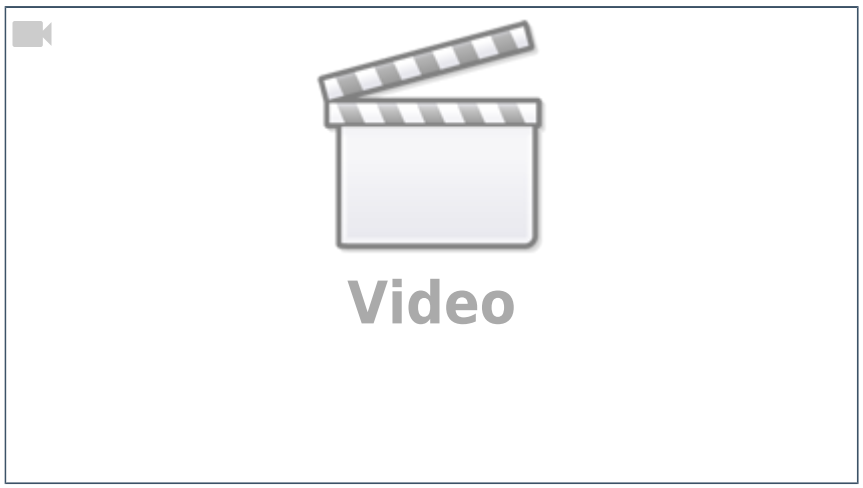
Farbwechsel und Regenbogen zur dekorativen Beleuchtung besonderer Objekte

Diese Anleitung widmet sich vollumfänglich der Farbwechsel-Funktion, wie wir sie aus zahlreichen Lampen für den Heimgebrauch kennen. Im Gegensatz zu Produkten aus dem heimischen Wohnzimmer wird in dieser Anleitung auf die oftmals verwendete siebte Farbe Weiß im Wechsel verzichtet. Das Hinzufügen dieser Farbe ist allerdings ein Kinderspiel.

Im Gegensatz zu Hardis Lamborghini habe ich den Farbwechsel ausschließlich mit RGB-Werten statt mit HSV-Werten aufgebaut. Anfangs war mir die HSV-Funktion im Pattern Configurator nicht bekannt, also half ich mir mit den RGB-Werten. Im Nachgang stellte sich diese Herangehensweise als flexibler in Bezug auf die Helligkeiten der jeweiligen Farben dar, mit denen ich nach dem kleinen Einführungsvideo starten werde. Im ersten Video ist meinFaller Martinstor zu sehen, das zum Zeitpunkt der Aufnahme noch mit identischen Helligkeiten über alle sechs Farben programmiert war. Viel Spaß:



Individuelle Helligkeitswerte

Beim Farbwechsel gibt es zahlreiche Dinge zu beachten, damit das beleuchtete Objekt standesgemäß in Szene gesetzt wird. Es fängt im Pattern Configurator bei der Eingabe von Zeit und Helligkeitswerten an (Bits pro Wert), geht mit den individuellen Helligkeitswerten jeder einzelnen Farbe weiter und am Ende soll das Ganze auch noch richtig geschaltet werden. Wer das Ganze noch für die Verwendung auf dem Arduino speicheroptimiert anlegen will, bekommt hier die entsprechende Hilfe. Keine Angst: Mit der folgenden Anleitung wird der Farbwechsel zum Kinderspiel.

Fangen wir mit den individuellen Helligkeitswerten jeder einzelnen Farbe an. Von den drei in einem WS2812 verbauten Chips ist Rot der hellste und Blau der dunkelste. Lässt man nur die drei Grundfarben wechseln, fällt schnell auf, dass Blau kaum wahrnehmbar ist. Kommen die drei Mischfarben Cyan, Magenta und Gelb mit ins Spiel, wird es zunehmend komplizierter, da bei den Mischfarben immer zwei Chips gleichzeitig aktiv sind. Sichtbar wird das insbesondere bei direkt beleuchteten Objekten (z. B. eine Fassade, die mit Flutlichtstrahlern direkt angestrahlt wird). Strahlen alle Farben mit jeweils 100%, erhält man nicht nur einen Farbwechsel, sondern vor allem einen Helligkeitswechsel.

Um eine gleichbleibende Helligkeit über alle sechs Farben (Rot, Gelb, Grün, Cyan, Blau, Magenta) zu erhalten, ist es daher empfehlenswert, jede Farbe individuell in ihrer Helligkeit anzupassen.

Wie viele Bits pro Wert?

Da im Speziellen Blau bis zu sechs Mal so hell sein muss wie beispielsweise Cyan, hat diese individuelle Einstellung einen unmittelbaren Einfluss auf die minimalen Helligkeitswerte (Bits pro Wert) im Pattern Configurator. Werden beispielsweise für Cyan zu wenig Helligkeitswerte eingestellt, kommt es beim langsamen Farbwechsel trotz analogem Überblenden zu sichtbaren Abstufungen. Je mehr Helligkeitswerte vorgegeben werden und je kürzer der Wechsel von einer auf die andere Farbe, desto weicher wird der Übergang zur nächsten Farbe.

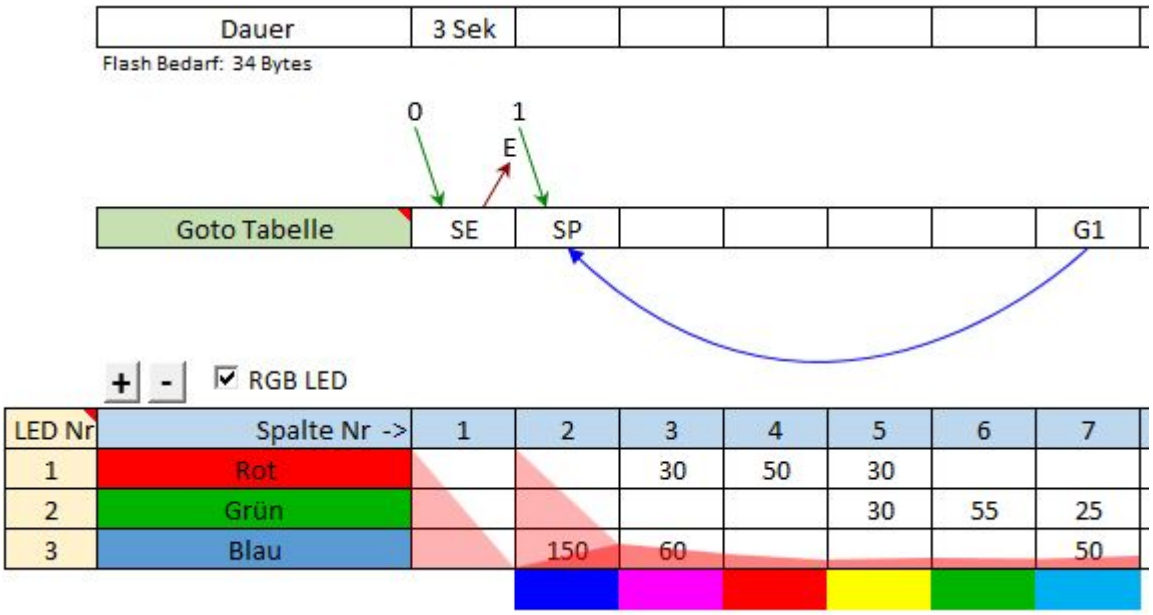
Die nächste Herausforderung wird das Reduzieren der gesamten Helligkeit. Sind die LEDs insgesamt zu hell, müssen die individuellen Helligkeitswerte aller Farben gleichmäßig reduziert werden, da sich die maximale Helligkeit im Pattern Configurator immer nur auf den Maximalausschlag bezieht und keinen prozentualen Einfluss auf niedrige Werte als 100% hat.

Wird beispielsweise mit 6 Bits pro Wert gearbeitet, stehen 64 Helligkeitswerte zur Verfügung. Soll hier Blau beispielsweise auf 60% seiner Helligkeit gedimmt werden, so ergibt das 38 Helligkeitswerte für Blau. Für Cyan benötigt man nun Blau mit einem Drittel und Rot mit einem Sechstel der Helligkeit von Blau, also ca. 6 Helligkeitswerte. Diese sechs Abstufungen nimmt das Auge beim Wechsel wahr.

Etwas besser sieht es bei 7 Bits pro Wert aus. Hier stehen 128 Helligkeitswerte zur Verfügung. Bei 60% Helligkeit für Blau entspricht das 76 Werten. Cyan würde hier mit ca. 13 Werten je Farbe dargestellt, was beim direkten Beleuchten von Fassaden immer noch zu sichtbaren Abstufungen führt.

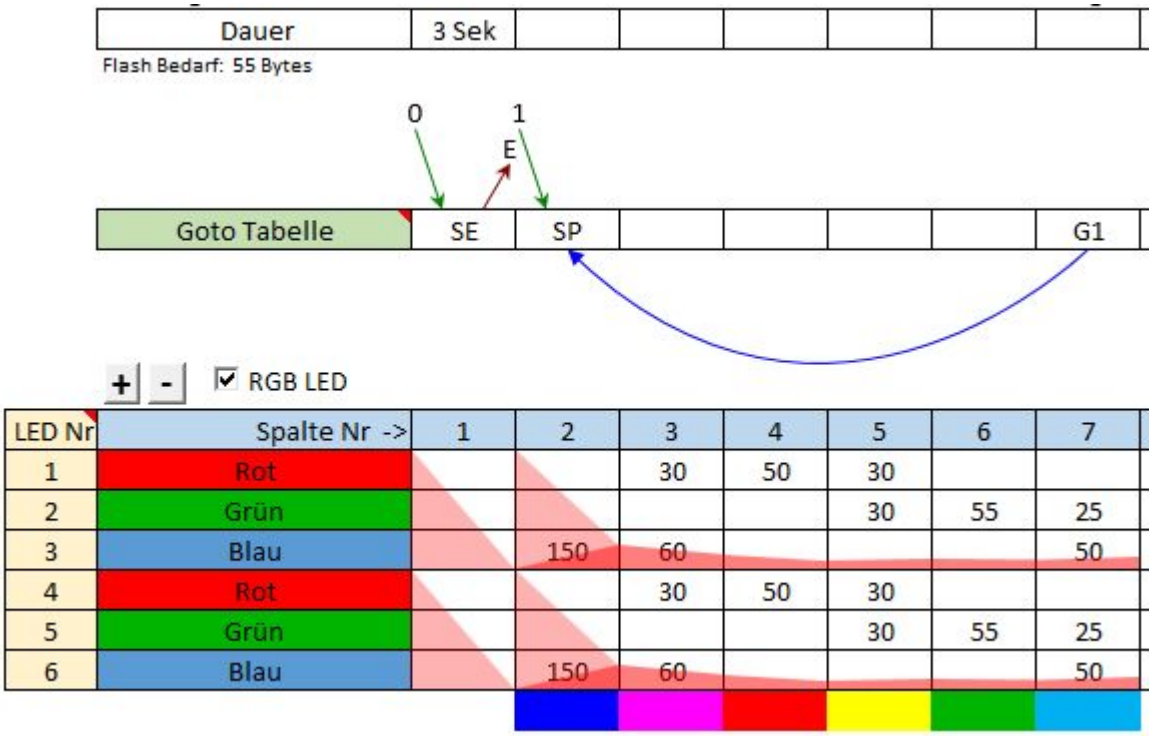
In diesem Fall erweist es sich also als sinnvoll, mit den maximalen **8 Bits pro Wert** ins Rennen zu gehen. Somit verfügt ein auf 60% gedimmtes Blau über 150 Abstufungen und Cyan immer noch über ca. 25 Abstufungen. Bei der direkten Beleuchtung von Objekten nimmt das Auge in diesem Fall bis zu einer Zeit von ca. 5 Sekunden je Farbe keine Abstufungen mehr wahr.

Im Folgenden ist ein Beispiel zu sehen, bei dem die Helligkeitswerte der einzelnen Farben visuell angepasst wurden:



Speicher sparen mit dem "LED-Werte kopieren-Befehl"

Solange man das Objekt mit nur einem einzigen Strahler beleuchtet, muss man wohl oder übel mit dem Speicherbedarf im Pattern Configurator leben um ein überzeugendes Ergebnis zu erzielen. Ab dem zweiten Strahler kann man sich jedoch eines einfachen Tricks bedienen. Mithilfe des LED-Werte kopieren-Befehls können alle folgenden LEDs speichersparend betrieben werden. Wie im folgenden Beispiel zu sehen, erhöht sich der Speicherbedarf um 21 Bytes beim Hinzufügen einer zweiten Farbwechsel-LED. Diesen und weiteren Speicher kann man mit diesem Trick sparen.



Darf's ein bisschen mehr sein?

Nun gibt es Objekte auf der Modellbahn, die eine Ausleuchtung mit deutlich mehr als zwei oder drei

Flutlichtstrahlern erlauben. Das können Kirchen, Kloster, Stadtmauern oder Viadukte sein (oder Burgen). Die einfachste Beleuchtung wäre oben gezeigtes Beispiel mit entsprechenden Copy-LED-Befehlen. Doch bei mehr als drei bis vier Strahlern kann man die wechselnden Farben alternativ von links nach rechts, von innen nach außen oder jeweils umgekehrt durchlaufen lassen.

Die Idee zu diesem Effekt entstand am 6. August 2021 bei einem Besuch Hamburgs. Bei einem der abendlichen Spaziergänge gingen wir am Amerikanischen Generalkonsulat in Hamburg vorbei. Mein damals 9-jähriger Sohn war von diesem Effekt derart begeistert, dass ich ihm auf der Stelle versprach, diesen Effekt gemeinsam mit ihm im Modell nachzubilden. Unser Foto entstand gegen 20:30 Uhr. Wie das Generalkonsulat bei völliger Dunkelheit aussieht, zeigt Queer.de auf einem Bild des Tages.

Beim Generalkonsulat sorgen Strahler in sechs (vermutlich 6×2) Reihen für einen durchlaufenden Farbwechsel. Die sechs Grundfarben Rot, Gelb, Grün, Cyan, Blau und Magenta laufen dabei sehr langsam von links nach rechts durch. Der Effekt fließt dabei so langsam, dass man es kaum wahrnimmt.

Das Modell

Lasst mich zunächst ein wenig auf die Entwicklung des Modells eingehen, bevor ich aufzeige, wie man das Beispiel entsprechend programmiert.

In mehr als vier Wochen Bauzeit entstand im Anschluss die Burg Falkenstein von Kibri (#39010). Mit einer Kantenlänge von gut 50cm und 3,5cm Abstand zwischen den Strahlern kommt die Burg auf beachtliche 15 Strahler. Die Anzahl von 15 Strahlern ist der Entwicklung geschuldet. Bei der ersten Umsetzung montierten wir sieben Flutlichter, um am ersten und am letzten Strahler jeweils dieselbe Farbe zu erzeugen (das hatte uns am Amerikanischen Generalkonsulat gestört). Es sollte wirken, als ob die Farbe, die rechts verschwindet links wieder reinkommt. In der Dunkelheit funktionierte der Effekt gut, doch mit wachsendem Umgebungslicht wurden aus dem Verlauf einzelne Flecken (eben wie beim Vorbild).

Somit war klar, dass wir die Anzahl auf 13 erhöhen mussten und den Regenbogen zweimal hintereinander anlegen mussten (Rot > Gelb > Grün > Cyan > Blau > Magenta > Rot > Gelb > Grün > Cyan > Blau > Magenta > Rot). Bei längerem Betrachten fiel dann auf, dass jeweils die Farbe, die links, rechts und in der Mitte zu sehen war, unterschiedlich große Flecken erzeugte. Links war der Fleck am größten, in der Mitte war er bedingt durch einen Mauervorsprung zu klein und rechts war die Burg nicht bis an den Rand ausgeleuchtet. Es folgten zwei weitere Strahler, die jeweils die selbe Farbe imitieren wie ihre jeweiligen Nachbarn. Da wir alternativ auch einen klassischen Farbwechsel abbilden wollten, belegte das Muster im Pattern Configurator zu diesem Zeitpunkt weit über ein Kilobyte.

Doch so ganz wollte sich der WOW-Effekt nicht einstellen. Der doppelte Farbwechsel war einfach zu viel und für das Auge schwer zu verfolgen. Einfach jeweils zwei Strahler dieselbe Farbe darstellen zu lassen (wie im Vorbild) erschien uns als Verschwendung der 15 in Handarbeit gefertigten RGB-Strahler und hätte aus heutiger Erfahrung auch wieder mehr Speicher belegt. Also wechselten wir im Pattern Configurator die Richtung der rechten acht Strahler, sodass sich die an beiden Seiten startende Farbe am Ende oben am Hügel traf. Für unseren Geschmack war das der Durchbruch, lediglich die Richtung war falsch. Schließlich läuft Wasser den Berg runter und nicht hoch. Das Ergebnis ist im Video zu sehen. Die erste Farbe startet auf dem Hügel in der Mitte und „fließt“ gleichmäßig zu beiden Seiten den Berg hinunter.

Wie funktioniert das?

Beim Anlegen eines Musters im Pattern Configurator muss man sich zwischen der Aktivierung über N_Buttons und Binary entscheiden. Die Aktivierung über N_Buttons ist eleganter, da man über Tasten einfach von einem zum nächsten Effekt wechseln kann (z. B. von aktivem Regenbogen zum Farbwechsel oder umgekehrt). Bei einer Aktivierung über Binary muss ein aktiver Regenbogen per DCC-Adresse zunächst deaktiviert werden, bevor man den Farbwechsel aktiviert. Das kann zu Fehlbedienungen führen. Welche Aktivierung man hier wählt, hängt allerdings vom verwendeten

Digital-System ab. Mit der CS von Märklin wird man hier die Aktivierung über die Tasten rot/grün (N_Buttons) wählen. Bei Verwendung einer Z21 benötigt man die Aktivierung über Schalter an/aus (Binary).

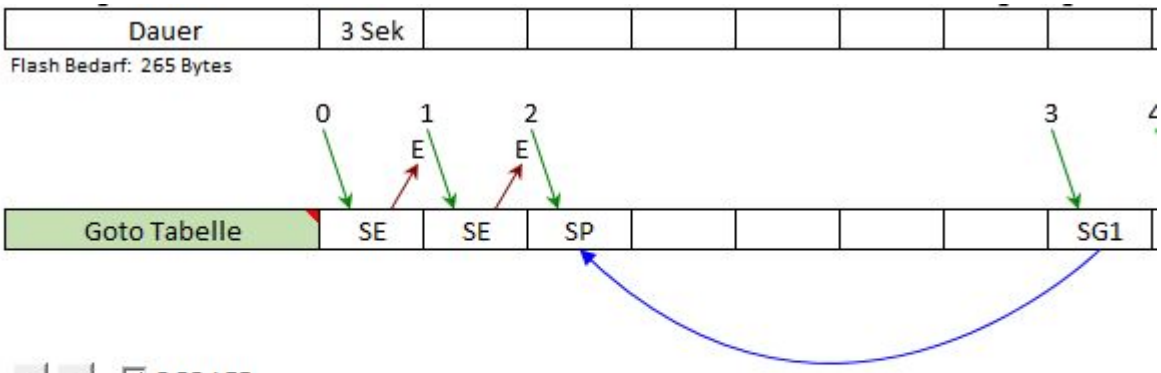
Bei einer Aktivierung über Binary benötigt man vier Zustände im Pattern Configurator:
1. Aus
2. Weiß
3. Regenbogen
4. Farbwechsel

Mit einer Goto-Tabelle lassen sich diese vier Zustände einfach in einem Pattern abbilden. Regenbogen und Farbwechsel sollen sich unabhängig vom eingeschalteten Weiß schalten lassen. Das ist nicht ganz so trivial, wie es klingt. Das erfordert jeweils zwei Goto-Sprünge für den Regenbogen und für den Farbwechsel. Die Goto-Befehle für den Regenbogen und den Farbwechsel lauten also „SP“ und „SG1“ sowie „SP“ und „SG2“. Je nachdem, ob das weiße Licht über Goto-1 aktiviert oder über Goto-0 deaktiviert ist, starten Regenbogen und Farbwechsel nun über SP oder über SGx und laufen ab da in Dauerschleife.

Dabei wenden wir gleich das oben gelernte an und stellen trotz 15 Flutlichtern nur die ersten sechs RGB-LEDs im Pattern Configurator dar. Das hat in unserem Fall den Speicherbedarf von ca. 1250 Bytes auf 265 Bytes reduziert.

Der Regenbogen

Die oben gezeigte Programmierung wird nun sechsmal untereinander kopiert, dabei aber jeweils um eine Spalte nach links verschoben. Der um eine Spalte verschobene Inhalt kommt nach hinten. Mit den Zuständen „Aus“ und „Weiß“ sieht das Ganze dann so aus:



\+ - ☑ RGB LED

LED Nr	Spalte Nr ->	1	2	3	4	5	6	7	8
1	Rot		15		30	50	30		
2	Grün		13				30	55	25
3	Blau		8	150	60				50
4	Rot		15	30	50	30			
5	Grün		13			30	55	25	
6	Blau		8	60				50	150
7	Rot		15	50	30				30
8	Grün		13		30	55	25		
9	Blau		8				50	150	60
10	Rot		15	30				30	50
11	Grün		13	30	55	25			
12	Blau		8			50	150	60	
13	Rot		15				30	50	30
14	Grün		13	55	25				30
15	Blau		8		50	150	60		
16	Rot		15			30	50	30	
17	Grün		13	25				30	55
18	Blau		8	50	150	60			

Der Farbwechsel

Streng genommen benötigt man den für den klassischen Farbwechsel nur ein Pattern Muster (wie oben gezeigt). Da wir aber schon den Regenbogen mit Copy-LED-Befehlen abbilden, ist es einfacher, den Farbwechsel für die sechs LEDs im gleichen Pattern abzubilden. Dazu wird das Muster der ersten LED einfach nach rechts und von dort auf alle sechs LEDs kopiert. Die komplette Programmierung sieht dann so aus:

Dauer	3 Sek													
Flash Bedarf: 265 Bytes														
	0	1 E	2 E					3	4					5
Goto Tabelle	SE	SE	SP					SG1	SP					SG2

+ - ☑ RGB LED

LED Nr	Spalte Nr ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Rot		15		30	50	30				30	55	30		
2	Grün		13				30	55	25				30	55	25
3	Blau		8	150	60				50	150	60				50
4	Rot		15	30	50	30					30	55	30		
5	Grün		13			30	55	25					30	55	25
6	Blau		8	60				50	150	150	60				50
7	Rot		15	50	30				30		30	55	30		
8	Grün		13		30	55	25						30	55	25
9	Blau		8				50	150	60	150	60				50
10	Rot		15	30				30	50		30	55	30		
11	Grün		13	30	55	25							30	55	25
12	Blau		8			50	150	60		150	60				50
13	Rot		15				30	50	30		30	55	30		
14	Grün		13	55	25				30				30	55	25
15	Blau		8		50	150	60			150	60				50
16	Rot		15			30	50	30			30	55	30		
17	Grün		13	25				30	55				30	55	25
18	Blau		8	50	150	60				150	60				50

Copy => Paste

Nun geht es darum, die sechs Farbwechsel auf die verbleibenden neun LEDs zu kopieren. Hier ist Konzentration gefragt.

Bei mir startet der Regenbogen mit der LED Nummer 55, folglich ist die sechste in der Kette Nummer 60. Nun folgt das Pärchen in der Mitte (Flutlicht 7 & 8), das immer dieselbe Farbe wie die erste LED (also Nummer 55) haben muss. Da der Regenbogen ab hier in die andere Richtung fließt, muss ab LED 63 rückwärts kopiert werden. Es folgen also die LEDs 60, 59, 58, 57, 56 und 55. Letztere wird zweimal kopiert, weil der einzelne Fleck rechts an der Burg zu klein war (siehe Einleitung).

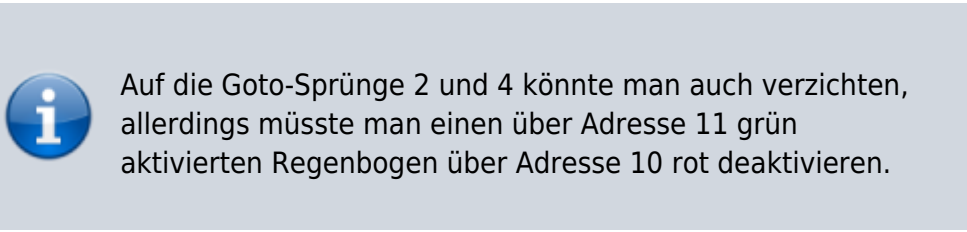
An dieser Stelle wird auch klar, wie die Goto-Sprünge 0/1, 2/3 und 4/5 über die DCC-Adressen bzw. über die Buttons rot/grün aktiviert werden.

✓		100	AnAus	0	Burgmauer weiß (Goto 0+1)	V08	2		Logische Verknüpfung	Logic(Rainbow0, #InCh)		
✓		11	AnAus	0	Burgmauer Regenbogen (Goto 2+3)	V08	2		Logische Verknüpfung	Logic(Rainbow1, #InCh)		
✓		12	AnAus	0	Burgmauer Farbwechsel (Goto 4+5)	V08	2		Logische Verknüpfung	Logic(Rainbow2, #InCh)		
		10	Rot						Logische Verknüpfung	Logic(R1, #InCh)		
		10	Grün		Aktivierung über N-Buttons für CS2/3				Logische Verknüpfung	Logic(R2, #InCh)		
		11	Rot						Logische Verknüpfung	Logic(R3, #InCh)		
		11	Grün						Logische Verknüpfung	Logic(R4, #InCh)		
		12	Rot						Logische Verknüpfung	Logic(R5, #InCh)		
		12	Grün						Logische Verknüpfung	Logic(R6, #InCh)		
✓		Rainbow0	AnAus		Weiß, Regenbogen und Farbwechsel	V08	2		Muster Pattern_Configurator	// Activation: BinaryBin_I	0-55	
✓					Flutlicht 7	V08	2		LED-Werte kopieren	CopyLED(#LED, #InCh, 55)	0-61	
✓					Flutlicht 8	V08	2		LED-Werte kopieren	CopyLED(#LED, #InCh, 55)	0-62	
✓					Flutlicht 9	V08	2		LED-Werte kopieren	CopyLED(#LED, #InCh, 60)	0-63	
✓					Flutlicht 10	V08	2		LED-Werte kopieren	CopyLED(#LED, #InCh, 59)	0-64	
✓					Flutlicht 11	V08	2		LED-Werte kopieren	CopyLED(#LED, #InCh, 58)	0-65	
✓					Flutlicht 12	V08	2		LED-Werte kopieren	CopyLED(#LED, #InCh, 57)	0-66	
✓					Flutlicht 13	V08	2		LED-Werte kopieren	CopyLED(#LED, #InCh, 56)	0-67	
✓					Flutlicht 14	V08	2		LED-Werte kopieren	CopyLED(#LED, #InCh, 55)	0-68	
✓					Flutlicht 15	V08	2		LED-Werte kopieren	CopyLED(#LED, #InCh, 55)	0-69	

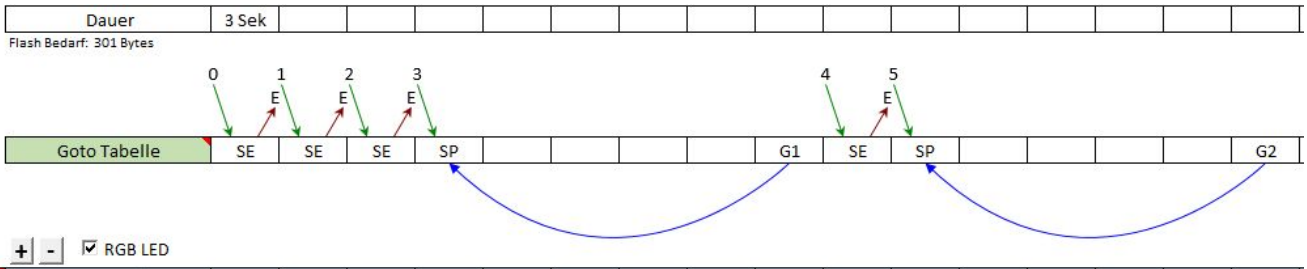
N-Buttons

Für die Märklinisten unter uns sieht es etwas anders aus. Hier benötigt das Pattern zur Steigerung des Komforts jeweils einen zusätzlichen „Aus“-Zustand für Regenbogen und Farbwechsel. Die Aktivierung im Programm Generator ist im obigen Beispiel dargestellt aber deaktiviert (hellblau unterlegt).

Bei einer Aktivierung über N_Buttons benötigt man sechs Zustände im Pattern Configurator:
1. Aus
2. Weiß
3. Aus
4. Regenbogen
5. Aus
6. Farbwechsel



Auf die Goto-Sprünge 2 und 4 könnte man auch verzichten, allerdings müsste man einen über Adresse 11 grün aktivierten Regenbogen über Adresse 10 rot deaktivieren.



\+ - RGB LED

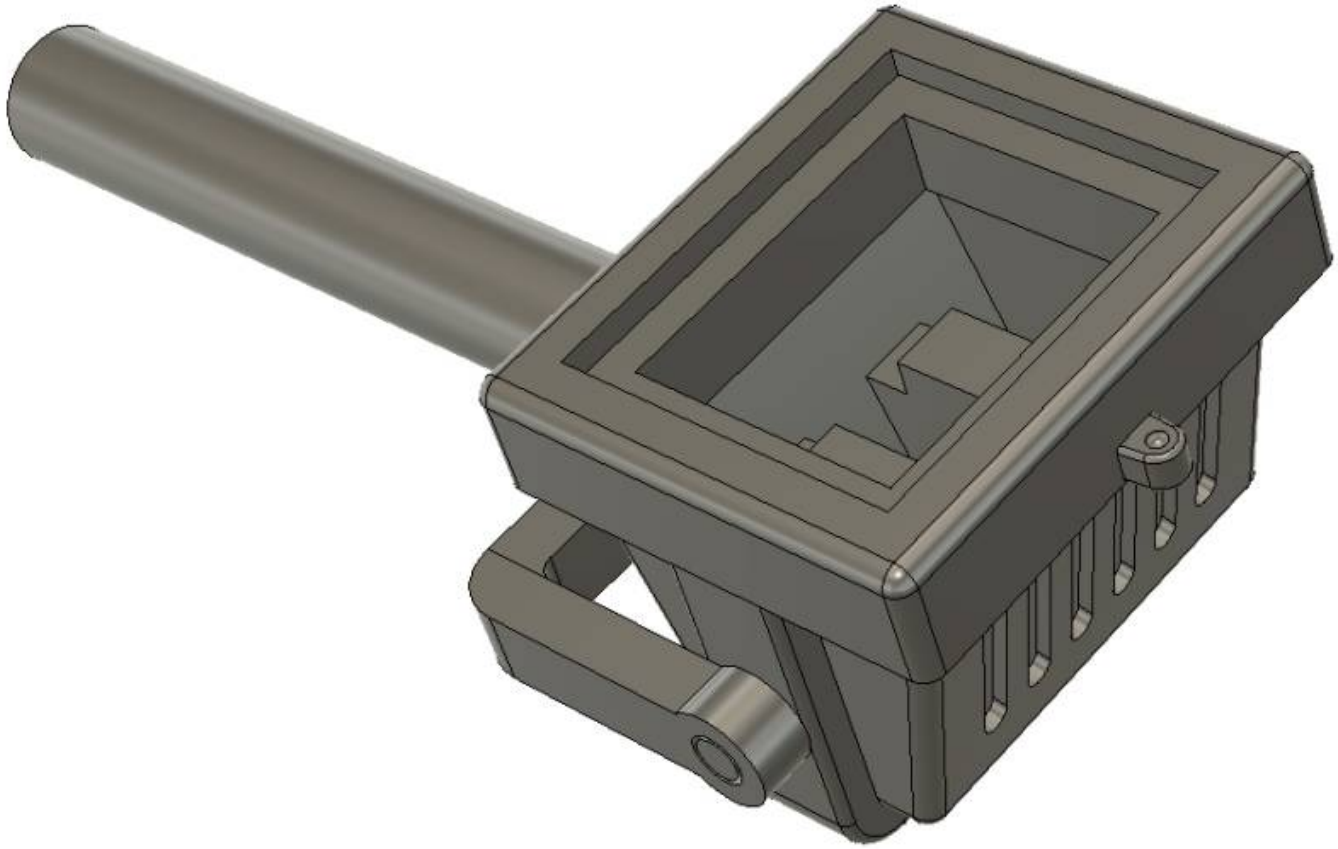
LED Nr	Spalte Nr ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Rot	.	15	.		30	50	30			.		30	55	30		
2	Grün	.	13	.				30	55	25	.				30	55	25
3	Blau	.	8	.	150	60				50	.	150	60				50
4	Rot	.	15	.	30	50	30				.		30	55	30		
5	Grün	.	13	.			30	55	25		.				30	55	25
6	Blau	.	8	.	60				50	150	.	150	60				50
7	Rot	.	15	.	50	30				30	.		30	55	30		
8	Grün	.	13	.		30	55	25			.				30	55	25
9	Blau	.	8	.				50	150	60	.	150	60				50
10	Rot	.	15	.	30				30	50	.		30	55	30		
11	Grün	.	13	.	30	55	25				.				30	55	25
12	Blau	.	8	.			50	150	60		.	150	60				50
13	Rot	.	15	.				30	50	30	.		30	55	30		
14	Grün	.	13	.	55	25				30	.				30	55	25
15	Blau	.	8	.		50	150	60			.	150	60				50
16	Rot	.	15	.			30	50	30		.		30	55	30		
17	Grün	.	13	.	25				30	55	.				30	55	25
18	Blau	.	8	.	50	150	60				.	150	60				50

Flutlicht selbst herstellen mit 3D-Druck

Doch was nützt der schönste Farbwechsel ohne entsprechende Ausstattung. Wer über einen Resin-Drucker verfügt, kann sich die Flutlicht-Strahler mithilfe von SK6812 Mini-E 3228 LEDs selbst herstellen. Ohne entsprechenden Drucker könnte Viessmanns Flutlichtstrahler eine gute Basis für einen Umbau sein. Sollte die zuvor genannte LED dafür zu groß sein, könnte dort die WS2812-2020 Abhilfe schaffen.

Beim selbst gedruckten Flutlicht gehen wir wie folgt vor.
Wir nutzen Kupferlackdraht in 0,1mm und 0,15mm. Laut Datenblatt ist der 0,1er bis 30mA und der 0,15er bis 60mA zugelassen. Bei zwei zu jeweils 60% aktivierten Chips würde der 0,1mm starke Draht vollkommen reichen. Wir nutzen diese beiden Stärken jedoch zur Identifikation. So bekommt Plus einen langen 0,15mm Draht, Minus einen kurzen 0,15mm Draht. Data In bekommt einen langen 0,1mm Draht und Data Out einen kurzen 0,1mm Draht. Der Unterschied zwischen lang und kurz beträgt bei uns immer 3cm. Alle vier Kupferlackdrähte passen durch die 0,5mm Bohrung im Flutlicht-Gehäuse und durch die Haltestange.
Wenn man nun die Kupferlackdrähte anschließen will, kann man sie aufgrund von Stärke und Länge auch unter der Anlage eindeutig zuordnen.

Eignung für 3D-Drucker: FFF / FDM ★☆☆☆ SLA / STL ★★★★



Downloadlink:

- Flutlicht
- Flutlicht Halter

Viel Erfolg bei der Umsetzung

Wer es bis hierher geschafft hat, und trotzdem noch offene Fragen hat, darf sich gern an mich wenden.
— *Michael 2022/09/21*

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

Permanent link:
https://wiki.mobaledlib.de/anleitungen/spezial/codevorlagen/farbwechsel?rev=1663827175

Last update: **2022/09/22 07:12**