InCh, 1, 1, ROOM_WARM_W)
Weiche1_1			Weiche	Taster 5	4					Muster Pattern_Configurator	// Activation: BinaryBin_InCh_to_TmpVar
Entkupplungsgleis1_1			Entkuppler	Taster 6	4					Mono-Flop	MonoFlop(Entkupplungsgleis, #InCh, 300)
Strassenleuchten1_1			Strassenleuchten	Taster 7	5					Straßenbeleuchtung	GasLights(#LED, #InCh, NEON_LIGHT1D, NEON
MOS1_1			MOS	Taster 8	6					Straßenbeleuchtung	GasLights(#LED, #InCh, GAS_LIGHT1, GAS_LIG
Kleine_Haeuser1_1			Kleine Häuser	Taster 9	7					Belebtes Haus	HouseT(#LED, #InCh, 4, 6, 1, 3, ROOM_WARM
MIX1_1			Feuer	Taster 10	8					Feuer	Fire(#LED, #InCh, 11, 100)
MIX1_1			Ampel	Taster 10	9					Ampel RGB Soft	RGB_AmpelXFade(#LED, #InCh)
MIX1_1			Kamin	Taster 10	9					Belebtes Haus	House(#LED, #InCh, 3, 3, ROOM_CHIMNEY, ROO
MIX1_1			Gaslichter	Taster 10	10					Straßenbeleuchtung	GasLights(#LED, #InCh, GAS_LIGHT, GAS_LIGH

Hinweis: es dürfen keine Umlaute verwendet werden. z.B. **Haeser** anstatt **Häuser**.

Zur Verdeutlichung wurden in der PB Funktion die gleichen Namen wie bei den Verbrauchern verwendet.

Die verwendeten Excel-Dateien sind in GitHub verfügbar.

Vielen Dank an Hardi und Dominik für die Hilfe bei der Programmierung und Beschreibung

a.hein 21. Nov.. 2021