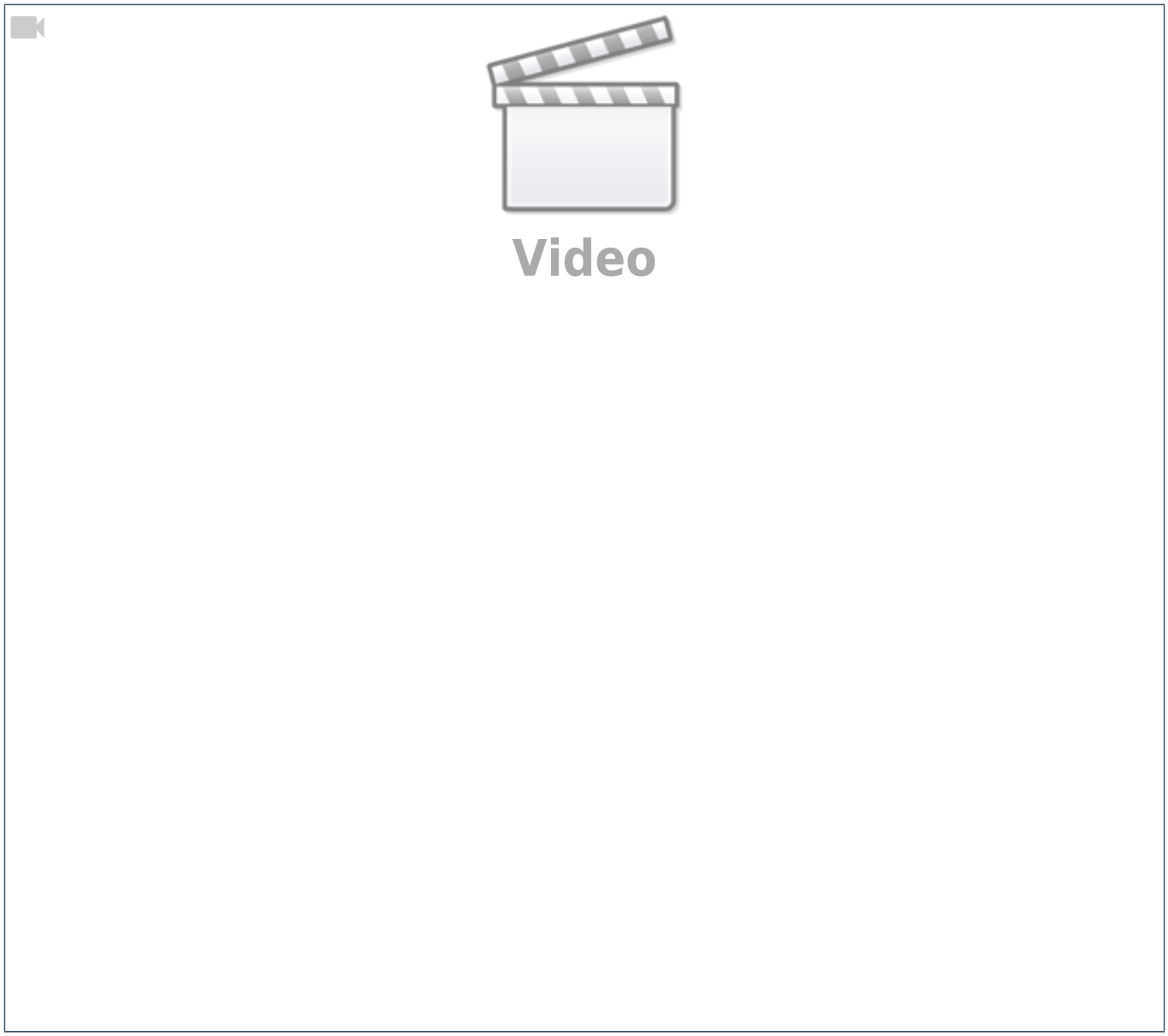
ED mit der MobaLedLib - Ein ungewollt belebtes Haus



Beschreibung der Programmierung mit dem ProgGenerator und Pattern_Configurator.

- Aufgabe

Die Idee war eine Szene darzustellen bei der in ein „belebtes Haus“ eingebrochen wird.
Die Bewohner sollen dabei aber „aus Sicherheitsgründen“ nicht zu Hause sein.
Das Haus ist bei Abwesenheit mit einer Alarmanlage abgesichert und zusätzlich gibt es einen wachsamen Hund in der Nachbarschaft.
Mit dem Bellen des Nachbarhundes soll die Szenerie beginnen. Anschließend ist das Zerschlagen einer Fensterscheibe zu hören und dann sieht man im Haus die/den Täter mit der Taschenlampe nach

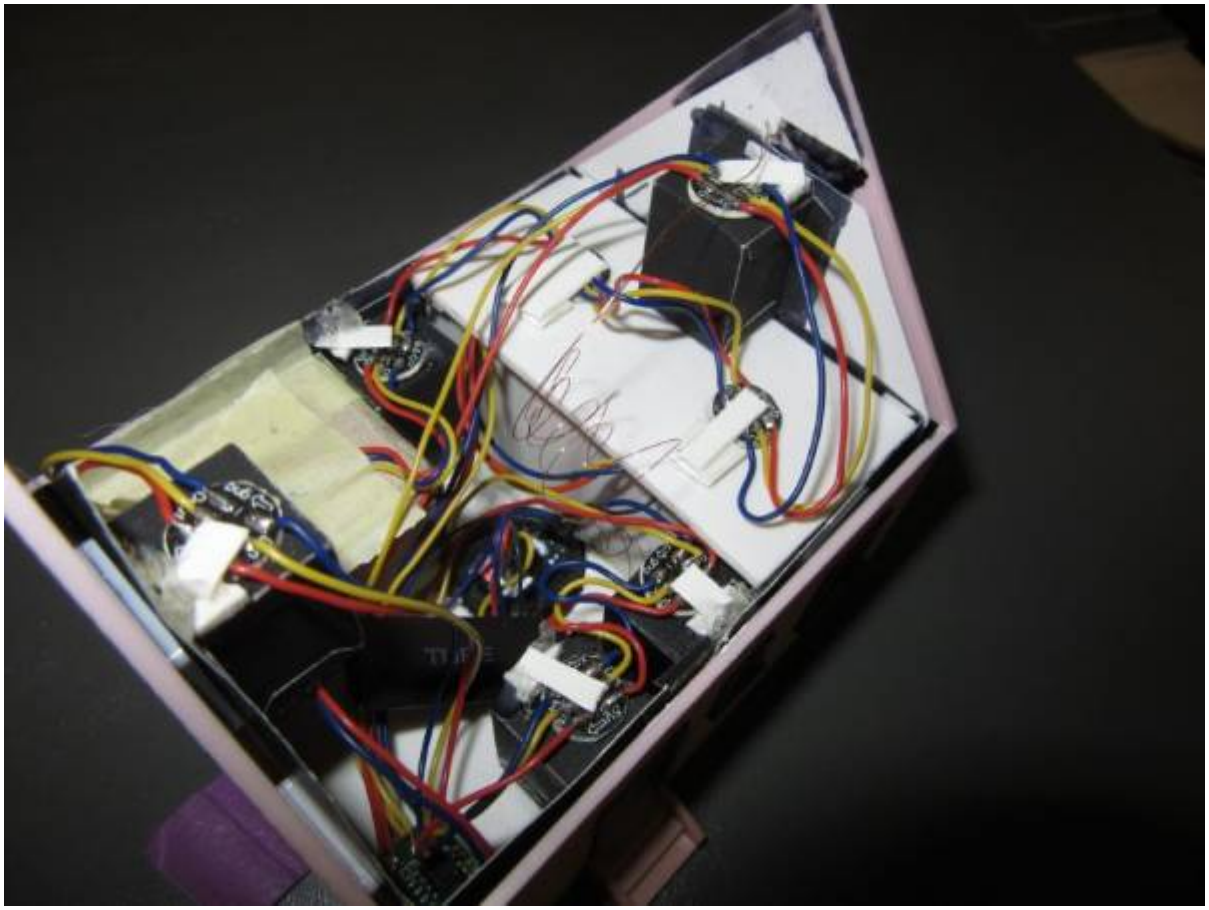
Wertvollem suchen - bis die Polizei kommt.
Optional gibt es noch die Idee den Fluchtweg mit „bewegten Taschenlampen“ nachzustellen und das Zufahren von Polizeifahrzeugen mit einem Car-System (derzeit nicht umgesetzt).

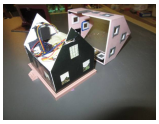
- Modellauswahl

Fertighaus Faller ArtikelNr. 130223 (älteres Modell)



Wände im Bereich der „Einbruchsfenster“ aus Polystyrolplatten. Die Lichtboxen für die Zimmerbeleuchtung sind aus schwarz lackiertem Papier nach dieser Vorlage geklebt.
https://www.modellbahn-anlage.de/2018/02/25/vorlage-fensterbeleuchtung-in-modellbahnhaeuser/
Die Halterung für die LEDs wurden aus Polystyrol Streifen geklebt.





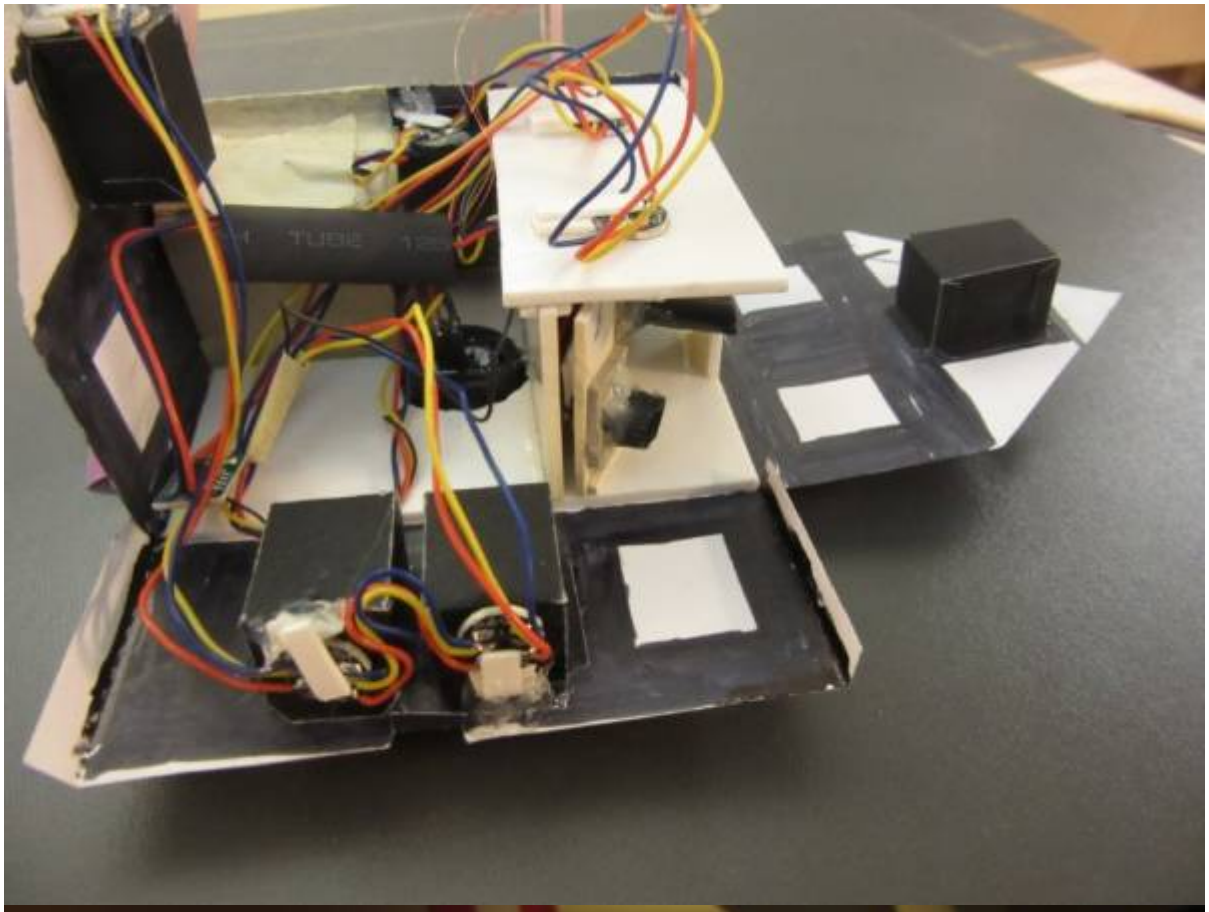
- Hausverkabelung

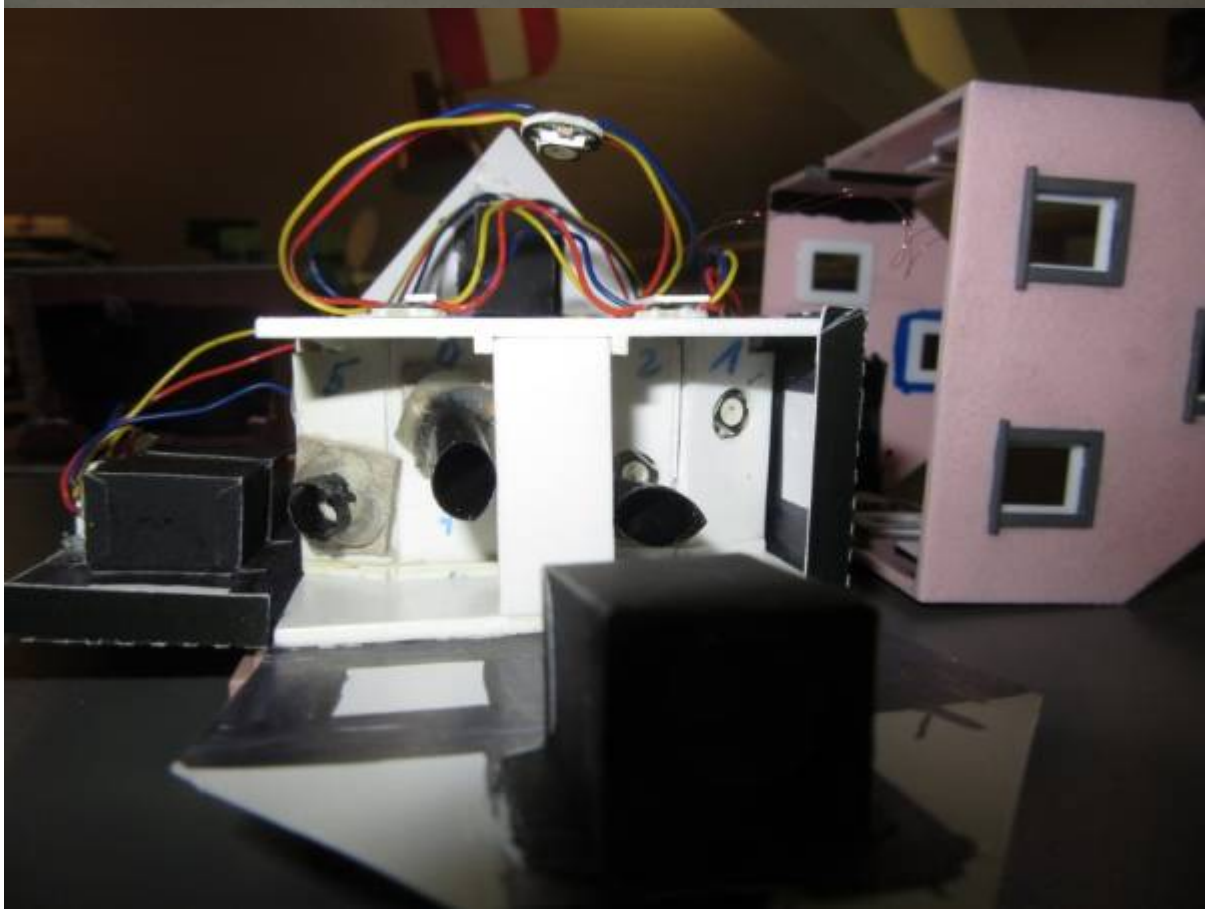
Im Haus sind 8 x WS2812 RGB LED´s als „normale Beleuchtung“ für das <HouseT> Makro im ProgGenerator verbaut.
Zusätzlich wurde eine rote LED an einer WS2811 Platine für die Alarmanlage angeschlossen. 2 Ausgänge/Kanäle sind bei dieser Platine noch frei (Nutzung event. für Gartenwegbeleuchtung etc.).

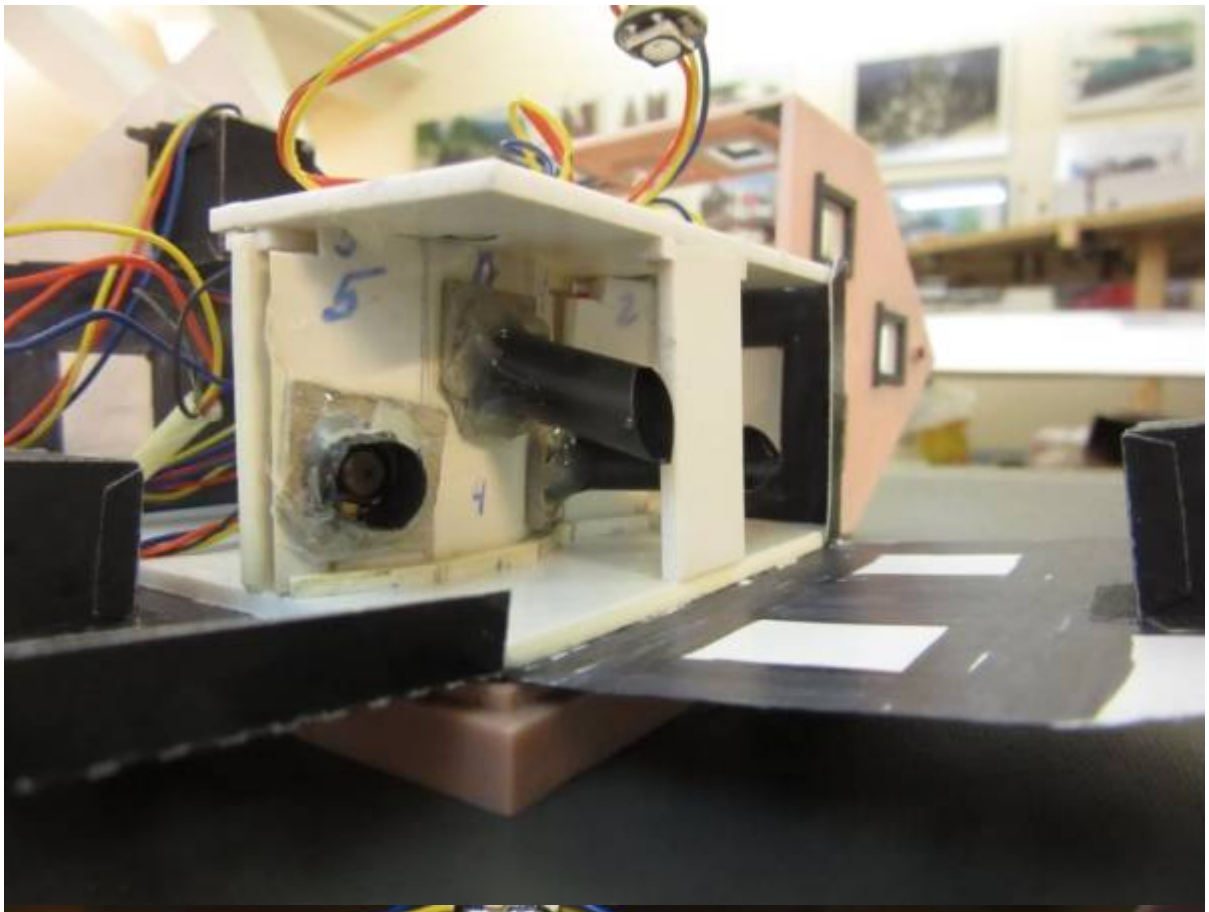


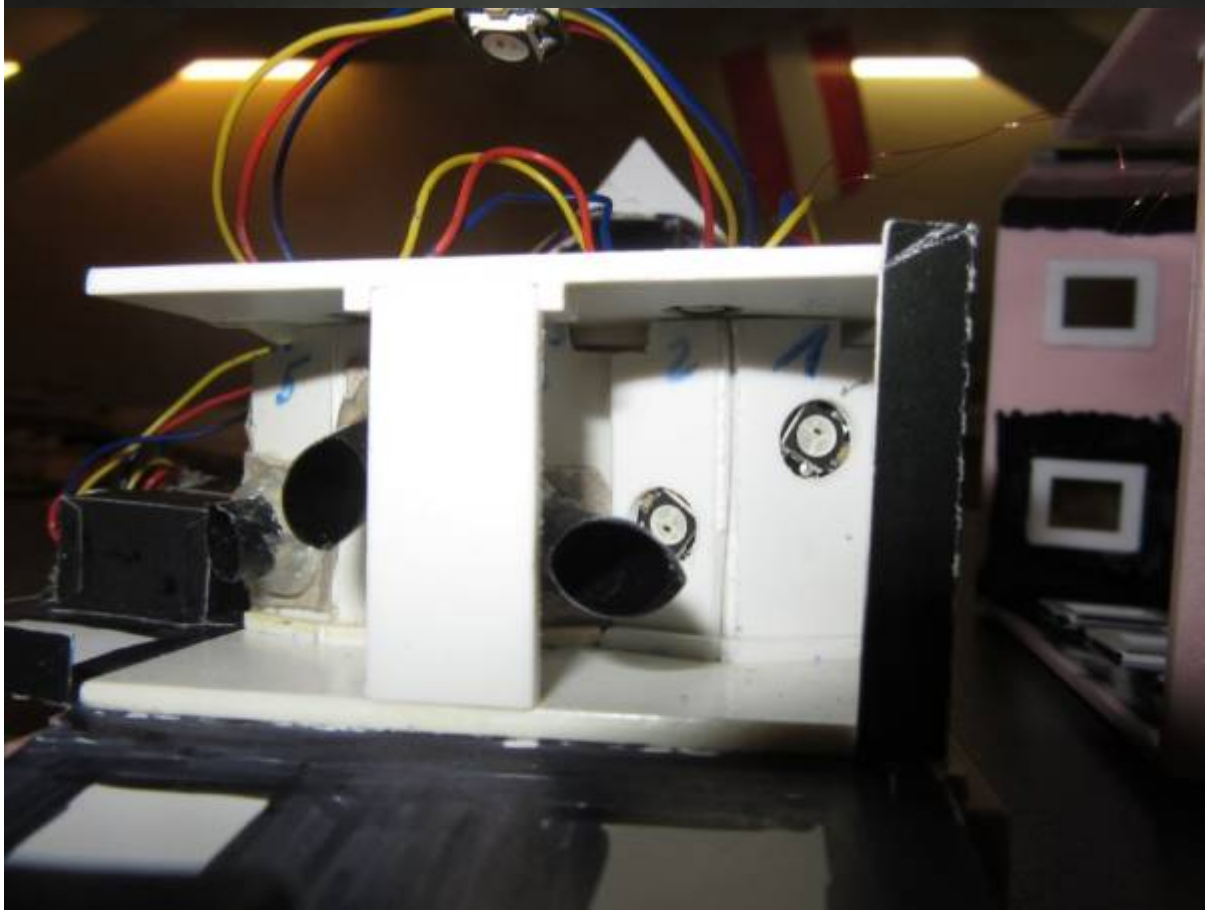
Für die Darstellung der bewegten Taschenlampen wurde ursprünglich eine Lösung mit Servos in Betracht gezogen.
Aufgrund des Aufwandes war aber die Entscheidung für LED´s, die als spezielles Lauflicht im Pattern_Configurator programmiert wurden.
Für dieses „Taschenlampen-Lauflicht“ sind zusätzlich 5 x WS2812 RGB LED verbaut.

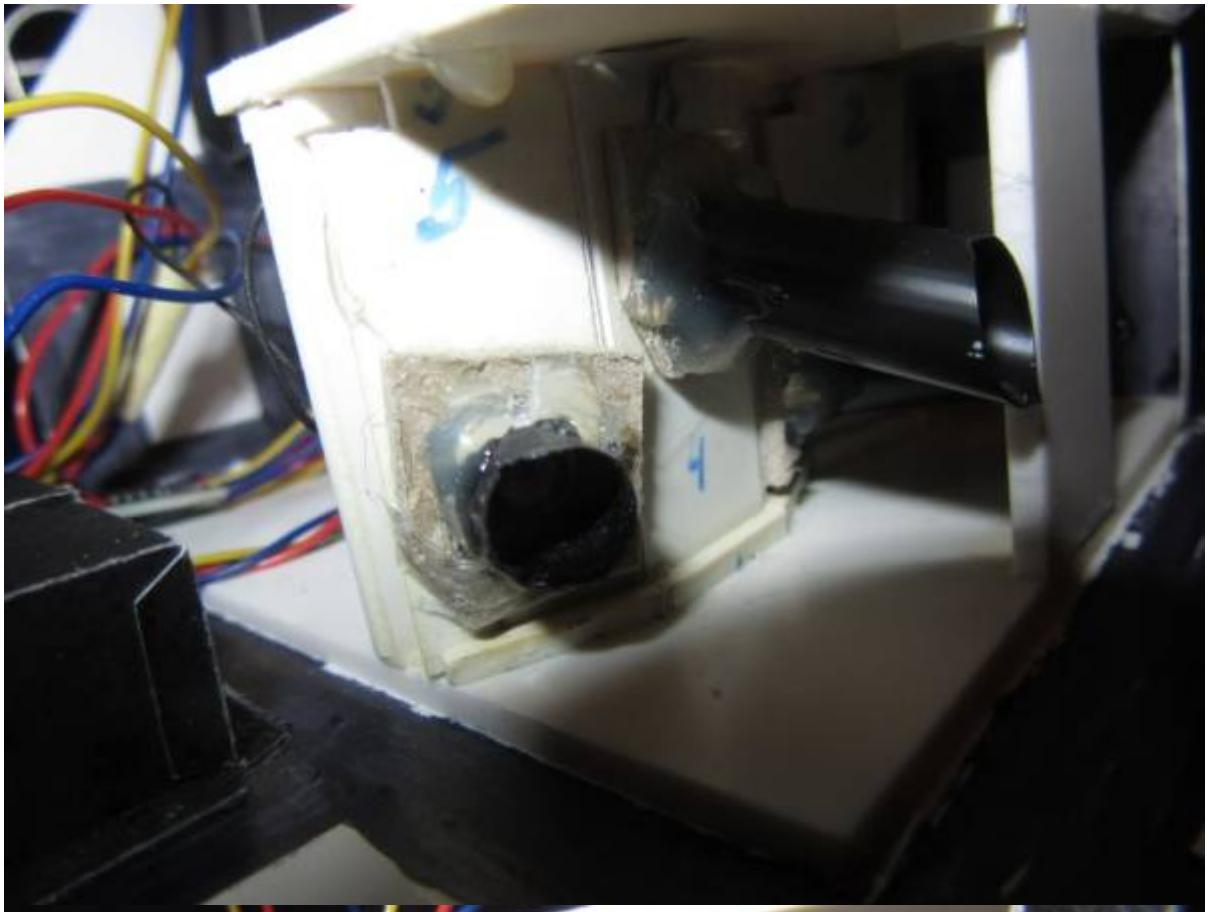
Um den runden Kegelschein der Taschenlampen zu simulieren wurden 3 der 5 RGB LED mit schwarzen Strohhalmen verkleidet.

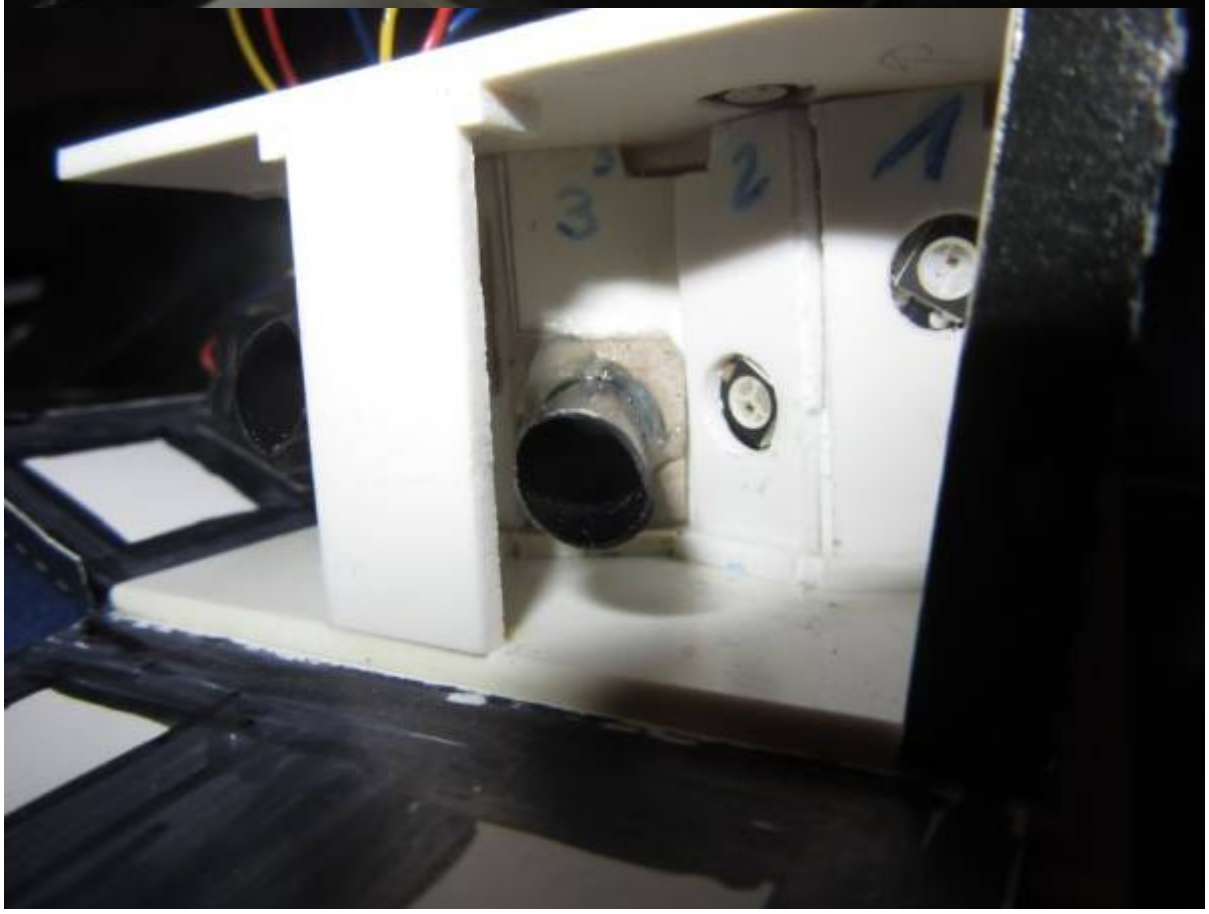


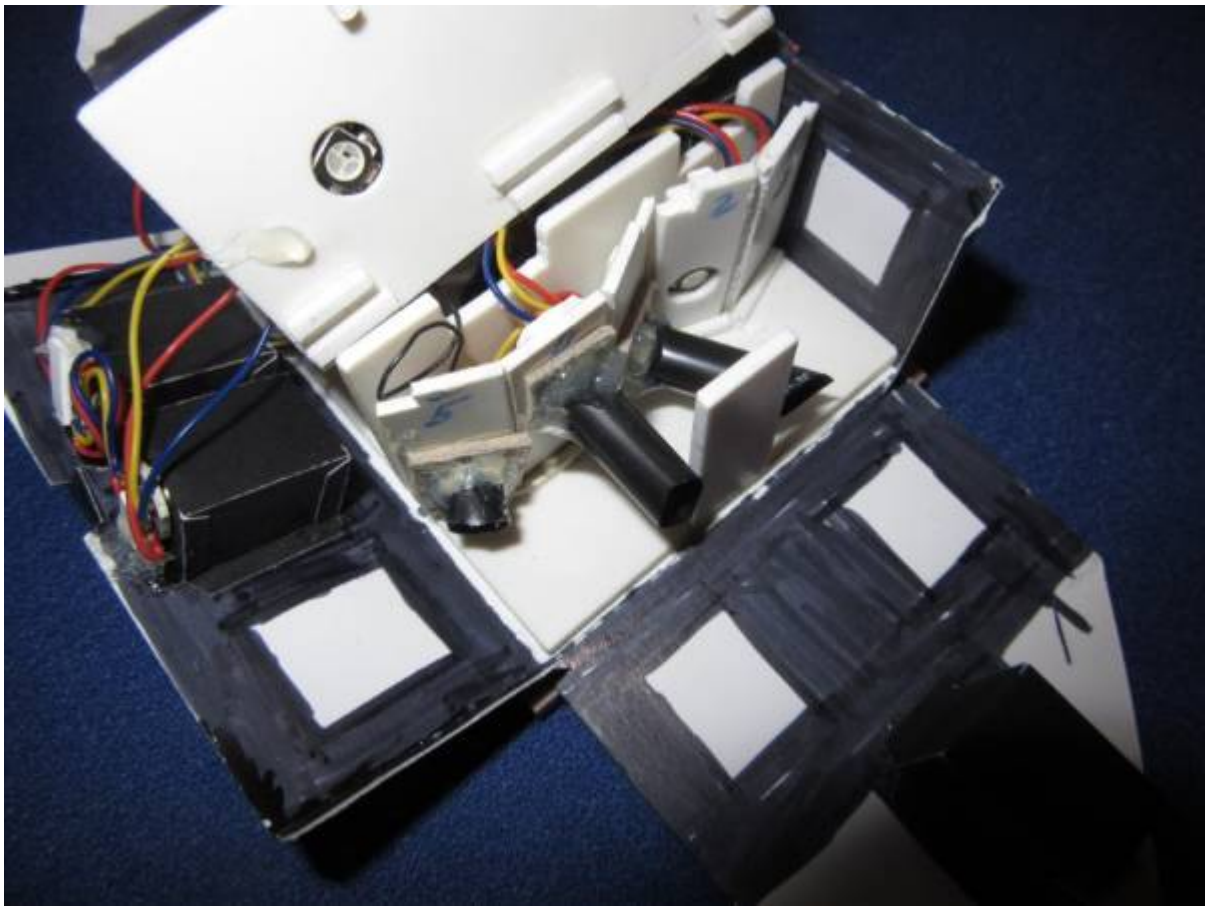






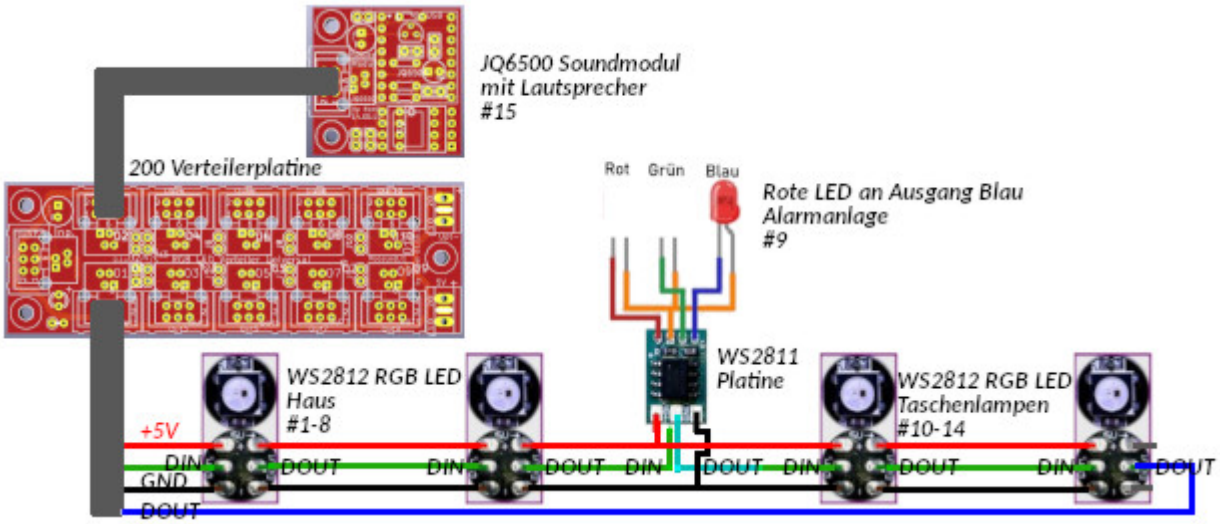



3
2
1



Der Grundanschluss des Hauses an die MLL Verkabelung bzw. Verteilerplatine erfolgt über den 3D – Gehäuseadapter für Pfostenstecker von Jürgen.
adapter_fuer_pfostenstecker

- Der Schaltplan



- Sound

MLL Soundmodul JQ6500 mit Lautsprecher von Reichelt. soundplatinen

- Programmierung

- Szenenablauf festlegen

- Welche Aktionen möchte ich darstellen.
- Wann soll welche Aktion starten.
- Wie sollen die entsprechenden Aktionen gestartet werden.

- DCC und Taster programmieren

- Die „normale“ Hausbeleuchtung und die Einbruchszene sollen entweder mit einer DCC Adresse oder als Gag mittels Taster am Anlagenrand gestartet werden.
- Als Taster wurden vorerst 2 Taster der Hauptplatine <SwitchD1> und <SwitchD2> mit LED Anzeige programmiert. Später werden Anlagentaster verwendet.
- Als DCC Adresse wurde für die „normale“ Hausbeleuchtung vorerst die 1 als Ein/Ausschalter und für die „Einbruchszene“ Adresse 2 als Taster definiert.

- Um die Preiserlein nicht zu gefährden müssen sie alle das Haus verlassen bevor die Einbruchszenerie gestartet wird.
- Sprich es müssen alle Lichter des <HouseT> Makro aus sein und erst dann darf/kann die Einbruchsszenerie gestartet werden.
- Für diese Absicherung wurden <Logic> Makros verwendet.

- Eine Besonderheit gab es beim <HouseT> Makro zu beachten.
- Nach dem Abschalten des Makros leuchten verschiedene LED´s aber noch weiter und gehen erst später zufällig aus.
- Nähere Beschreibung dieser Problematik im Stammtischvideo Jänner 2021 Stammtisch MLL Januar 2021
- Um diesen Effekt abzufangen wurden insgesamt 8 <Led_to_Var> Makros eingefügt um den Helligkeitswerte der einzelnen LED abzufragen.
- Die Variablen <licht> – <licht7> werden aktiv wenn die einzelnen LED´s der Hausbeleuchtung dunkel (Helligkeitswert=0) sind.
- Diese Werte werden im <Logic> Makro „HausEDBeginn“ mit „AND“ Verknüpfungen abgefragt.
- D.h. wenn alle „licht“ Variablen aktiv sind dann sind alle Lichter im Haus ausgeschaltet.
- Erst dann ist es möglich das Schrittschaltwerk/Einbruchszenerie „HausEDBeginn1“ zu starten.

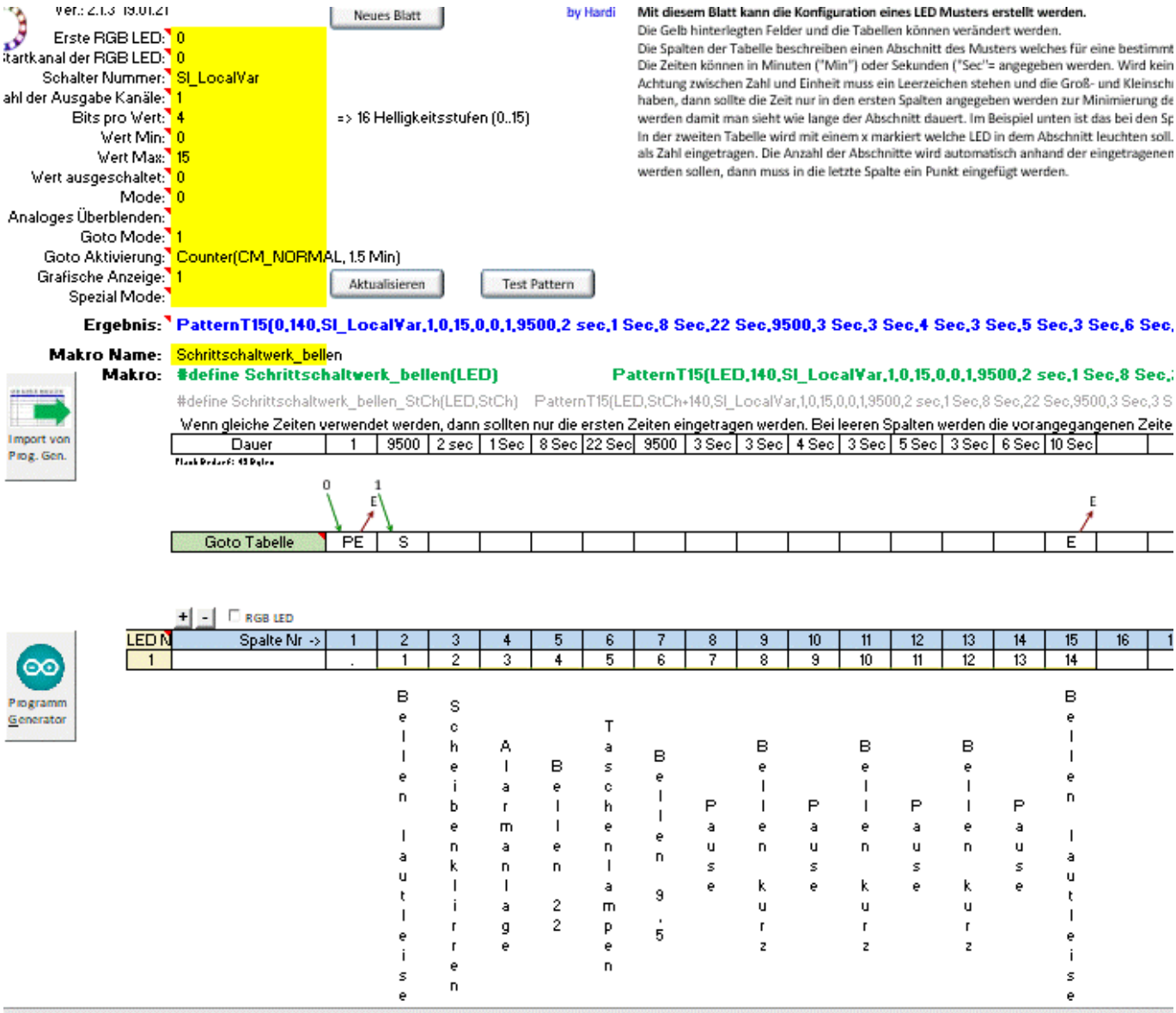
Aktiv	Filter	Adresse oder Name	Typ	Start-wert	Beschreibung	Verteiler-Numm	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte	Start Led	LEDs	InCh	Loc InCh	LED Kan	Start Tast	Start LED	Start LED
✓					Zeigt an, dass die LEDs angesteuert werden			RGB_Heartbeat2(#LED, 5, 100)	0	1	0	0	0			
✓	Tast LED	SwitchD1			Normales belebtes Haus#1			PushButton_RGB_0_2(#LED, #InCh, TastL0, 1, 0, 0, 0, 0, 10,		1	1	0	1	0		
✓	Tast LED	SwitchD2			Start des Einbruchs			PushButton_RGB_0_1(#LED, #InCh, TastM0, 1, 1, 0, 0, 5 Sek,		1	1	0	1	1		
✓	Tast LED	TastL1			LED Taster Hauptplatine links gelb			Mainboard_LED(1, #InCh)			1	0				
✓	Tast LED	TastM1			LED Taster Hauptplatine mitte rot			Mainboard_LED(2, #InCh)			1	0				
✓		1	AnAus 0					Logic(HausED1, #InCh OR TastL1)			1	0				
✓		HausED1						Logic(HausED, #InCh AND NOT Alarm AND NOT Klirren AND NOT			1	0				
✓		HausED			Normales belebtes Haus#1			HouseT(#LED, #InCh, 5, 7, 1, 5, ROOM_WARM_W, ROOM_BRIGHT,	1	8	1	0	0			
✓								LED_to_Var(licht, 1, =, 0)								
✓								LED_to_Var(licht1, 3, =, 0)								
✓								LED_to_Var(licht2, 6, =, 0)								
✓								LED_to_Var(licht3, 9, =, 0)								
✓								LED_to_Var(licht4, 12, =, 0)								
✓								LED_to_Var(licht5, 15, =, 0)								
✓								LED_to_Var(licht6, 18, =, 0)								
✓								LED_to_Var(licht7, 21, =, 0)								
✓		2	Rot		Haus#1 Beginn des ED			Logic(HausEDBeginn, #InCh OR TastM1 AND NOT HausED)			1	0				
✓		HausEDBeginn						Logic(HausEDBeginn1, #InCh AND licht AND licht1 AND licht2			1	0				
✓		HausEDBeginn1			Schrittschaltwerk_bellen (pc)			// Activation: Counter(CM_NORMAL, 1.5 Min) New_Local_Var() Counter(CF_ONLY_LOCALVAR | CM_NORMAL, #InCh, SI_1, 1.5 Min, 2) PatternT15(#LED,140,SI_LocalVar,1,0,15,0,0,1,9500,2 sec,1 Sec,8 Sec,22 Sec,9500,3 Sec,3 Sec,4 Sec,3 Sec,5 Sec,3 Sec,6 Sec,10 Sec,16,50,84,118,152,186,220,14		C1-1	1	0	1	2		

- HouseMakro erstellen

- Mit dem ProgGenerator ein beliebiges <HouseT> Makro für die „normale belebtes Hausbeleuchtung“ erstellen.

- Schrittschaltwerk / Zustandsautomat erstellen

- Im Pattern_Configurator wurde ein zeitlicher Ablauf der Szene erstellt.



- 14 Szenenschritte sind im Ablauf vorhanden. Die nähere Beschreibung ist vertikal vermerkt.
- Diesen 14 Schritten sind Helligkeitswerte zugeordnet.
- Die Helligkeitsstufen werden im Feld <Bits pro Wert> festgelegt.
- Im Feld <Wert Max> ist ein maximaler Wert eingetragen. In diesem Fall sind die Helligkeitswerte von 1-14.
- Diese Werte werden im ProgGenerator mit dem Makro <Led_to_Var> abgefragt um die entsprechende Aktion ablaufgerecht zu starten.
- Um den Ablauf zu starten wurde ein <Goto Mode> eingetragen. Näheres im Stammtischvideo vom Jänner 2021 Stammtisch MLL Januar 2021
- Dieses Schrittschaltwerk-Pattern wird anschließend in den ProgGenerator übertragen.
- Im ProgGenerator werden mit den oben beschriebenen Makros <Led_to_Var) Variablen erzeugt die bei bestimmten Helligkeitswerten aktiviert werden.
- Hier gibt es verschiedene Vergleichsmöglichkeiten.
- In diesem Fall wurden vorwiegend = Vergleiche durchgeführt.
- Da die Alarmanlage ab dem Fensterklirren eingeschaltet ist wurde der > (größer als) Vergleich angewendet.
- Diesen Variablen sind im Anschluss die Aktionen zugeordnet.
- Z.B. Variable <Alarm> das <Blink2> Makro für die rote LED der Alarmanlage, Variable <TLampen> das im Pattern_Configurator erzeugte Makro für die Taschenlampen, usw. bis zu den Sounddateien und Polizeiblinklichtern.

Aktiv	Filter	Adresse oder Name	Typ	Start-wert	Beschreibung	Verteiler-Numm	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte	Start Led	LEDs	InCh	Loc InCh	LED Kan	Start Tast	Start LED	Start LED
✓		2	Rot		Haus#1 Beginn des ED			Logic(HausEDBeginn, #InCh OR TastM1 AND NOT HausED)			1	0				
✓		HausEDBeginn						Logic(HausEDBeginn1, #InCh AND licht AND licht1 AND licht2			1	0				
✓		HausEDBeginn1			Schrittschaltwerk_bellen (pc)			// Activation: Counter(CM_NORMAL, 1.5 Min) New_Local_Var() Counter(CF_ONLY_LOCALVAR | CM_NORMAL, #InCh, SI_1, 1.5 Min, 2) PatternT15(#LED,140,SI_LocalVar,1,0,15,0,0,1,9500,2 sec,1 Sec,8 Sec,22 Sec,9500,3 Sec,3 Sec,4 Sec,3 Sec,5 Sec,3 Sec,6 Sec,10 Sec,16,50,84,118,152,186,220,14 ,127,128,0,0,0,0,0,0,0,0,0,0,0,0,63)		C1-1	1	0	1	2
✓								LED_to_Var(bellLautLeis95, 0, =, 1)			0	0				
✓								LED_to_Var(Klirren, 0, =, 2)			0	0				
✓								LED_to_Var(Alarm, 0, >, 2)			0	0				
✓								LED_to_Var(bellLang29, 0, =, 4)			0	0				
✓								LED_to_Var(TLampen, 0, >, 4)			0	0				
✓								LED_to_Var(bell95, 0, =, 6)			0	0				
✓								LED_to_Var(bell3, 0, =, 8)			0	0				
✓								LED_to_Var(bell31, 0, =, 10)			0	0				
✓								LED_to_Var(bell32, 0, =, 12)			0	0				
✓								LED_to_Var(polizei, 0, >, 12)			0	0				
✓								LED_to_Var(bell4, 0, =, 14)			0	0				
											0	0				
✓		Alarm						Blink2(#LED, C3, #InCh, 0.4 Sek, 0.4 Sek, 0, 60)	9	C3-3	1	0	0			
		TLampen			RGB_Einbrecher_lowHell (pc)			// Activation: Counter(CM_NORMAL, 50 Sek) New_Local_Var() Counter(CF_ONLY_LOCALVAR | CM_NORMAL, #InCh, SI_1, 50 Sek, 2)	10	5	1	0	0			

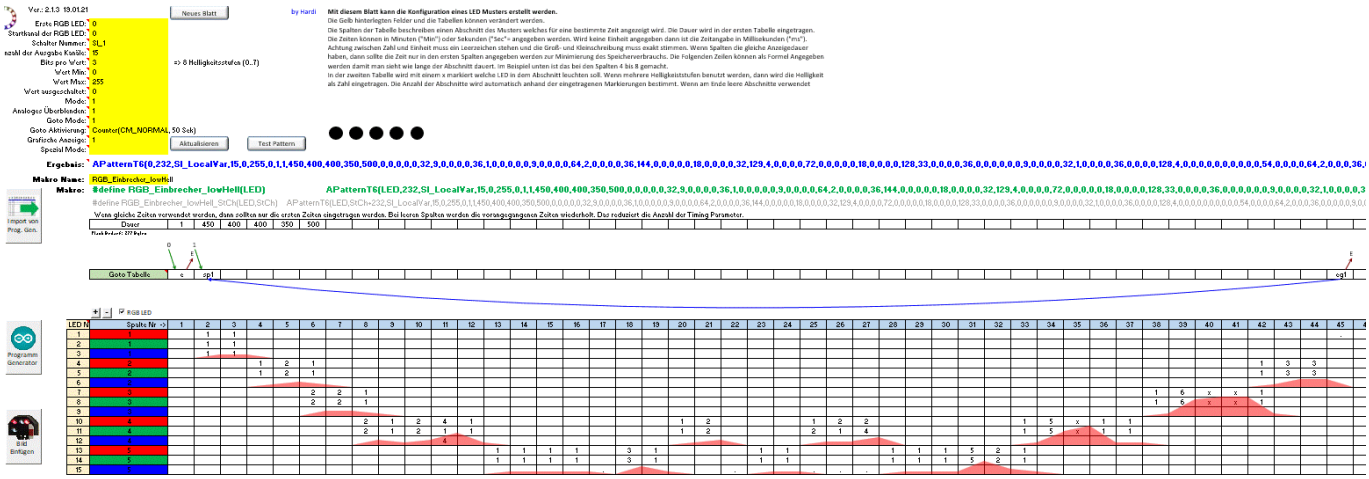
Aktiv	Filter	Adresse oder Name	Typ	Start-wert	Beschreibung	Verteiler-Numm	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte	Start Led	LEDs	InCh	Loc InCh	LED Kan	Start Tast	Start LED	Start LED
											0	0				
✓		Alarm			Alarmanlage-rote LED an WS2811 Kanal Blau			Blink2(#LED, C3, #InCh, 0.4 Sek, 0.4 Sek, 0, 60)	9	C3-3	1	0	0			
✓		TLampen			RGB_Einbrecher_lowHell (pc)			// Activation: Counter(CM_NORMAL, 50 Sek) New_Local_Var() Counter(CF_ONLY_LOCALVAR | CM_NORMAL, #InCh, SI_1, 50 Sek, 2) APatternT6(#LED,232,SI_LocalVar,15,0,255,0,1,1,450,400,400,350,500,0,0,0,0,0,32,9,0,0,0,0,36,1,0,0,0,0,0,9,0,0,0,0,64,2,0,0,0,0,36,144,0,0,0,0,0,18,0,0,0,0,32,129,4,0,0,0,0,72,0,0,0,0,0,0,18,0,0,0,0,128,33,0,0,0,0,36,0,0,0,0,0,0,9,0,0,0,0,32,1,0,0,0,0,36,0,0,0,0,128,4,0,0,0,0,0,0,0,0,0,54,0,0,0,0,64,2,0,0,0,36,0,0,0,0,0,9,0,0,0,0,0,0,0,0,0,0,36,0,0,0,0,128,4,0,0,0,136,0,0,0,0,0,10,0,0,0,0,64,4,0,0,0,0,0,72,0,0,0,0,0,0,9,0,0,0,0,32,1,0,0,0,0,180,0,0,0,0,0,9,0,0,0,72,144,0,0,0,0,45,0,0,0,0,224,7,0,0,0,0,36,0,0,0,0,128,4,0,0,0,72,0,0,0,0,0,54,0,0,0,0,224,7,0,0,0,0,252,0,0,0,64,130,4,0,0,0,216,0,0,0,0,0,27,0,0,0,0,0,0,0,0,0,0,0	10	5	1	0	0			
✓		bellLautLeis95			alle Sounddateien bearbeitet mit Audacity			Sound_JQ6500_BG_Seq3(#LED, #InCh)	15	C1-1	1	0	0			
✓		Klirren			Fensterscheibenklirren			Sound_JQ6500_Seq5(#LED, #InCh)	15	^ C1-1	1	0	0			
		bellLang29			Unterschiedlich lange Bell-Sequenzen			Sound_JQ6500_BG_Seq4(#LED, #InCh)		^ C1-1	1	0	0			
✓		bell95						Sound_JQ6500_Seq2(#LED, #InCh)	15	^ C1-1	1	0	0			
✓		bell3						Sound_JQ6500_Seq1(#LED, #InCh)	15	^ C1-1	1	0	0			
✓		bell31						Sound_JQ6500_Seq1(#LED, #InCh)	15	^ C1-1	1	0	0			
✓		bell32						Sound_JQ6500_Seq1(#LED, #InCh)	15	^ C1-1	1	0	0			
✓		bell4						Sound_JQ6500_BG_Seq3(#LED, #InCh)	15	^ C1-1	1	0	0			
✓								// Next_LED(0)	16	0	0	0	0			
✓		polizei			RGB-Leds zur Simulation der Polizei für den Film			BlueLight2(#LED, C3, #InCh)	16	C3-3	1	0	0			
✓		polizei						BlueLight1(#LED, C3, #InCh)	17	C3-3	1	0	0			

- Sounddateien schneiden

- Frei zugängliche Sounddateien aus dem Internet wurden mit der Software Audacity https://www.audacity.de/ in unterschiedlicher Lautstärke und Länge bearbeitet.
- Die fertigen 5 Sounddateien wurden in das Soundmodul JQ6500 übertragen.
- Beschreibung in der MLL Wiki 500DE-Sound Modul JQ6500 - Version 1.1
- ACHTUNG: die Sounddateien müssen im mp3 Format sein und auch nicht zu groß. Falls eine Fehlermeldung beim Hochladen kommen sollte ist das in der Regel der Grund.

- Taschenlampenlauflicht erstellen

- Mit dem Pattern_Configurator wurde für die 5 x WS2812 LEDs ein Lauflicht erstellt.



- Um einen realistischen Ablauf zu generieren waren sehr viele Versuche und Tests notwendig um die oben ersichtliche Programmierung zu entwerfen.
- Die Schwierigkeit war, für die Taschenlampenbewegung einen weichen Verlauf des Auf- und Abblendens von einer LED zur Nächsten zu erstellen.
- Die Zeiten für die einzelnen Schritte sind in den ersten Zeilen angegeben.
- Der gesamte Ablauf wird aber aufgrund der <Goto Aktivierung> Counter.... bis 50 Sek. wiederholt.
- Anschließend springt der Ablauf zur Position 0 und wird erst wieder durch den entsprechenden Helligkeitswert im Schrittschaltwerk des Szenerie Ablaufs aktiviert.
- Dieses Pattern wird anschließend in den ProgGenerator übertragen. Siehe Variable <TLampen>

- Download

- Dateien Download auf Github _kommt in Kürze 💪

Gerald Bock 2021/02/11

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

Permanent link:
https://wiki.mobaledlib.de/anleitungen/videos/ed_ungewollt_belebtes_haus?rev=1613078818

Last update: **2021/02/11 22:26**