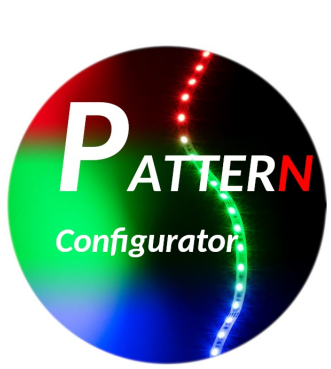
Heft 3

Zeitlicher Ablauf mittels „Schrittschaltwerk“ in der weltbesten MobaLedLib



Wie ich einen festen Ablauf in der MobaLedLib programmiere

Januar 2025

Inhaltsverzeichnis

Hier ein Hinweis in eigener Sache:

1. Alle Produkt- oder Firmennamen in meinem Text nenne ich nur des Berichtes halber. Es handelt sich um die Nennungen keinesfalls um Werbung für diese Marken bzw. deren Produkte. Ich habe weder Geld- noch Sachzuwendungen für die Nennungen erhalten.

2. Aus Einfachheitsgründen verzichte ich auf die heute richtigerweise übliche männlich/weiblich/divers-Schreibweisen. Selbstverständlich ist mit dem zum Beispiel mit dem Begriff „Modellbahner“ nicht nur die männliche, sondern auch die weibliche und auch die diverse Ansprache gemeint.

3. Alle von mir beschriebenen Vorgänge, Tätigkeiten, Schaltpläne und sonstige Sachen veröffentliche ich hier ohne Anspruch auf Richtigkeit und Fehlerfreiheit. Alles hier genannte spiegelt lediglich meine Vorgehensweise wieder und stellt keine Garantie dar, dass es bei anderen auch so funktioniert.

4. Einige Texte und Abbildungen habe ich dem MobaLedLib-Wiki (https://wiki.mobaledlib.de) entnommen.

5. Selbstverständlich beziehen sich alle Angaben auf meine Installation. Es ist durchaus möglich, dass mittlerweile neue Versionen der Software und der Platinen erhältlich sind und dadurch in der Anwendung etwas abweichen.

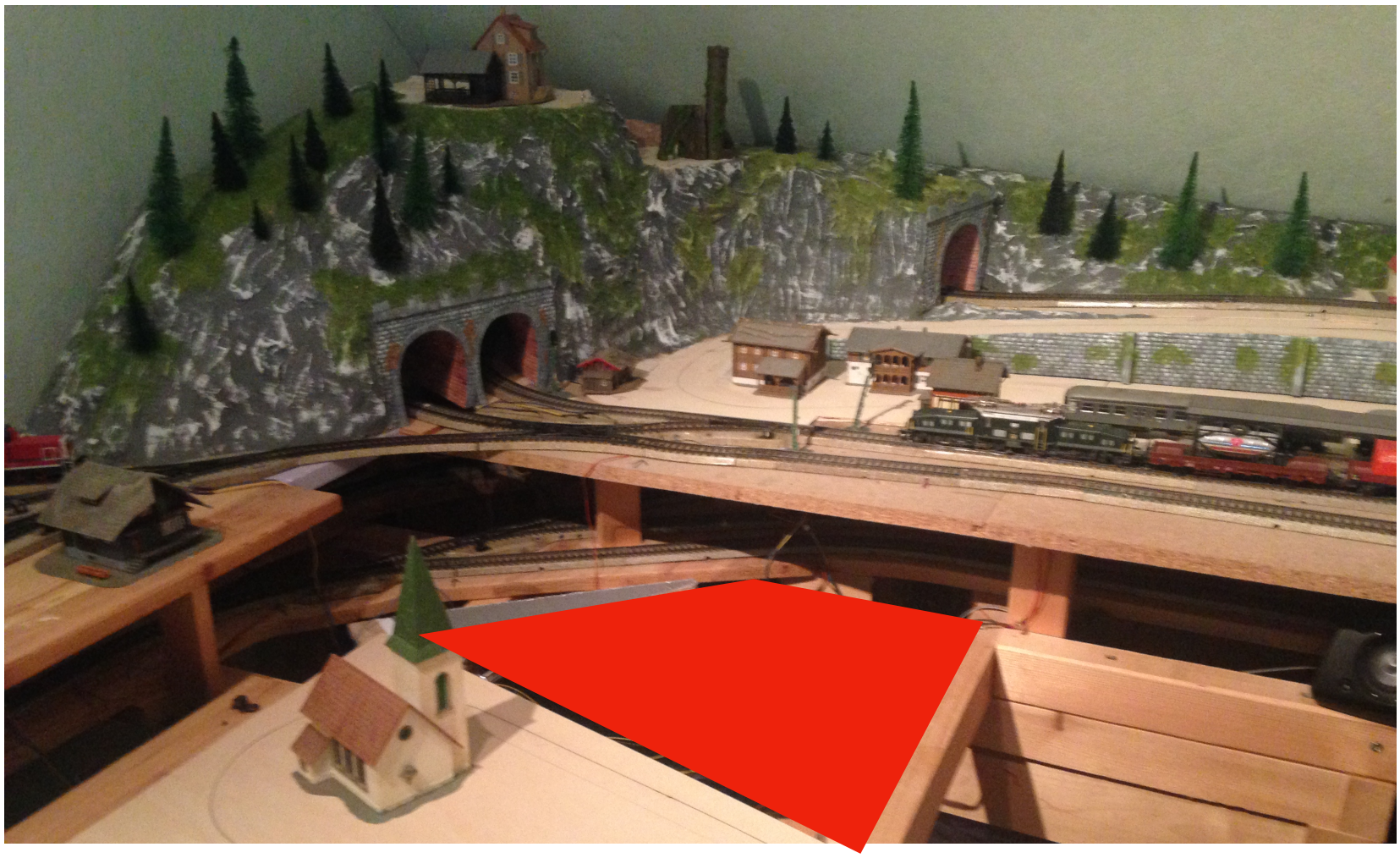
Vorwort

Wir schreiben das Jahr 2025 - Januar. Mittlerweile konnte ich meine Modelleisenbahn wieder aufbauen, diese wurde leider durch das katastrophale Hochwasser im Juli 2021 stark in Mitleidenschaft gezogen. Aber ich will nicht Jammern, trotz vieler Schwierigkeiten es geht nur aufwärts! Stand Januar 2025: die tragenden Füße und Rollen, die Elektroinstallation und teilweise die Grundplatte - also alles, was unterhalb der ca. 1 Meter hohen Wasserlinie war - ist ersetzt bzw. neu. Die ersten Züge können auch wieder fahren und auch die Hard- und Software ich wieder einsatzfähig.

Jetzt könnte ich mich weiter mit meiner Modelleisenbahn beschäftigen. Mir fallen hier viele Themen ein wie z. B. die Steuerung mit Traincontroller Gold perfektionieren (z. B. Zugfahrten bzw. einen Fahrplan o. ä.), die Landschaft weiter aufbauen und gestalten oder unter der Hauptplatte einen Schattenbahnhof anlegen. Ideen und Beschäftigung hätte ich wahrlich genug, wenn da nicht die MobaLedLib wäre.

Immer wieder kommen mir neue Ideen, die ich dann auch versuche für mich umzusetzen. Vor einigen Wochen war dies eine selbst entworfene „Mini-Verteilerplatine“, auf der ein Eingangs-Wannenstecker auf 5 weitere Wannenstecker verteilt wird (jeder durch Jumper terminierter). Je nach Bedarf kann ich bis zu 4 der 5 Verteiler-Wannenstecker „absägen“. Diese Platine dient mir zur Verteilung von Beleuchtung in Gebäuden (entweder diese Platine im Gebäude oder unter der Grundplatte). Funktioniert einwandfrei - also einige Platinen bei Aisler bestellt, bestückt und in Betrieb genommen. Aber mit meiner Moba bin ich währenddessen wieder nicht weiter gekommen (Grüße hier an den Hauptinitiator der MLL Hardi, siehe seine Signatur im Stummi-Forum).

Aber trotzdem hatte ich vor einigen Wochen die Idee (da ich ja grundsätzlich nichts vorher plane), in meiner Modellbahn-Anlage ein Feuerwehrszenario einzubinden. Den Platz habe ich sofort gefunden: da klafft doch tatsächlich noch ein Loch in der Grundplatte wo ich eigentlich eine „Stadt“ vorgesehen hatte (Foto auf der nächsten Seite). Meine Idee: hier könnte ich doch das Feuerwehrszenario installieren. Wie fast alles auf meiner Anlage nutze ich hierfür Häuser-Bausätze, die mittlerweile 30 bis 40 Jahre alte sind: das „Brennende Finanzamt“ von Faller, eine Schmiede und ein Feuerwehrgerätehaus. Auch die zum Einsatz kommenden Feuerwehrfahrzeuge haben schon über 40 Jahre „auf dem Buckel“. Das Blaulicht des Leiterwagens wird original über eine kleine Platine mit 16V-Anschluss gesteuert und auch das „brennende Finanzamt“ benötigt 16 Volt. Alle anderen Einsatzfahrzeuge habe ich „aufgepimpt“: mit einem Handbohrer habe ich die Blaulichter von innen her aufgebohrt und dort (für mich) extrem kleine LEDs eingeklebt. Und auch die Frontscheinwerfer habe ich mittels LEDs beleuchtet. Jede LED endet unter der Platte in einem 2-poligen Stecker.

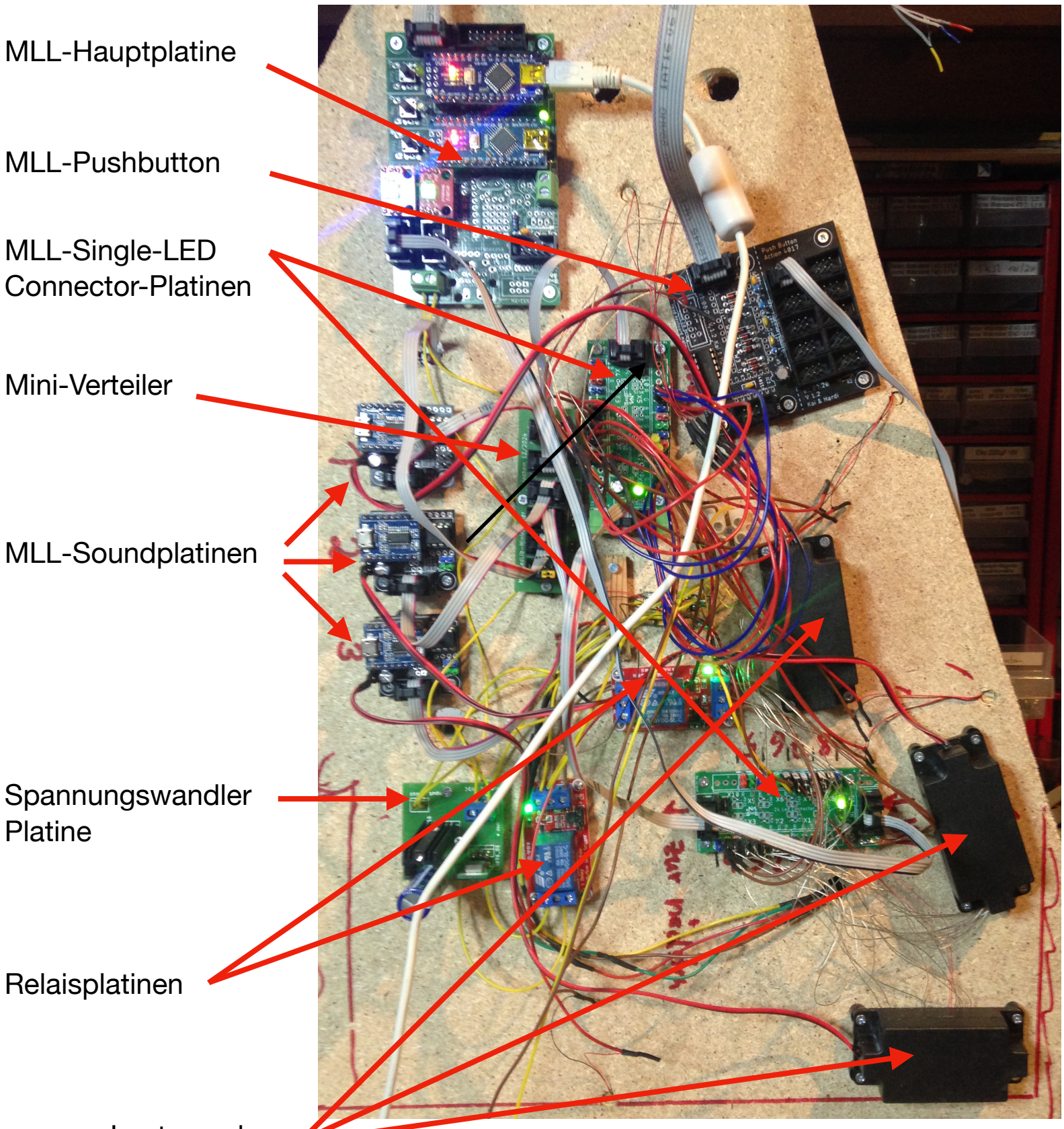


Hier ist der Platz für das Feuerwehr-Szenario (rot markiert).

So sieht das Szenario aus: links das Feuerwehrgerätehaus, mitte die Schmiede, rechts das „brennende Finanzamt“, im Vordergrund links ein Mercedes-Gerätewagen, dann weiter nach rechts: Unimog-Gerätewagen, Leiterwagen, Kranwagen, im Vordergrund mitte der Einsatzleitungswagen. Dazu 5 Straßenlampen und (nicht sichtbar) Soundplatinen.

Unterhalb der Platte sieht es - und das ist ganz typisch für mich - chaotisch aus. Dort habe ich folgendes montiert, hier meine Bauteile:

1	Hauptplatine MobaLedLib mit 2 Arduino Nano
2	Single-LED Connector-Platinen MobaLedLib
3	Soundplatinen MobaLedLib mit JQ6500 Soundmodulen
3	Lautsprecher
1	Pushbutton-Platine MobaLedLib
2	Relais-Platinen (AZ-Delivery)
1	Spannungswandler (Eigenkonstruktion, von 17 V Wechselstrom-Ringleitung auf 5 V Gleichstrom)
1	MobaLedLib-Miniverteiler (Eigenkonstruktion)



Hier steht das Modul hochkant (um 90 Grad gedreht) auf meinem Arbeitstisch damit ich die Unterseite einfacher bearbeiten kann.

Und folgendes Szenario wollte ich gerne darstellen (das ist gleichzeitig der erste Entwurf für den zeitlichen Ablauf):

- per Relais das „brennende Finanzamt“ einschalten: Feuer-LEDs gehen an und der Rauchgenerator fängt an zu qualmen.
- gleichzeitig wird die Straßenbeleuchtung (5 Straßennamen) eingeschaltet, außerdem die Beleuchtung der Schmiede („belebtes Haus“)
- nach einiger Zeit der Rauchentwicklung wird die Beleuchtung des Spritzenhauses eingeschaltet und der Sound „Alarm für Löschzug 1, bitte komplett ausrücken“ wird abgespielt
- nacheinander gehen die Blaulichter der Einsatzfahrzeuge und die Frontlichter an, gleichzeitig werden verschiedene Martinshorn-Sounds abgespielt
- nach einiger Zeit (des Löschens des Brandes) erklingt der Sound „Das Feuer ist gelöscht, bitte wieder einrücken“
- das „brennende Finanzamt“ wird wieder ausgeschaltet
- die Martinshorn-Sounds, die Blaulichter und die Frontlichter gehen aus
- zum Schluss werden die Lichter in der Schmiede und im Spritzenhaus und auch die Straßenbeleuchtung ausgeschaltet

Das ganze sollte über einen am Anlagenrand befindlichen Pushbutton ausgelöst werden und automatisch ablaufen. Dieser Pushbutton soll - wie die anderen bei mit installierten Knöpfe - im Ruhezustand abgedunkelt grün leuchten und sobald er gedrückt wird heller grün blinken bis die Aktion beendet ist.

ACHTUNG:

Bei den folgenden Texten zur Programmierung und zum Einsatz der MobaLedLib gehe ich davon aus, dass die geneigte Leserschaft hier schon Vorkenntnisse hat. Die Grundlagen vom Programm-Generator oder Pattern-Konfigurator werde ich hier nicht abhandeln sondern einfach nur mein Vorgehen schildern. Die MLL-Spezialisten, Computerfachleute, Programmierer und Spezis mögen bitte entschuldigen, aber oft reicht mein elektronisches Verständnis und auch meine Programmierfähigkeiten nicht aus, das letzte Bit einzusparen oder alles absolut perfekt zu machen. Ich bitte dies zu entschuldigen.

Programmierung MobaLedLib

Zuerst einmal habe ich die beiden Relaisplatinen, die 3 Soundplatinen und die beiden LED-Connector-Platinen (hier habe ich alle einzelnen LEDs angeschlossen) wie gewohnt im ProgGenerator angelegt und diese Datei auf den Arduino übertragen. So konnte ich testen, ob alles funktioniert - also die LEDs, die Sounds usw. Testweise hatte ich diese ansteuerbar per DCC-Adresse angelegt. Alle Tests waren erfolgreich, jede einzelne LED kann ich ein- und ausschalten und auch jeden Sound auslösen. Da ich aber die Blaulichter der verschiedenen Einsatzfahrzeuge unterschiedlich blinken lassen möchte habe ich hierfür einmal z. B. das Blinklicht des Bahnübergangs genutzt, einmal ein eigenes Pattern angelegt und weitere Möglichkeiten der MLL genutzt.

Da ich dann einen zeitlichen Ablauf verschiedener „Steuerimpulse“ mit der MobaLedLib nicht abbilden konnte habe ich längere Zeit in der „MobaLedLib-Tageszeitung“ im Stummiforum gesucht. Plötzlich fiel es mir wie Schuppen aus den Augen: ich benötige ein sogenanntes „Schrittschaltwerk“.

Mit meinen Worten beschrieben handelt es sich hier um eine Pattern-Datei, die nach definierten Zeiten die Helligkeitswerte einer LED erzeugt. Also z.B. 12 Sekunden lang Helligkeit 2, dann 2 Sekunden Helligkeit 3, dann 5 Sekunden Helligkeit 4 usw. Nachdem ich händisch - also mit Stift und Papier - einen Ablaufplan für mein Szenario erstellt hatte konnte ich relativ genau festlegen, wie viele Sekunden eine Helligkeitsstufe dauern muss. Dabei steht jede Helligkeitsstufe später für das Auslösen einer Funktion (brennendes Finanzamt einschalten, Straßenlicht einschalten, Blaulichter einschalten, Sound einschalten usw.).

Dieses Pattern wird dann in den ProgGenerator übertragen. Dem ProgGenerator muss ich aber noch irgendwie mitteilen, welche Helligkeitswerte für welche Funktion genutzt werden sollen. Das geht über mehrere Zeilen „LED-Werte als Variablen“ (LED_to_Var“). Hier wird z.B. definiert, dass die Variable mit dem Namen „finanzamt“ dem Helligkeitswert 2 entspricht oder die Variable „sirene“ dem Wert 3 usw. Für mich laienhaft übersetzt bedeutet dies: bevor das Pattern die Helligkeit 3 sendet (welches eine Variable mit dem Namen „sirene“ einschaltet) wird 15 Sekunden lang die Helligkeit 2 gesendet und somit die Variable mit dem Namen „finanzamt“ eingeschaltet. Da der Rauchgenerator eine gewisse Zeit benötigt um den Rauch zu erzeuge hatte ich nach dem Einschalten des „finanzamt“ mehrere Sekunden gewartet bis der Sound „Sirene“ ausgelöst werden soll.

Jetzt muss ich nur noch dem ProgGenerator mitteilen, welche Helligkeit für welche Variable gedacht ist. Dies geht mit den Befehlen „LED-Werte als Variable“ (LED_to_Var). Hier muss bei der Definition der Variablenname, z. B. „finanzamt“ eingegeben werden und der Operator („ist gleich“, „ist größer als“, „ist kleiner als“ usw.) und der Vergleichswert (also der Helligkeitswert aus dem Pattern) definiert werden. Zum

Beispiel bei mir: „finanzamt“ „größer als“ „1“ bedeutet: bei einem Pattern-Helligkeitswert größer als 1 wird die Variable/Funktion „finanzamt“ eingeschaltet.

Damit das ganze funktioniert muss ich jetzt noch die Namen der Variablen in den richtigen Zeilen statt der DCC-Nummer eintragen. Bedeutet: in der Zeile, die bei meinem Test mittels DCC das brennende Finanzamt eingeschaltet hatte ersetze ich in der Spalte „Adresse oder Name“ die DCC-Nummer durch den Variablenname „finanzamt“. Den Eintrag unter Typ (bisher „rot“) lösche ich komplett in der Zeile. Das mache ich bei allen Zeilen, die durch das Pattern bzw. durch die LED_to_Var-Definitionen geschaltet werden sollen und: fast fertig.

Leider doch nicht: durch das Einfügen der Pattern-Zeile im ProgGenerator haben sich die Nummern der LEDs und Sounds verschoben und sind nicht mehr richtig ansteuerbar. Dies kann verhindert werden, indem ein sogenannter „virtueller LED-Bus“ angelegt wird. Dafür gibt es im ProgGenerator den Befehl „Pins LED Bus definieren“. Hier muss als Wert „6 A4 V“ eingetragen werden (siehe Zeile 9 im Screenshot). In der Zeile 11 des Screenshots wird dann das Pattern übernommen und hierbei wird als LED-Bus die Nummer „2“ eingetragen (ganz wichtig - siehe Screenshot auf Seite 12). Somit zählt das Pattern in der LED-Reihenfolge nicht mit und alles ist wieder in richtiger Reihenfolge.

Jetzt noch einen Pushbutton mit einem MonoFlop (siehe Zeilen 6 und 7 des Screenshots) definieren und: FERTIG

Abschließend muss ich noch ein bisschen mit den Zeiten ausprobieren: im Pattern lässt sich in der Zeitzeile jeder Wert (in Millisekunden oder Sekunden) eingeben, wie lange läuft das Szenario insgesamt (wichtig für die Blinkfunktion des Pushbuttons) usw.

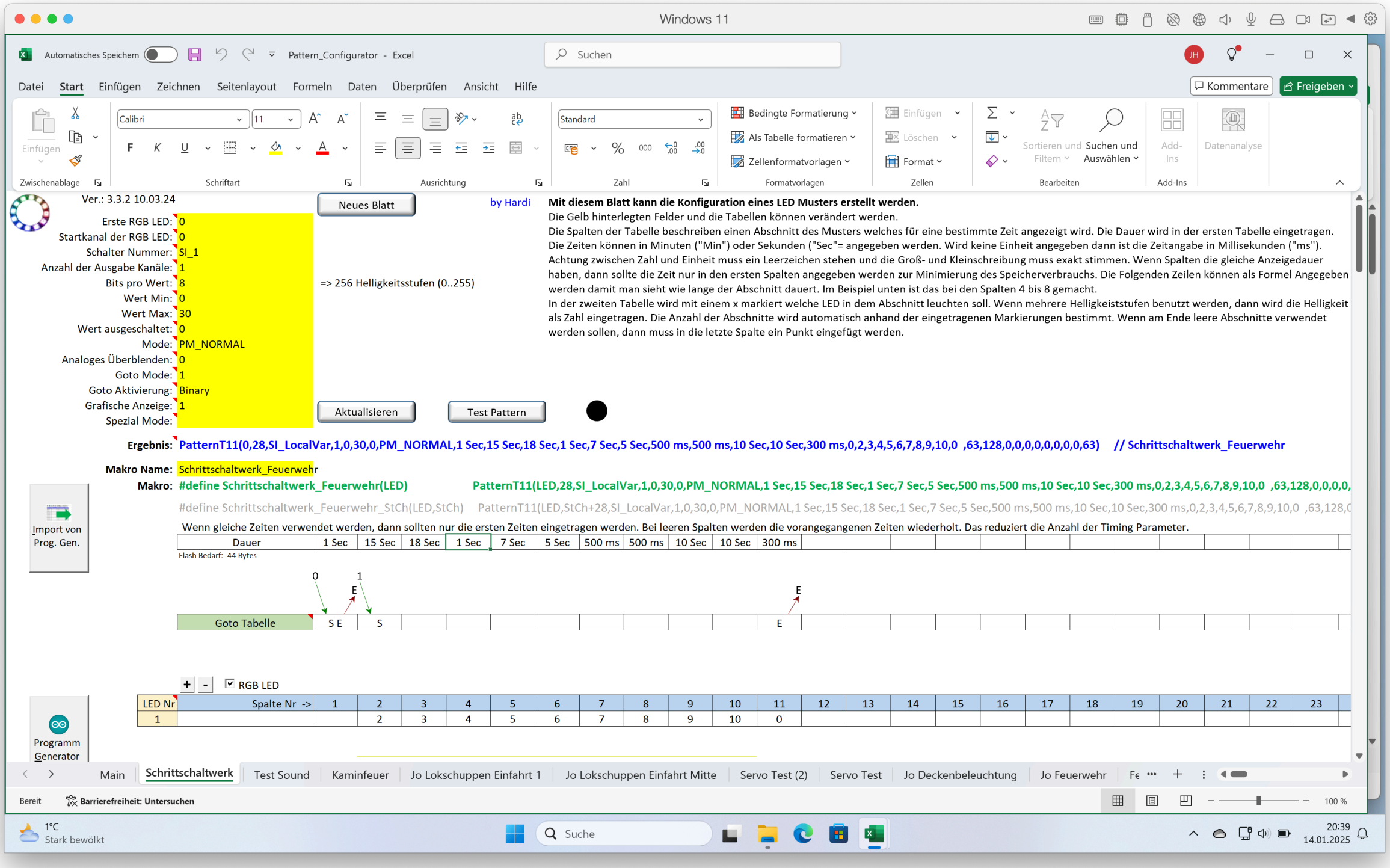
Aber nachdem meine Ehefrau aufgrund der vielen Sirenen und Martinshörner sehr genervt ist habe ich alles genau so eingestellt, wie ich es haben möchte.

Ein einfacher Druck auf den Pushbutton löst jetzt die gesamte Aktion aus und bringt mir wieder etwas mehr Abwechslung auf die Modellbahn. Mal sehen, wenn die Enkel kommen, was die dazu sagen . . .

Und jetzt viel Spaß beim Ausprobieren!

Übrigens habe ich auf den folgenden Seiten Screenshots des Pattern und des ProgGenerators zur Information. Interessierte können Sie die Pattern-Configurator-Datei und die ProgrammGenerator-Datei downloaden. Links hierzu siehe Seite 13.

So sieht das „Schrittschaltwerk“ im PatternConfigurator aus.



Windows 11

Automatisches Speichern — Prog_Generator_MobaLe... • Auf "diesem PC" gespeichert — Suchen

Datei | Start | Einfügen | Zeichnen | Seitenlayout | Formeln | Daten | Überprüfen | Ansicht | Hilfe — Kommentare — Freigeben

Einfügen | Arial | 10 | Standard | Bedingte Formatierung | Als Tabelle formatieren | Zellenformatvorlagen | Einfügen | Löschen | Format | Sortieren und Filtern | Suchen und Auswählen | Add-Ins | Datenanalyse

Zwischenablage | Schriftart | Ausrichtung | Zahl | Formatvorlagen | Zellen | Bearbeiten | Add-Ins

Dialog | Z. Arduino schicken | Zeile einfügen | Lösche Zeilen | Verschiebe Zeilen | Kopiere Zeilen | Aus- oder Einblenden | Alle Einblenden | Lösche Tabelle | Optionen | Help

Ver. 3.3.2B by Hardi & Jürgen

-COM4 AutoDet --board arduino:avr:nano:cpu=atmega328old --pref programme — COM? AutoDet --board arduino:avr:nano:cpu=atmega328old --pref programmer=arduino:arduinoisp

Aktiv	Filter	Adresse oder Name	Typ	Start-wert	Beschreibung	Vert eiler-Nr	Stecker-Nu	Icon	Name	Beleuchtung, Sound, oder andere Effekte	Start LedNr	LEDs	InCnt	Loc InCh	LED/ Sound Kanal
✓									Heartbeat LED	RGB_Heartbeat(#LED)	0-0	1	0	0	0
✓		SwitchB1			PushButton2 schaltet die Variable KnopfA0, 88 s Blinken				Taster beleuchtet, 1 Funktion	PushButton_w_LED_BL_0_1(#LED, C1, #InCh, KnopfA0, 1, 0, 1, 0	1-0	C1-1	1	0	1
✓		KnopfA1			schaltet schrittschaltwerk1 für 88 Sekunden ein				Mono-Flop	MonoFlop(schrittschaltwerk1, #InCh, 88 Sek)			1	0	
✓									Pins LED Bus definieren	// Set_LED_OutpPinLst(6 A4 V)			0	0	
✓		schrittschaltwerk1			Schrittschaltwerk_Feuerwehr (pc)				Muster Pattern_Configurator	// Activation: BinaryBin_InCh_to_TmpVar(#InCh, 1)PatternT11(	2-0	C1-1	1	0	2
✓									LED-Werte als Variable	LED_to_Var(finanzamt, 0, >, 1)			0	0	
✓									LED-Werte als Variable	LED_to_Var(schmiede, 0, >, 1)			0	0	
✓									LED-Werte als Variable	LED_to_Var(sirene, 0, =, 3)			0	0	
✓									LED-Werte als Variable	LED_to_Var(alarm, 0, >, 3)			0	0	
✓									LED-Werte als Variable	LED_to_Var(licht1, 0, >, 4)			0	0	
✓									LED-Werte als Variable	LED_to_Var(licht2, 0, >, 5)			0	0	
✓									LED-Werte als Variable	LED_to_Var(licht3, 0, >, 6)			0	0	
✓									LED-Werte als Variable	LED_to_Var(licht4, 0, >, 7)			0	0	
✓									LED-Werte als Variable	LED_to_Var(licht5, 0, >, 8)			0	0	
✓									LED-Werte als Variable	LED_to_Var(licht6, 0, >, 9)			0	0	
✓									LED-Werte als Variable	LED_to_Var(ende, 0, =, 10)			0	0	
✓		licht1			LED 1 rot+grün Feuerwehr-Jochem (pc)	1			Muster Pattern_Configurator	APatternT1(#LED,4,#InCh,2,0,255,0,0,40 ms,3,0,51,0,0,204,0,1	0-1	C1-2	1	0	0
✓					LED Nummer manipulieren -1	1			LED Nummer manipulieren	// Next_LED(-1)	0-2	-1	0	0	0
✓		licht1			LED1, blau - Licht Unimog 1	1			LED einstellbar	Const(#LED, C3, #InCh, 0, 30)	0-1	C3-3	1	0	0
✓		licht1			LED2, rot - Licht Unimog 2	1			LED einstellbar	Const(#LED, C1, #InCh, 0, 30)	0-2	C1-1	1	0	0
✓		licht2			LED2, grün - Licht Passat 1	1			LED einstellbar	Const(#LED, C2, #InCh, 0, 20)	0-2	C2-2	1	0	0
✓		licht2			LED2, blau - Licht Passat 2	1			LED einstellbar	Const(#LED, C3, #InCh, 0, 20)	0-2	C3-3	1	0	0
✓		licht4			LED 3 rot+grün Blaulicht Passat	1			Andreaskreuz	Andreaskreuz(#LED, C1, #InCh)	0-3	C1-2	1	0	0
✓					LED Nummer manipulieren -1	1			LED Nummer manipulieren	// Next_LED(-1)	0-4	-1	0	0	0
✓		licht3	AnAus		LED3, blau - Licht Leiterwagen 1	1			LED einstellbar	Const(#LED, C3, #InCh, 0, 25)	0-3	C3-3	1	0	0
✓		licht3	AnAus		LED4, rot - Licht Leiterwagen 2	1			LED einstellbar	Const(#LED, C1, #InCh, 0, 25)	0-4	C1-1	1	0	0
✓		licht5			LED4, grün - Blaulicht Leiterwagen				LED einstellbar	Const(#LED, C2, #InCh, 0, 255)	0-4	C2-2	1	0	0
✓		9	AnAus 0		LED4, blau - leer				LED einstellbar	Const(#LED, C3, #InCh, 0, 255)	0-4	C3-3	1	0	0
✓		10	AnAus 0		LED5, rot - leer				LED einstellbar	Const(#LED, C1, #InCh, 0, 127)	0-5	C1-1	1	0	0
✓		11	AnAus 0		LED 5, grün - leer				LED einstellbar	Const(#LED, C2, #InCh, 0, 127)	0-5	C2-2	1	0	0
✓		12	AnAus 0		LED 5 blau - leer				LED einstellbar	Const(#LED, C3, #InCh, 0, 127)	0-5	C3-3	1	0	0
✓		13	AnAus 0		LED 6, rot - leer				LED einstellbar	Const(#LED, C1, #InCh, 0, 127)	0-6	C1-1	1	0	0
✓		14	AnAus 0		LED 6, grün - leer				LED einstellbar	Const(#LED, C2, #InCh, 0, 127)	0-6	C2-2	1	0	0
✓		finanzamt			LED 6, blau - Finanzamt				LED eins	(#LED, C3, #InCh, 0, 255)	0-6	C3-3	1	0	0
✓		licht2			LED 7, rot - Blaulicht 1 Kranwagen	1			Blaulicht	ight2(#LED, C1, #InCh)	0-7	C1-1	1	0	0
✓		licht2			LED 7, grün - Blaulicht 2 Kranwagen	1			Blaulicht	ight1(#LED, C2, #InCh)	0-7	C2-2	1	0	0
✓		licht5			LED 7, blau - Suchscheinwerfer Kranwagen	1			LED eins	(#LED, C3, #InCh, 0, 255)	0-7	C3-3	1	0	0
✓		licht4			LED 8, rot - Licht Kranwagen 1	1			LED eins	(#LED, C1, #InCh, 0, 25)	0-8	C1-1	1	0	0

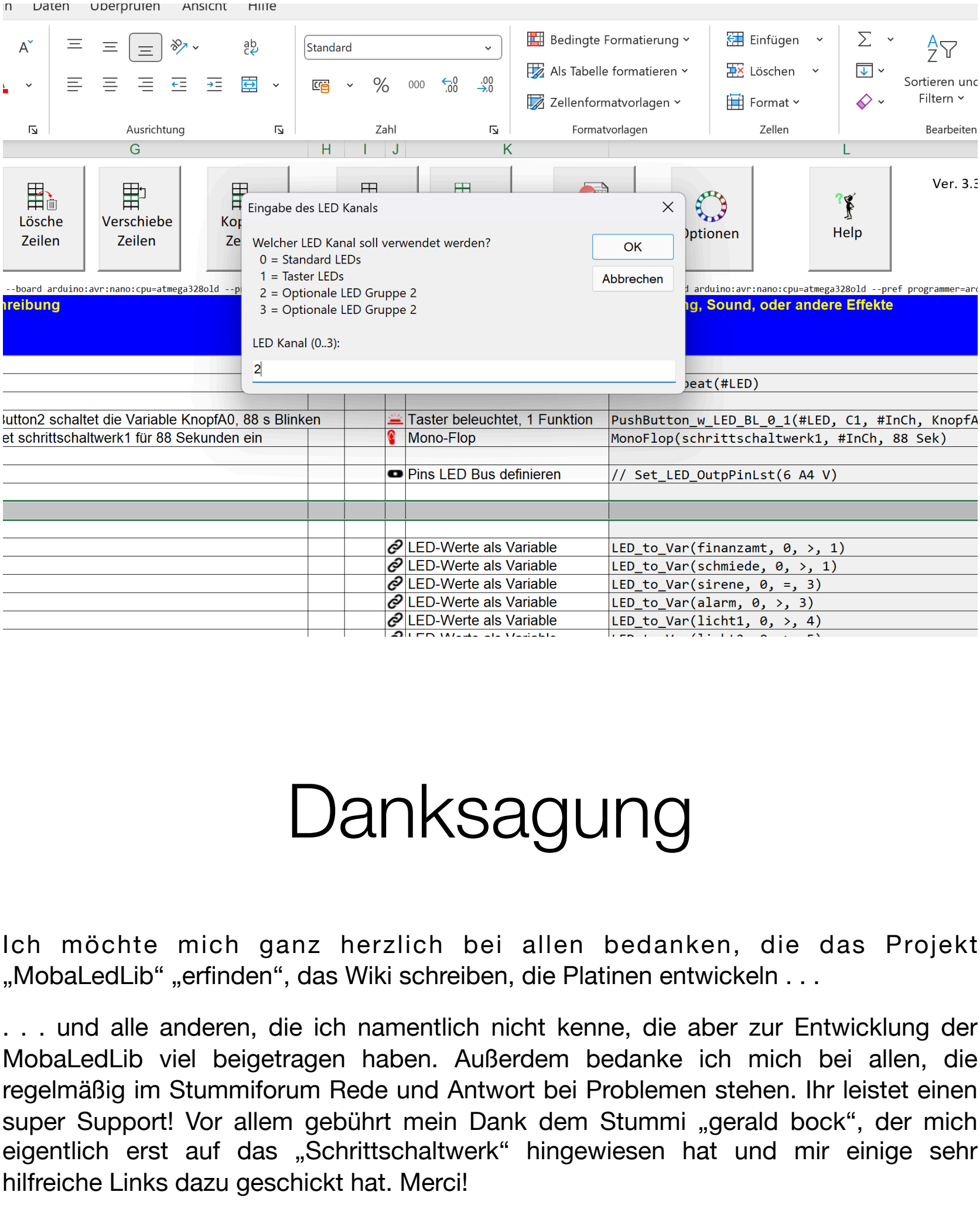
Win-Dateien – Datei-Ex...

49	✓		licht4			LED 8, rot - Licht Kranwagen 1	1		LED einstellbar	Const(#LED, C1, #InCh, 0, 25)	0-8	C1-1	1	0	0
50	✓		licht4			LED 8, grün - Licht Kranwagen 2	1		LED einstellbar	Const(#LED, C2, #InCh, 0, 25)	0-8	C2-2	1	0	0
51	✓		19	AnAus 0		LED 8, blau - leer			LED einstellbar	Const(#LED, C3, #InCh, 0, 250)	0-8	C3-3	1	0	0
52																
53						**Platine 2:**										
54	✓		finanzamt			LED1 R+G+B, LED2 R+G+B, Led 3 R			Straßenbeleuchtung	GasLights(#LED, #InCh, GAS_LIGHT1, GAS_LIGHT2, GAS_LIGHT3, GA	0-9	3	1	0	0
55	✓		27	AnAus 0		LED 3 - grün			LED einstellbar	Const(#LED, C2, #InCh, 0, 255)	0-12	C2-2	1	0	0
56	✓		28	AnAus 0		LED 3 - blau			LED einstellbar	Const(#LED, C3, #InCh, 0, 255)	0-12	C3-3	1	0	0
57	✓		licht4			LED 4 - rot - Licht Bus vorne linkw			LED einstellbar	Const(#LED, C1, #InCh, 0, 30)	0-13	C1-1	1	0	0
58	✓		licht4			LED 4 - grün - Licht Bus vorne rechts			LED einstellbar	Const(#LED, C2, #InCh, 0, 30)	0-13	C2-2	1	0	0
59	✓		29	AnAus 0		LED 4 - blau - leer			LED einstellbar	Const(#LED, C3, #InCh, 0, 127)	0-13	C3-3	1	0	0
60	✓		32	AnAus 0		LED 5 - rot - leer			LED einstellbar	Const(#LED, C1, #InCh, 0, 120)	0-14	C1-1	1	0	0
61	✓		33	AnAus 0		LED 5 - grün - leer			LED einstellbar	Const(#LED, C2, #InCh, 0, 120)	0-14	C2-2	1	0	0
62	✓		34	AnAus 0		LED 5 - blau - leer			LED einstellbar	Const(#LED, C3, #InCh, 0, 255)	0-14	C3-3	1	0	0
63	✓		35	AnAus 0		LED 6 - rot - leer			LED einstellbar	Const(#LED, C1, #InCh, 0, 255)	0-15	C1-1	1	0	0
64	✓		alarm			LED 6 - grün - Torlampe links			LED einstellbar	Const(#LED, C2, #InCh, 0, 255)	0-15	C2-2	1	0	0
65	✓		alarm			LED 6 - blau - Torlampe rechts			LED einstellbar	Const(#LED, C3, #InCh, 0, 255)	0-15	C3-3	1	0	0
66	✓		licht4			LED 7 - rot - Blaulicht Bus beide vorne			Blaulicht 1	BlueLight1(#LED, C1, #InCh)	0-16	C1-1	1	0	0
67	✓		licht4			LED 7 - grün - Blaulicht Bus hinten			Blaulicht 2	BlueLight2(#LED, C2, #InCh)	0-16	C2-2	1	0	0
68	✓		39	AnAus 0		LED 7 - blau - belegt aber nicht benötigt			LED einstellbar	Const(#LED, C3, #InCh, 0, 30)	0-16	C3-3	1	0	0
69																
70																
71	✓		licht1			Soundmodul 1 - Horn 1			Titel 1 abspielen	Sound_JQ6500_Seq1(#LED, #InCh)	0-17	C1-1	1	0	0
72	✓		1012	Rot		Horn 2			Titel 2 abspielen	Sound_JQ6500_Seq2(#LED, #InCh)	0-17	^ C1-1	1	0	0
73	✓		licht3			Horn 3			Titel 3 abspielen	Sound_JQ6500_Seq3(#LED, #InCh)	0-17	^ C1-1	1	0	0
74	✓		1014	Rot					Titel 4 abspielen	Sound_JQ6500_Seq4(#LED, #InCh)	0-17	^ C1-1	1	0	0
75	✓		1015	Rot					Titel 5 abspielen	Sound_JQ6500_Seq5(#LED, #InCh)	0-17	^ C1-1	1	0	0
76	✓		1051	Rot					Lauter	Sound_JQ6500_IncVol(#LED, #InCh, 1)	0-17	^ C1-1	1	0	0
77	✓		1052	Rot					Leiser	Sound_JQ6500_DecVol(#LED, #InCh, 1)	0-17	^ C1-1	1	0	0
78																
79	✓		licht4			Soundmodul 2 - Horn 1			Titel 1 abspielen	Sound_JQ6500_Seq1(#LED, #InCh)	0-18	C1-1	1	0	0
80	✓		licht2			Horn 2			Titel 2 abspielen	Sound_JQ6500_Seq2(#LED, #InCh)	0-18	^ C1-1	1	0	0
81	✓		1023	Rot		Alarmierung			Titel 3 abspielen	Sound_JQ6500_Seq3(#LED, #InCh)	0-18	^ C1-1	1	0	0
82	✓		1024	Rot					Titel 4 abspielen	Sound_JQ6500_Seq4(#LED, #InCh)	0-18	^ C1-1	1	0	0
83	✓		1025	Rot					Titel 5 abspielen	Sound_JQ6500_Seq5(#LED, #InCh)	0-18	^ C1-1	1	0	0
84	✓		1053	Rot					Lauter	Sound_JQ6500_IncVol(#LED, #InCh, 1)	0-18	^ C1-1	1	0	0
85	✓		1054	Rot					Leiser	Sound_JQ6500_DecVol(#LED, #InCh, 1)	0-18	^ C1-1	1	0	0
86																
87	✓		ende			Soundmodul 3 - fertig, einrücken			Titel 1 abspielen	Sound_JQ6500_Seq1(#LED, #InCh)	0-19	C1-1	1	0	0
88	✓		alarm			Alarmierung			Titel 2 abspielen	Sound_JQ6500_Seq2(#LED, #InCh)	0-19	^ C1-1	1	0	0
89	✓		sirene			Sirene			Titel 3 abspielen	Sound_JQ6500_Seq3(#LED, #InCh)	0-19	^ C1-1	1	0	0
90	✓		1034	Rot					Titel 4 abspielen	Sound_JQ6500_Seq4(#LED, #InCh)	0-19	^ C1-1	1	0	0
91	✓		1035	Rot					Titel 5 abspielen	Sound_JQ6500_Seq5(#LED, #InCh)	0-19	^ C1-1	1	0	0
92	✓		1055	Rot					Lauter	Sound_JQ6500_IncVol(#LED, #InCh, 1)	0-19	^ C1-1	1	0	0
93	✓		1056	Rot					Leiser	Sound_JQ6500_DecVol(#LED, #InCh, 1)	0-19	^ C1-1	1	0	0
94																
95						**Haus Schmiede:**										
96	✓		schmiede			LED1 R - Licht Speicher vorne			LED einstellbar	Const(#LED, C1, #InCh, 0, 255)	0-20	C1-1	1	0	0
97	✓		schmiede			LED1 G - Licht Speicher vorne			LED einstellbar	Const(#LED, C2, #InCh, 0, 255)	0-20	C2-2	1	0	0
98	✓		schmiede			LED1 B - Licht Speicher vorne			LED einstellbar	Const(#LED, C3, #InCh, 0, 255)	0-20	C3-3	1	0	0
99	✓		schmiede			LED2 R - Licht außen links hinten			LED einstellbar	Const(#LED, C1, #InCh, 0, 255)	0-21	C1-1	1	0	0
100	✓		schmiede			LED2 G - Licht außen links hinten			LED einstellbar	Const(#LED, C2, #InCh, 0, 255)	0-21	C2-2	1	0	0
101	✓		schmiede			LED2 B - Licht außen links hinten			LED einstellbar	Const(#LED, C3, #InCh, 0, 255)	0-21	C3-3	1	0	0
102	✓		schmiede			LED3 R - Licht außen links vorne			LED einstellbar	Const(#LED, C1, #InCh, 0, 255)	0-22	C1-1	1	0	0
103	✓		schmiede			LED3 G - Licht außen links vorne			LED einstellbar	Const(#LED, C2, #InCh, 0, 255)	0-22	C2-2	1	0	0
104	✓		schmiede			LED3 B - Licht außen links vorne			LED einstellbar	Const(#LED, C3, #InCh, 0, 255)	0-22	C3-3	1	0	0
105	✓		schmiede			LED4 R - Licht außen rechts vorne			LED einstellbar	Const(#LED, C1, #InCh, 0, 255)	0-23	C1-1	1	0	0
106	✓		schmiede			LED4 G - Licht außen rechts vorne			LED einstellbar	Const(#LED, C2, #InCh, 0, 255)	0-23	C2-2	1	0	0
107	✓		schmiede			LED4 B - Licht außen rechts vorne			LED einstellbar	Const(#LED, C3, #InCh, 0, 255)	0-23	C3-3	1	0	0
108	✓		schmiede			LED 5 - R+G+B + LED 6 R+G+B			Belebtes Haus	HouseT(#LED, #InCh, 0, 2, 1, 3, NEON_LIGHT, NEON_LIGHT)	0-24	2	1	0	0
109	✓		schmiede			LED7 RGB - Kaminfeuer außen - Chimney			Belebtes Haus	House(#LED, #InCh, 1, 1, ROOM_CHIMNEY)	0-26	1	1	0	0
110																
111	✓					Letzter Heartbeat zur Kontrolle auf Hauptplatine			Heartbeat LED	RGB_Heartbeat(#LED)	0-27	1	0	0	0
112																
113																
114																
115																
116																

Jochem Feuerwehr | Jochem Feuerwehr (2) | Jochem Feuerwehr (3) | Jochem Feuerwehr (4) | Jochem PushButton | DCC Soundplatine Test | DCC Jo

Bereit | Barrierefreiheit: Untersuchen | 100 %

1°C Stark bewölkt | Suche | 20:37 14.01.2025



Danksagung

Ich möchte mich ganz herzlich bei allen bedanken, die das Projekt „MobaLedLib“ „erfinden“, das Wiki schreiben, die Platinen entwickeln . . .

. . . und alle anderen, die ich namentlich nicht kenne, die aber zur Entwicklung der MobaLedLib viel beigetragen haben. Außerdem bedanke ich mich bei allen, die regelmäßig im Stummiforum Rede und Antwort bei Problemen stehen. Ihr leistet einen super Support! Vor allem gebührt mein Dank dem Stummi „gerald bock“, der mich eigentlich erst auf das „Schrittschaltwerk“ hingewiesen hat und mir einige sehr hilfreiche Links dazu geschickt hat. Merci!

Vielen Dank!

Links

Und ganz zum Schluss noch einige Links:

MobaLedLib im Stummiforum:

https://www.stummiforum.de/viewtopic.php?f=7&t=165060

MobaLedLib-Wiki:

https://wiki.mobaledlib.de

Und hier können Interessierte meine Dateien herunterladen. Diese sind gezippt und müssen nach dem Download noch entpackt werden. Anschließend im PatternConfigurator oder im ProgGenerator auf das „MLL-Farbrad“ klicken und unter „Dateien“ die Dateien in das jeweilige Programm laden. Ein einfacher Doppelklick auf meine Dateien reicht nicht aus. Viel Spass!

PatternConfigurator:

www.familieheinen.name/moba/Pattern_Schrittschaltwerk_Example.MLL_pcf.zip

ProgGenerator:

www.familieheinen.name/moba/Prog_Gen_Schrittschaltwerk.MLL_pgf.zip

Stand 1/2025 - Version 2

Rückfragen zu dieser Schrift bitte nur per eMail an:
jochem@familieheinen.name