# Anleitung zum Programm-Generator

Diese Anleitung bezieht sich auf die Version **3.1.0A** vom 05.01.2022 und neuer. (Aktuelle Version ist 3.3.0)
— *Michael 2022/01/27 14:43*

## Was ist der Programm Generator

Mit diesem Programm können die Beleuchtungs- und andere Effekte für eine Modelleisenbahn ganz einfach erstellt und verwaltet werden.

- Mit dem Arduino nano können bis zu 256 RGB-LEDS bzw. 768 Einzel-LEDs verwaltet werden.
- Mit dem ESP32-30pin (7 Kanäle) können je nach verwendeter Hauptplatine bis zu 14336 RGB-LEDS bzw. 43008 Einzel-LEDs verwaltet werden.
- Mit dem ESP32-38pin (8 Kanäle) können je nach verwendeter Hauptplatine bis zu 16384 RGB-LEDS bzw. 49152 Einzel-LEDs verwaltet werden.

Über eine einfach zu bedienende Benutzeroberfläche kann man die gewünschten Funktionen Auswählen und Konfigurieren. Es steht eine Vielzahl von Befehlen zur Verfügung. Der Wichtigste ist sicherlich das **„Belebte Haus“**. Man kann aber auch Andreaskreuze, Signale, Ampeln, Sounds, Servos und vieles mehr mit nur einem Klick auswählen.

Die Effekte können automatisch gestartet oder über eine Zentrale aktiviert werden. Dazu werden momentan folgenden Protokolle/Busse unterstützt: **DCC**, **Selectrix** und der **Märklin CAN Bus**.

Das alles wird in einer übersichtlichen Tabelle verwaltet. Hier kann man ganz einfach Zeilen einfügen, verschieben, kopieren und nach seinen Bedürfnissen anpassen.

Die Konfiguration kann von dort direkt zum Arduino geschickt und ausprobiert werden.

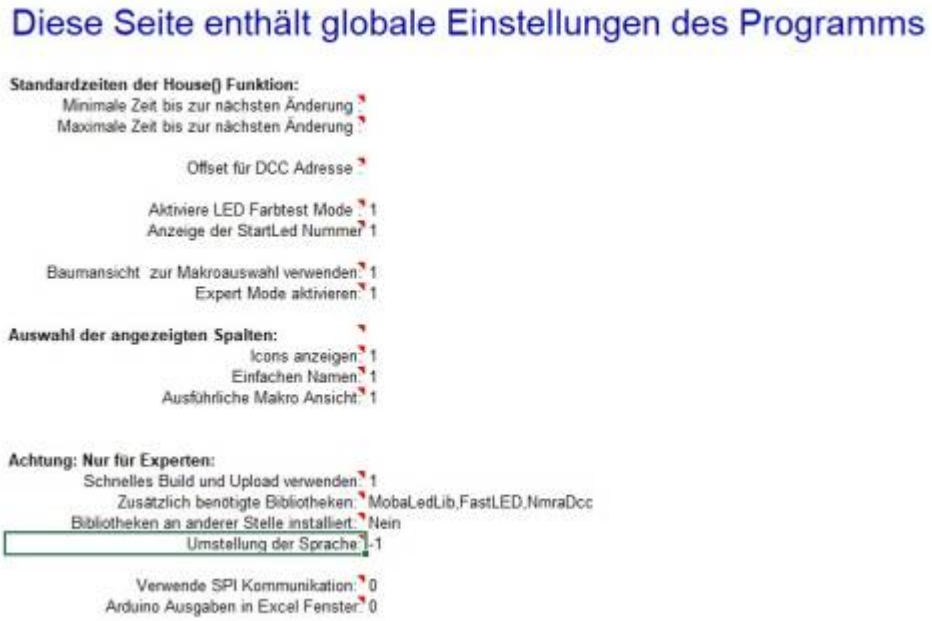
Die Programmerzeugung geschieht komplett im Hintergrund. Der Benutzter muss keinerlei Programmiererfahrung haben.

### Verfügbare Sprachen

Der Programm-Generator wurde durch fleißige Nutzer, bereits in sieben Sprachen übersetzt.
Aktuell stehen folgenden Sprachen zur Verfügung.

- Deutsch (0)
- Englisch (1)
- Niederländisch (2)
- Französisch (3)
- Italienisch (4)
- Spanisch (5)
- Dänisch (6), übersetzt von Saryk

Die Zahl in der Klammer ist die ID für die „Config-Seite“.
Der Wert „-1“ aktiviert die automatische Sprachauswahl, anhand der Systemsprache.
Wenn die Sprache nicht vorhanden ist, erfolgt die Ausgabe in „Englisch“.




Wenn jemand den Programm-Generator in weitere Sprachen übersetzen möchte, kann sich gerne an Hardi im Stummiforum wenden.

---

# Wichtige Hinweise

## Excel-Versionen

Der Programm-Generator funktioniert nur mit Microsoft Excel in der Desktop-Version.
Die Verwendung von OpenSource-Varianten[1)], sowie der „Web Version von Excel“[2)], ist wegen der enthaltenen Makros nicht fehlerfrei möglich.
Aktuell werden folgenden Versionen unterstützt:

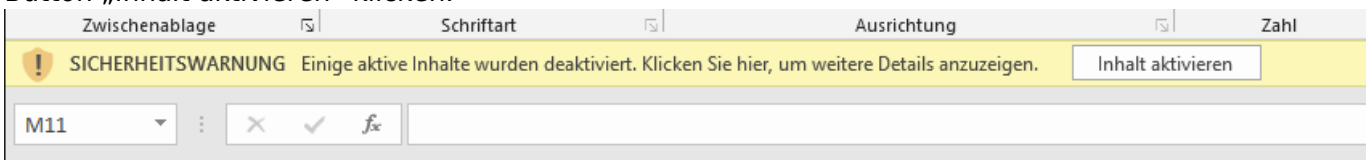
- ~~Excel 2010~~ (wird nicht mehr unterstützt in der neusten Version)
- Excel 2013
- Excel 2016
- Excel 2019
- Excel 2021
- Excel 365 Desktop-App

---

# Erster Start

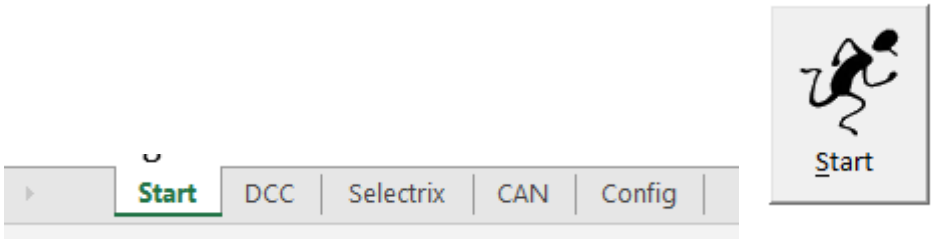
Beim ersten Start wird von Excel eine Warnung ausgeben, dass sich aktive Elemente („Makros“) in der Tabelle befinden und das diese aus Sicherheitsgründen deaktiviert wurden. Um den Programm-

Generator nutzen zu können ist das Aktiveren der Inhalte notwendig. Dazu einfach oben auf den Button „Inhalt aktivieren“ klicken.

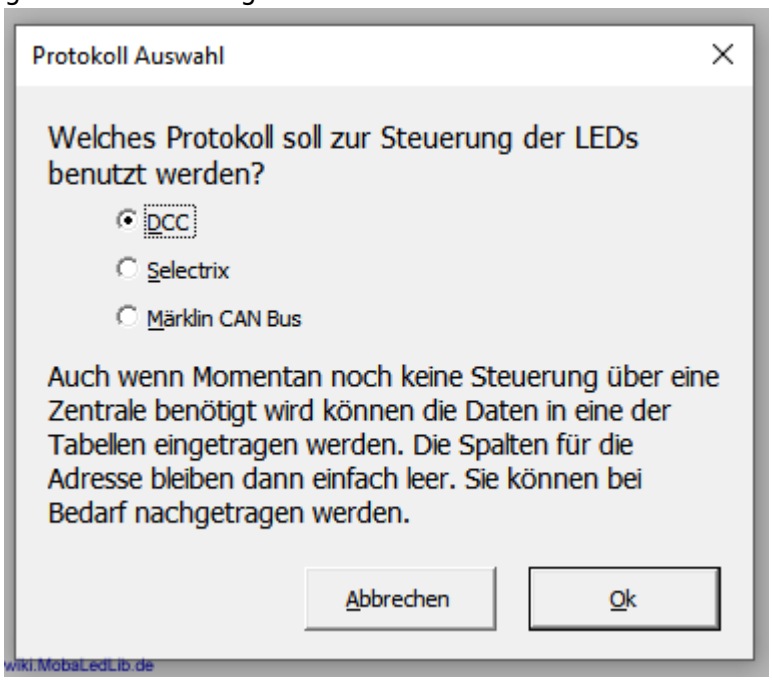



## Auswahl des Systems zur Einbindung

Je nachdem welches Steuerungssystem eingesetzt wird kann man am unteren Rand, das gewünschte System auswählen oder einfach auf den Button „Start“ klicken.




Bei einem Klick auf den Button „Start“ öffnet sich ein Auswahlmenü, welches ermöglicht eines der verschiedenen Systeme zur Steuerung zu verwenden. Ein Wechsel auf ein anderes System ist jederzeit möglich. Für Anfänger wird das System „DCC“ empfohlen, da es sehr einfach ist und die größte Verbreitung hat.




Die Auswahl eines Steuerungssystems ist notwendig um die Beleuchtungen zu verwalten. Allerdings muss man keines besitzen um die MobaLedLib oder den Programm-Generator verwenden zu können.

### Verwendung "CAN"

Damit die Daten aus dem Programm-Generator an den Arduino hochgeladen werden können, muss bei der Verwendung von **CAN** unbedingt eine Verbindung zu einer aktiven CAN-Zentrale vorhanden sein, andernfalls schlägt der Upload fehl.

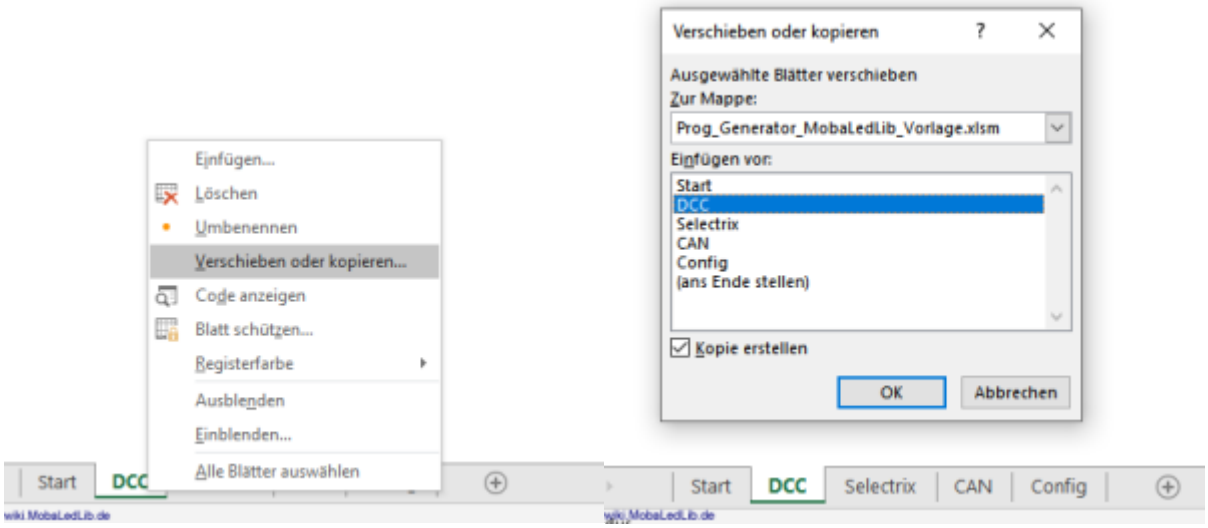
---

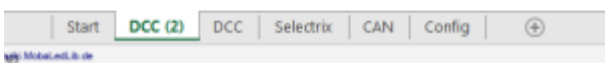
# Verwendung des Programm-Generators

Diese Anleitung erklärt das Verwenden des Tools anhand der Tabelle für die Steuerung per „DCC“. Die minimalen Abweichungen zu den Tabellen „Selectrix“ und „CAN“ werden jeweils im Anschluss erläutert, sofern es notwendig ist.

### Tabelle kopieren

Für die ersten Tests kann es hilfreich sein, nur eine Kopie einer Tabelle zu verwenden.
Um eine Kopie zu erstellen einfach einen Rechtsklick auf die gewünschte Tabelle machen und in dem sich öffnenden Dialog sagen, an welcher Stelle die Kopie eingefügt werden soll und das Excel eine Kopie erstellen soll. Andernfalls wird die Tabelle nur an den neuen Platz verschoben.

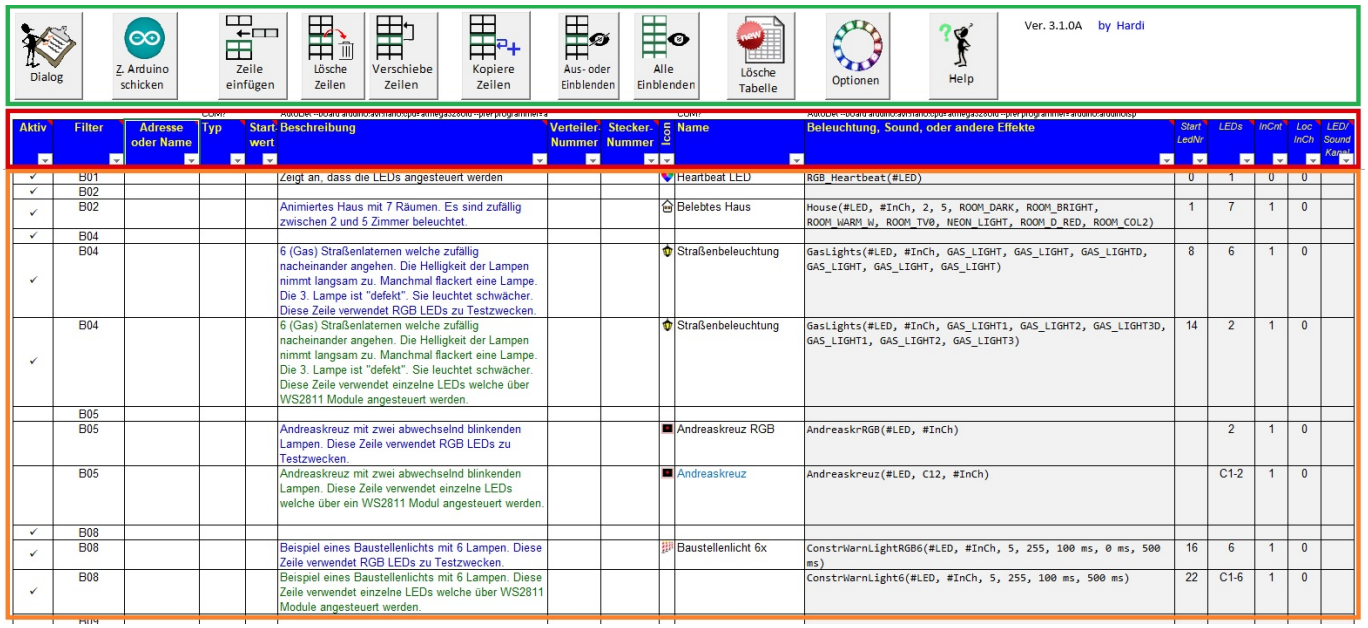


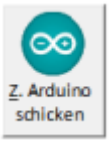

### Aufbau der Tabelle

Das Hauptfenster des Programm-Generators teilt sich in drei Hauptbereiche auf.

- Dem grünen Bereich für alle Buttons welche die Steuerung des Generators übernehmen.
- Der Filterliste, welche sich in dem roten Bereich befindet.
- sowie dem orangenen Bereich in dem alle Befehle gespeichert und verwaltet werden.



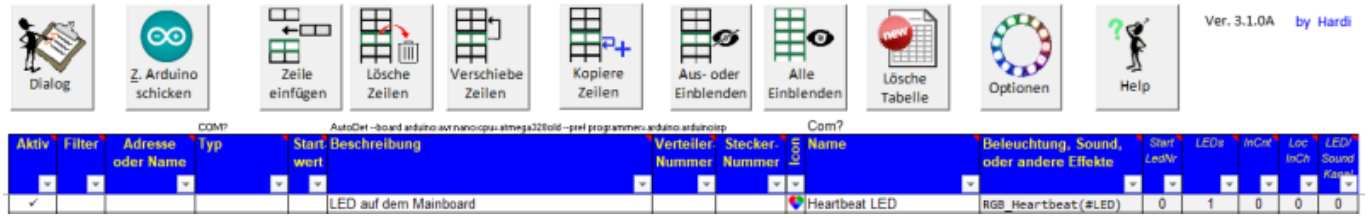

## Buttons

| Button | Beschreibung |
|---|---|
| | Der wichtigste Button von allen. Hierhinter verbirgt sich die meiste Magie und ermöglicht das Erstellen der Beleuchtungs- Sound und Bewegungseffekte. |
|  | wandelt die Befehle in C++-Code um und überträgt diese im Anschluss an den LED Arduino. |
| | fügt eine neue leere Zeile über der aktuell markierten Zeile ein |
| | löscht wie die Aufschrift bereits erklärt alle markierten Zeilen. Die Löschung erfolgt aber erst nach einer erneuten Bestätigung in dem aufgehendem Fenster. |
| | Hiermit können eine oder mehrere Zeilen in der Reihenfolge verschoben werden. Die ausgewählten Zeilen werden oberhalb des grünen Balkens eingefügt und das Programm berechnet die Reihenfolge der LEDs neu. Dies ist hilfreich wenn man z.B.: einen zusätzlichen Verteiler in einer vorhanden Verkabelung hinzufügt oder diesen woanders platziert. |
| | Hiermit lassen sich Zeilen kopieren für den Fall, das es z.B.: ein identisches oder ähnliches Beleuchtungskonzept für ein anderes Haus gibt. |

| Button | Beschreibung |
|---|---|
| | Hiermit lassen sich Zeilen aus- und auch wieder einblenden. Diese werden nicht gelöscht sondern nur versteckt. |
| | Dieser Button zeigt alle Zeilen, die vorher versteckt wurden, wieder an. |
| | löscht alles in dem Programmbereich und legt eine komplett neue Tabelle an. Das Programm fragt aber vorher ob es wirklich gewünscht ist.<br>Durch eine Filterliste ausgeblendete Zeilen werden nicht gelöscht. |
| | Öffnet das Menü für die Optionen, in dem man die Einstellungen für den LED Arduino und den DCC Arduino ändern kann. Es gibt auch die Möglichkeit, die Datei zu speichern oder die Beta Version der MobaLedLib zu installieren. |
| | Öffnet die Hilfeseite |
| Ver. 0.71 by Hardi | Zeigt die aktuelle Version des Programm-Generators an. |

## Filterliste

Die blauen Filterlisten ermöglichen es, die Anzahl der Zeilen zu verringern und nur bestimmte Zeilen anzuzeigen. Dies wird z.B.: gemacht um nur das Beispiel „B1“ beim ersten Aufrufen der Tabelle „DCC“, „Selectrix“ oder „CAN“ anzuzeigen.




### Erklärung der einzelnen Spalten

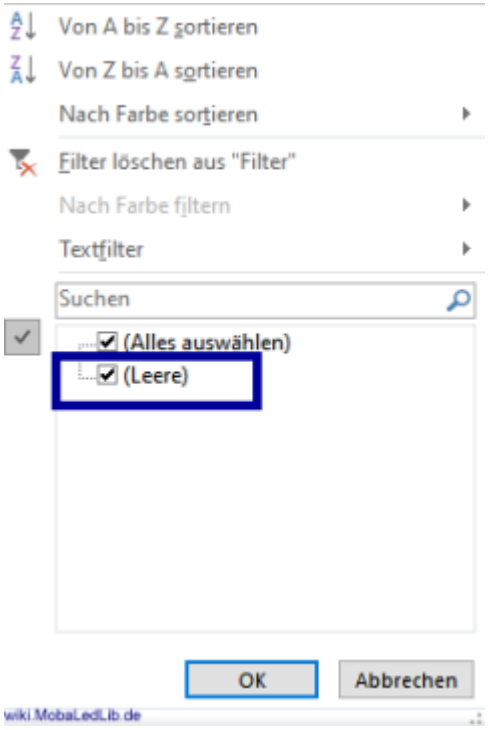
| Bezeichnung | Erklärung |
|---|---|
| Aktiv | Mit dieser Spalte kann man eine Zeile (zu Testzwecken) deaktivieren. Zeilen welche mit dem „Aus- oder Einblenden“ Knopf oder dem Autofilter ausgeblendet sind werden ebenfalls nicht benutzt. |
| Filter | Diese Spalte kann zum Filtern nach bestimmten Gruppen benutzt werden. Für jede Gruppe kann hier ein gleicher Wert eingetragen werden. Dann kann man mit dem Autofilter bestimmte Gruppen aktivieren. |
| Adresse oder Name | Hier wird die DCC Adresse zwischen 1 und 10240 eingetragen. Achtung nicht alle Zentralen unterstützen Adressen > 9999.<br>Alternativ kann hier ein Schalter (z.B. „SwitchB7“) oder eine selbst definierte Variable (z.B. „HausA“) eingetragen werden. |

| Bezeichnung | Erklärung |
|---|---|
| Typ | Typ des Eingangs:<br>AnAus (Schalter)<br>Rot (Taster)<br>Grün (Taster) |
| Startwert | Definiert den Startwert des Eingangskanals nachdem die Versorgungsspannung eingeschaltet wurde. Er ist gültig bis die entsprechende DCC Nachricht empfangen wird.<br><br>Ist die automatische Speicherung des Letztzustandes aktiviert kann durch Eingabe von '0' die Zustandsspeicherung für diese Zeile deaktiviert werden. Durch Eingabe von '*' wird die Zustandsspeicherung für diese Zeile erzwungen, sofern die Funktion einen Zustand speichern kann, die Standardeinstellung dies aber deaktiviert (z.B. alle Counter mit Timeout). |
| Beschreibung | Hier sollte ein beliebiger Text zur Dokumentation eingegeben werden damit man sich später wieder zurechtfindet. Mit Alt+Enter kann mit einer neuen Zeile begonnen werden. |
| Verteiler-Nummer | In dieser Spalte kann eine Nummer oder eine Bezeichnung des Verteilers eingetragen werden an dem die LEDs angeschlossen sind. |
| Stecker-Nummer | Hier kann die Nummer des Steckplatzes der Verteilerplatine eingetragen werden welche von der angeschlossenen Baugruppe benutzt wird. |
| Icon | Zeigt zu jedem Makro das passende Bild an. Dadurch kann die Funktion schnell erkannt werden. |
| Name | Zeigt zu jedem Makro einen einfachen Namen in der aktiven Sprache an. |
| Beleuchtung, Sound, oder andere Effekte | Hier werden mit einem Doppelklick oder dem „Dialog“ Knopf die Funktionen zum ansteuern der LEDs oder der anderen Verbraucher eingetragen.<br>**Achtung: Die Einträge sollten nur von Experten manuell verändert werden.** |
| Start LedNr | Diese Spalte enthält die Startnummer der ersten LED dieser Zeile. Die Nummer ergibt sich aus der Zeilenposition und der Anzahl der vorangegangenen LEDs.<br>**Die Zahl kann nicht verändert werden (siehe Hinweis am Ende der Tabelle).** |
| LEDs | Hier wird automatisch die Anzahl der von dieser Zeile angesteuerten LEDs eingetragen.<br>**Achtung: Nicht manuell ändern.** |
| InCnt | Sie enthält die Anzahl der lokal benutzten InCh Kanäle. Sie wird automatisch vom Programm geschrieben.<br>**Achtung: Nicht manuell ändern.** |
| Loc InCh | Sie enthält die Anzahl der lokal benutzten InCh Kanäle. Sie wird automatisch vom Programm geschrieben.<br>**Achtung: Nicht manuell ändern.** |
| LED Kanal | Benutzter LED Kanal.<br>0 = Standard<br>1 = Taster<br>Wird automatisch vom Programm geschrieben.<br>**Achtung: Nicht manuell ändern.** |
| Start Tast LED | LED Nummer im Taster Kanal.<br>Wird automatisch vom Programm geschrieben.<br>**Achtung: Nicht manuell ändern.** |
| Start LED G2 | LED Nummer für den benutzerdefinierten Kanal 2.<br>Wird automatisch vom Programm geschrieben.<br>**Achtung: Nicht manuell ändern.** |

| Bezeichnung | Erklärung |
|---|---|
| Start LED G3 | LED Nummer für den benutzerdefinierten Kanal 3.<br>Wird automatisch vom Programm geschrieben.<br>**Achtung: Nicht manuell ändern.** |

### Probleme mit Filterliste

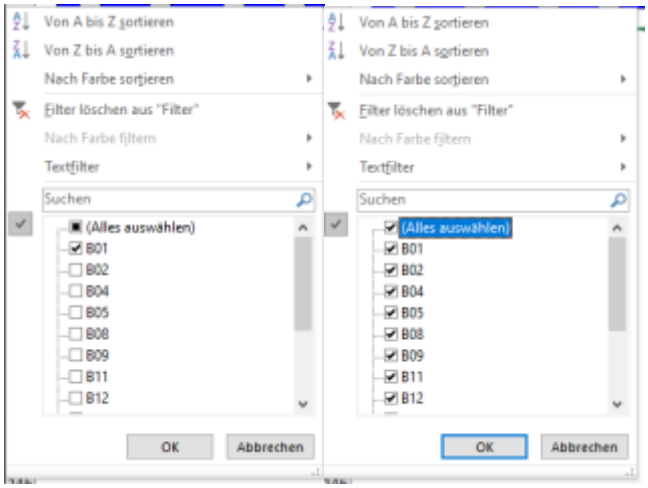
Bei einem Löschen von Zeilen oder dem ganzen Blatt, kann es vorkommen das nicht alles gelöscht wurde oder die Tabelle danach komplett leer ist. Um dieses Problem zu beheben bitte alle Filter löschen und das Kästchen (Leere) aktivieren.




---

# Mitgelieferte Beispiele

Die Beispiele befinden sich ab sofort auf der Extraseite „Examples“.
Beim ersten Aufruf der Tabelle ist nur ein Beispiel mit dem Namen „B01“ vorhanden. Um alle sehen zu können in der Filterliste den Filter auf „Alle auswählen“ setzen.




Danach wird eine längere Liste von Beispielen im Bereich für den Programmablauf angezeigt.

Wenn nicht wurden die Beispiele bereits aus dem Programm gelöscht.
Eine aktuelle und saubere Version des Programm-Generators kann jederzeit von hier wieder heruntergeladen werden.

---

# Das Erstellen von Programmen

## Aufrufen des Dialogs

Der Anfang geht ganz schnell. Dazu oben einfach auf den Button „Dialog“ klicken.

## Auswahl der Ansteuerung

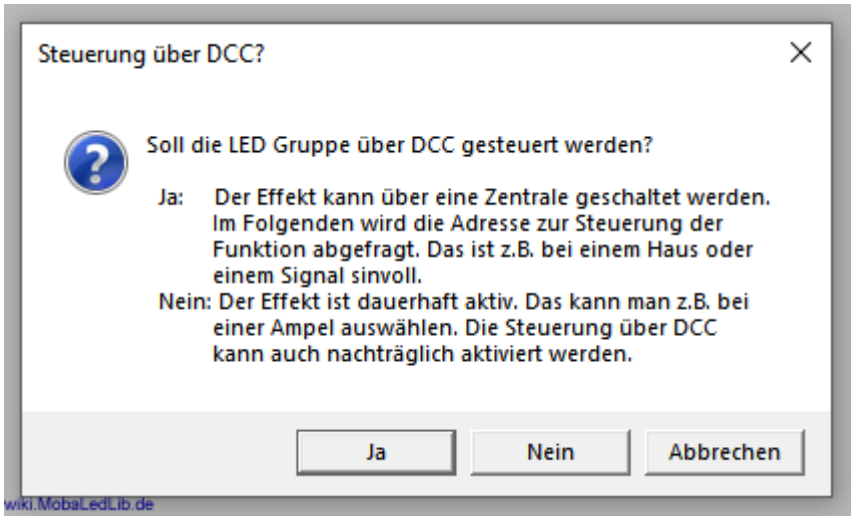
Hier kann ausgewählt werden, ob der Effekt über eine Zentrale (DCC, Selectrix oder Märklin CAN) angesteuert wird oder dauerhaft aktiv ist.
Beispiele für Ansteuerung über eine Zentrale

- Licht- oder Formsignale
- Beleuchtung von Häusern
- Soundeffekte wenn ein Zug im Bahnhof steht oder durchfährt
- Tagsüber Beleuchtung als normale Gaststätte, Abends Discobeleuchtung
- Das brennende Haus - Licht-, Sound und Raucheffekt

Beispiele für dauerhaft aktive Effekte

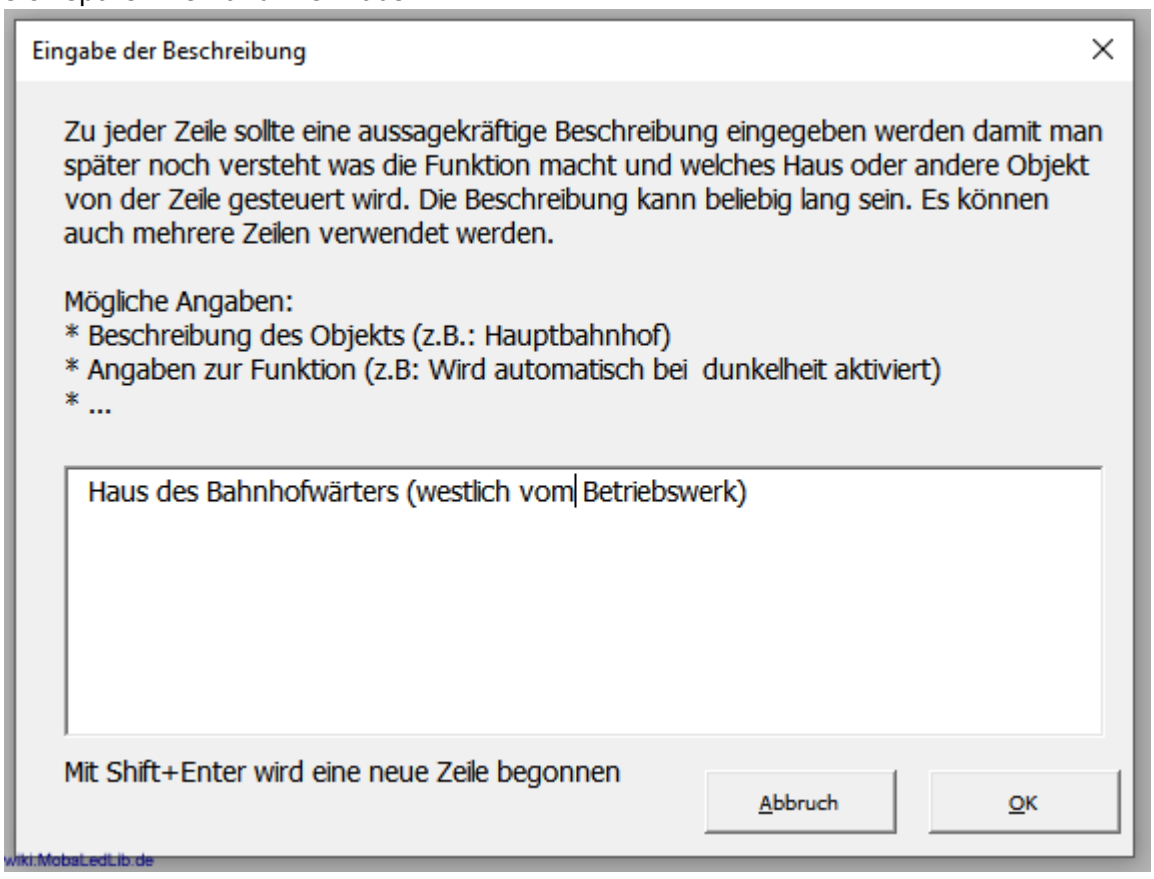
- Ampeln
- Schweißlicht im Bahnbetriebswerk
- Blitzlicht eines Fotografen bei Veranstaltungen
- Blinklichter auf hohen Gebäuden und Windrädern
- Baustellenwarnbaken
- Bewegung mit der MobaLedLib
- Schornsteinfeger bei der Arbeit

Alle Beispiele können natürlich auch dauerhaft aktiv oder von einer Zentrale gesteuert werden.
Möglich ist auch eine Verwendung der „Push Buttons“-Erweiterung für die MobaLedLib, um Effekte per Druckknöpfe welche auf der ganzen Anlage verteilt werden können, auszulösen.




## Vergabe eines schlagkräftigen Namens

In diesem Dialogfenster, sollte dem Effekt ein aussagekräftiger Name vergeben werden. Dies ermöglicht es später diesen leichter wiederzufinden. Wenn alles nur „Beleuchtung Haus“ heißt, kennt sich später niemand mehr aus.




Beispiele für gute Bezeichnungen

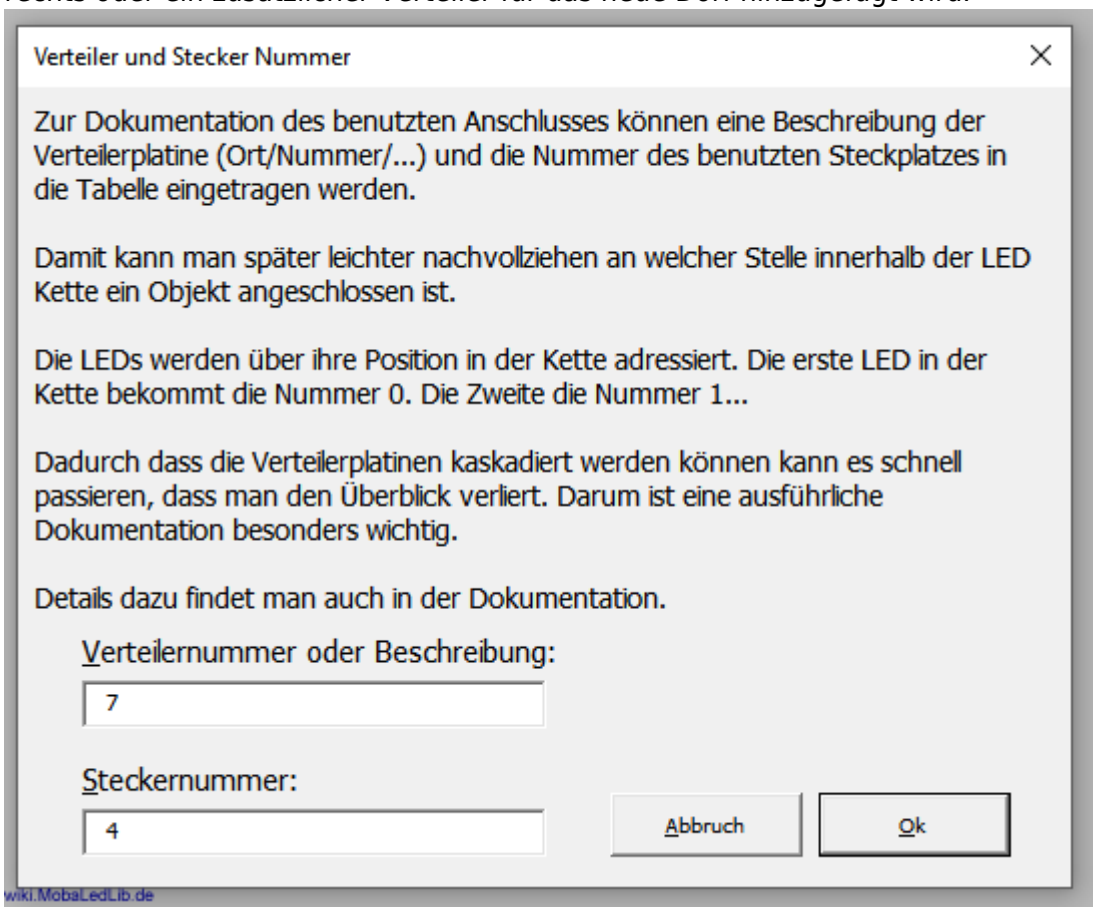
- Kirche am kleinen Marktplatz
- Bahnhofs „Meckershausen“
- Gasthof „zum schnellen Koch“
- Wohnhaus auf dem Berg
- Straßenlaternen Hauptstraße Meckershausen Südseite
- Beleuchtung Bahnsteig Hauptbahnhof

Beispiele für schlechte Namen

- Haus 1, Haus 2, Haus 3
- Straßenlaternen 1, Laternen 2

## Vergabe der Verteiler- und Steckplatznummer

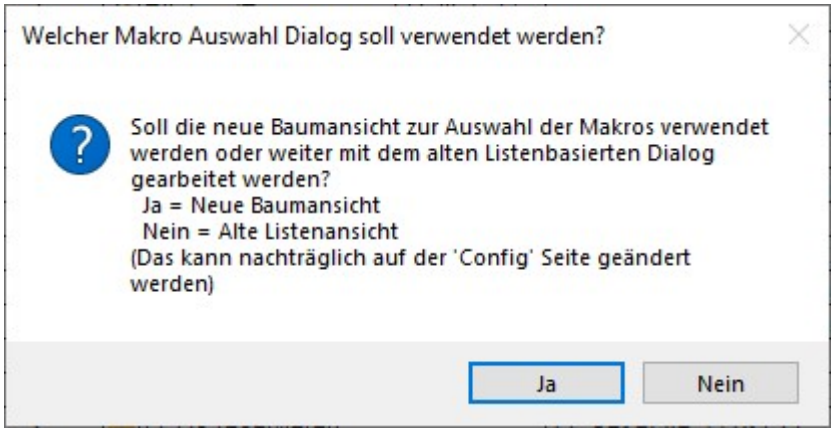
Das nächste Fenster wünscht die Vergabe eine Verteiler- und Steckplatznummer. Dies ermöglicht die Übersicht zu behalten, wenn später z.B. das Haus nicht mehr links auf der Anlage steht sondern rechts oder ein zusätzlicher Verteiler für das neue Dorf hinzugefügt wird.



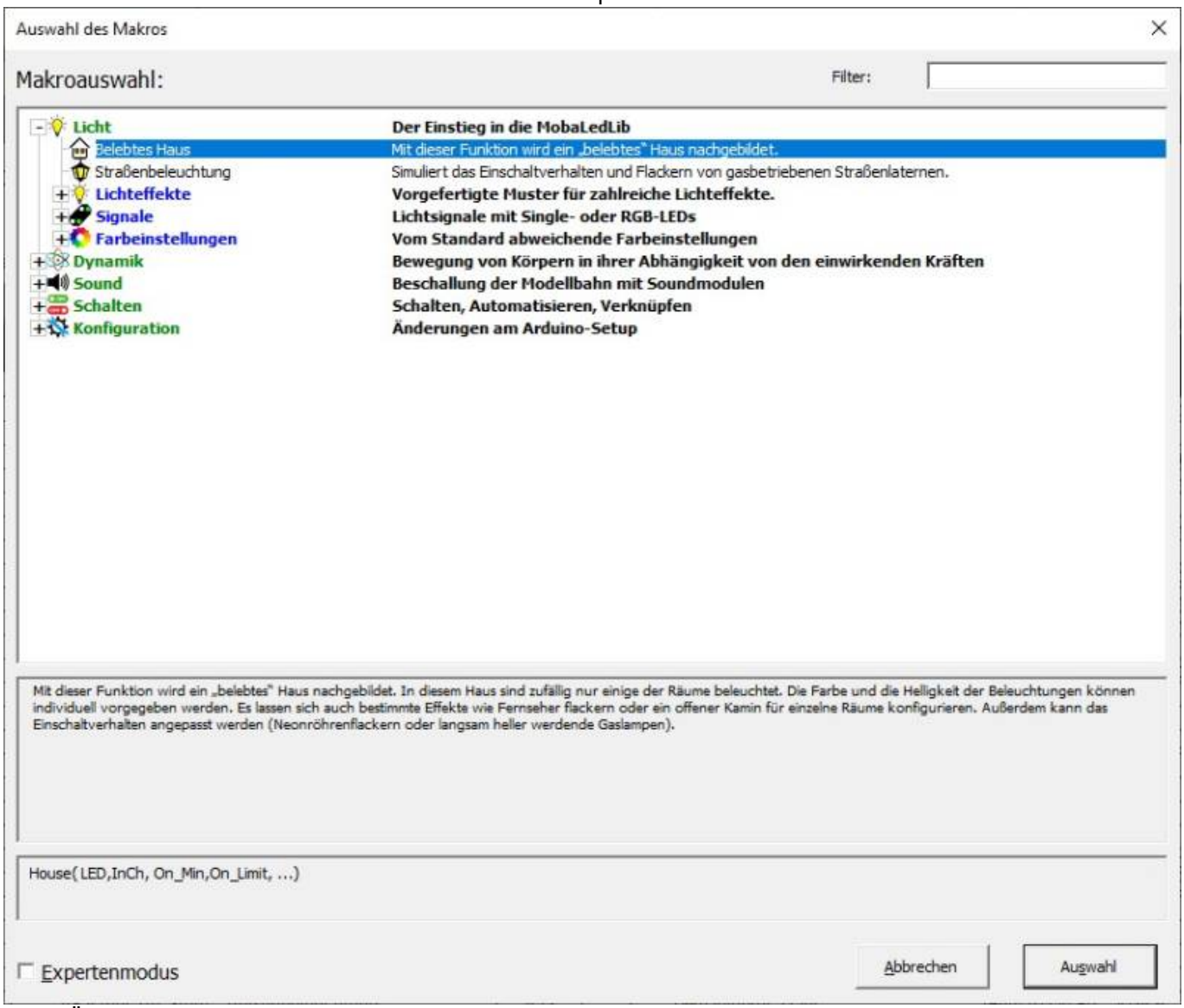

## Auswahl des Effekts

Die nachfolgenden Effekte sind in der Standardeinstellung verfügbar.
Die gesamte Liste der Effekte befindet sich am Ende der Seite als Kurzversion oder hier mit ausführlichen Erklärungen.

Beim Anlegen des ersten Makros wird man gefragt, ob man die neue Baumstruktur verwenden möchte. Diese Frage sollte unbedingt mit „Ja“ beantwortet werden, da das neue Menü sehr viel übersichtlicher gestaltet ist. Diese Einstellung kann selbstverständlich zu jederZeit über die Config-Seite geändert werden.

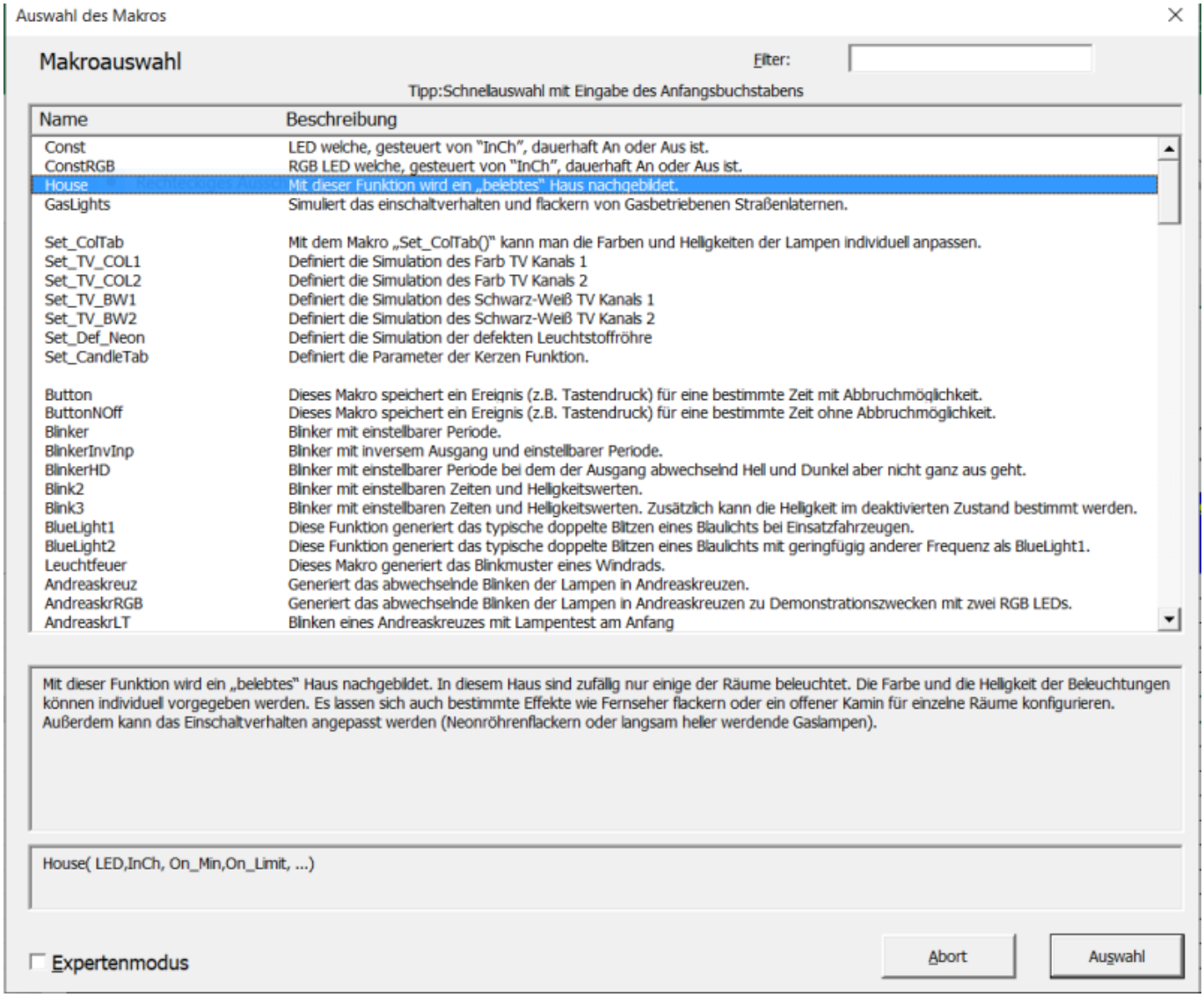



**Neue Listenansicht:** Mit der neuen Baumstruktur präsentiert sich das neue Makro-Auswahlmenü so:




Tipp: Über das Feld „Filter“ oben rechts können die gewünschten Effekte schnell gefunden werden.

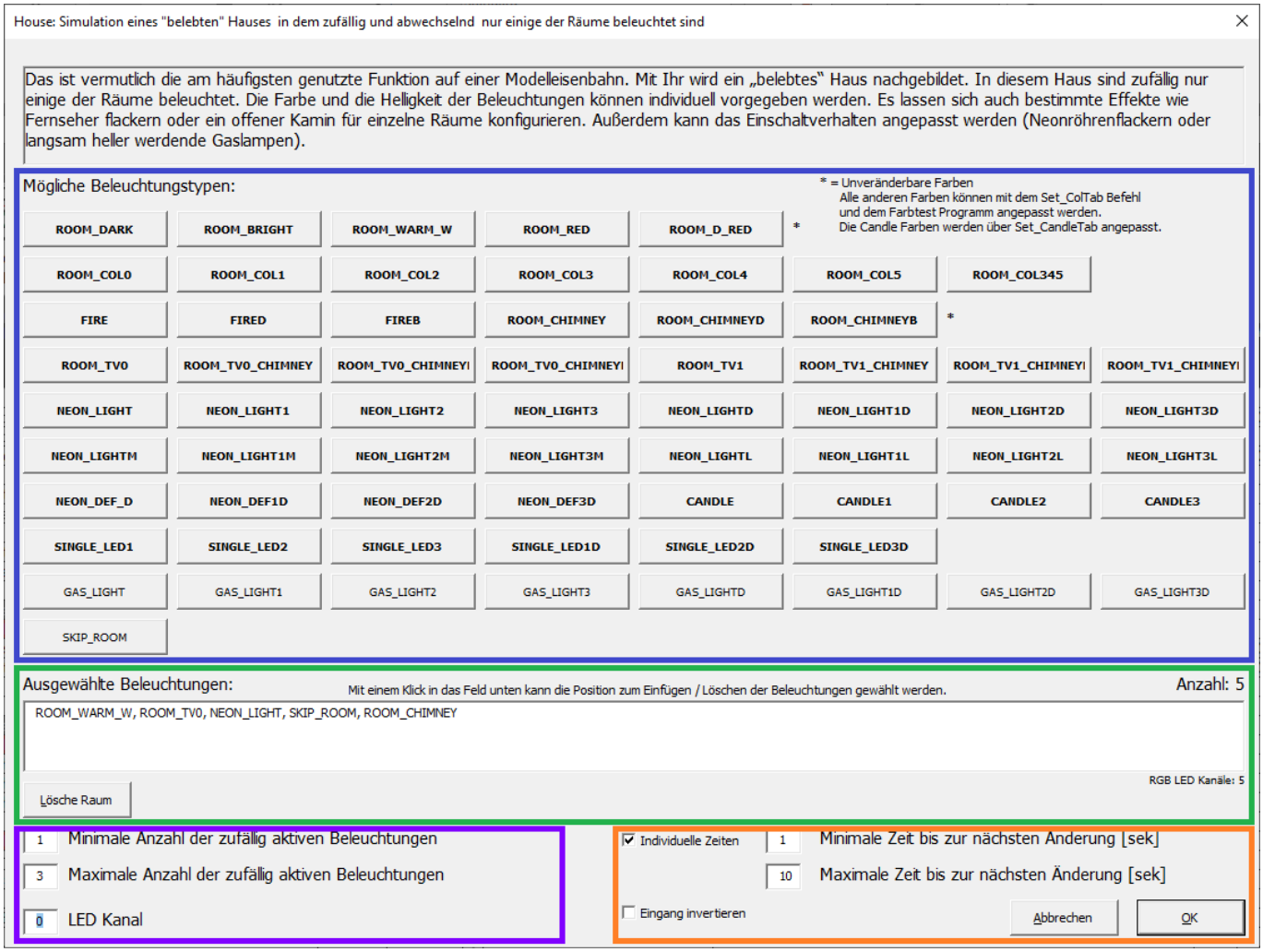
**Zum Vergleich die alte Listenansicht:** Das ist die alte Makro-Auswahl in Standard-Ansicht:




## Konfiguration des Effekts

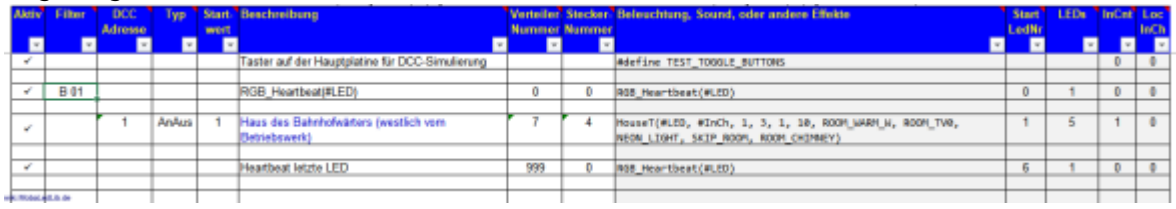
### Auswahl eines Beispiels (belebtes Haus)

Für unseren ersten Test wählen wir mit dem Select-Knopf das „House“ aus und erhalten die folgende Seite zur weiteren Auswahl der Funktionen:

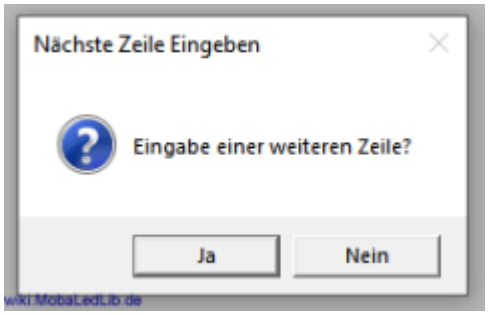



Über das Tastenfeld „mögliche Beleuchtungstypen“ (blauer Rahmen) können unterschiedliche Beleuchtungen für die Räume eines Hauses ausgewählt werden. Wenn in einem Gebäude fünf Räume beleuchtet sind, müssen fünf Beleuchtungen ausgewählt werde. Die ausgewählten Effekte werden in dem grünen Bereich angezeigt. Über die Taste „Lösche Raum“ lassen sich Räume löschen um eine andere Beleuchtung auszuwählen.
In dem violetten Rahmen lässt sich einstellen wie viele Beleuchtungen mindestens und wie viele maximal gleichzeitig leuchten sollen.
Durch Anklicken der Option „Individuelle Zeiten“ (orangener Rahmen) lassen sich die Zeiten für den Beleuchtungswechsel den eigenen Bedürfnissen entsprechend anpassen. Für Testzwecke bietet es sich an, die „Maximale Zeit bis zur nächsten Änderung“ auf 5-10 [Sec] zu setzen.
Das Kästchen „Eingang invertieren“ dreht die Logik des verbunden Schalters /Steuerkanals um. Dies wird z.B. in der Disco verwendet um tagsüber eine normale Gaststätte zu haben und Nachts über die gleichen LEDs die Beleuchtung einer Disco zu simulieren.

Mit einem Klick auf „OK“ werden die Einstellungen übernommen und in dem Programmbereich angezeigt.




Wenn man möchte, können im Anschluss weitere Effekte angelegt werden.




In dem Bild oben ist erkennbar, das vier Zeilen Aktiv geschaltet sind. In der obersten Zeile ist die Verwendung der Testbuttons von der Hauptplatine eingetragen. In der dritten Zeile steht das gerade erstellte Haus mit den getroffenen Einstellungen. Das ist an dem Haken in der Spalte „Aktiv“ zu erkennen. Zeilen können mit einem einfachen Mausklick in die Spalte aktiviert und deaktiviert werden. Nicht-aktive Zeilen werden nicht zum Arduino übertragen. Der grau hinterlegte Bereich ist automatisch befüllt worden und kann bzw. sollte nicht geändert werden. Über die roten Dreiecke in den Feldern können zur weiteren Erklärung Tooltips aufgerufen werden.

Die zweite Zeile mit dem „Heartbeat 1.LED“ sollte in jedem Projekt verwendet werden. Damit wird die erste LED in der Kette genutzt, um zu signalisieren, dass die Übertragung des Programms an den Arduino erfolgreich war und das System „lebt“. Falls die folgenden LEDs dann trotzdem nicht so arbeiten wie erwartet, hat man irgendwo in der Auswahl für die LEDs einen Fehler gemacht oder in der Verdrahtung der LEDs liegt ein Fehler vor.

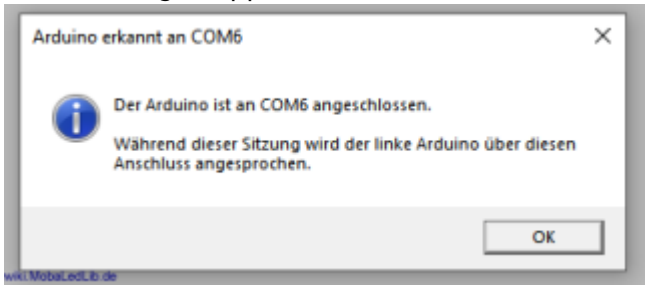
Zusätzlich kann auch auf der Hauptplatine die letzte LED installiert und für die vierte Zeile verwenden werden. Wenn auch diese im Regenbogenfarben blinkt, kann man davon ausgehen, das die Verkabelung stimmt und alles funktioniert.

### Besonderheit beim ersten Hochladen

Beim ersten Hochladen eines Programmes mit DCC-Steuerung wird gefragt, ob das Programm für den DCC-Arduino schon auf diesem installiert wurde. Sollte es schon erledigt sein, kann man das Dialogfenster mit einem Klick auf „Ja“ einfach schließen und der Upload zum LED-Arduino wird nun gestartet. Wenn man es noch noch nicht gemacht hat, kann man die Schaltfläche „Installieren“ auswählen, dann erfolgt die Installation des notwendigen Programmes. Dies ist nur einmalig notwendig oder wenn durch Hardi eine neue Version der Software veröffentlicht wird.

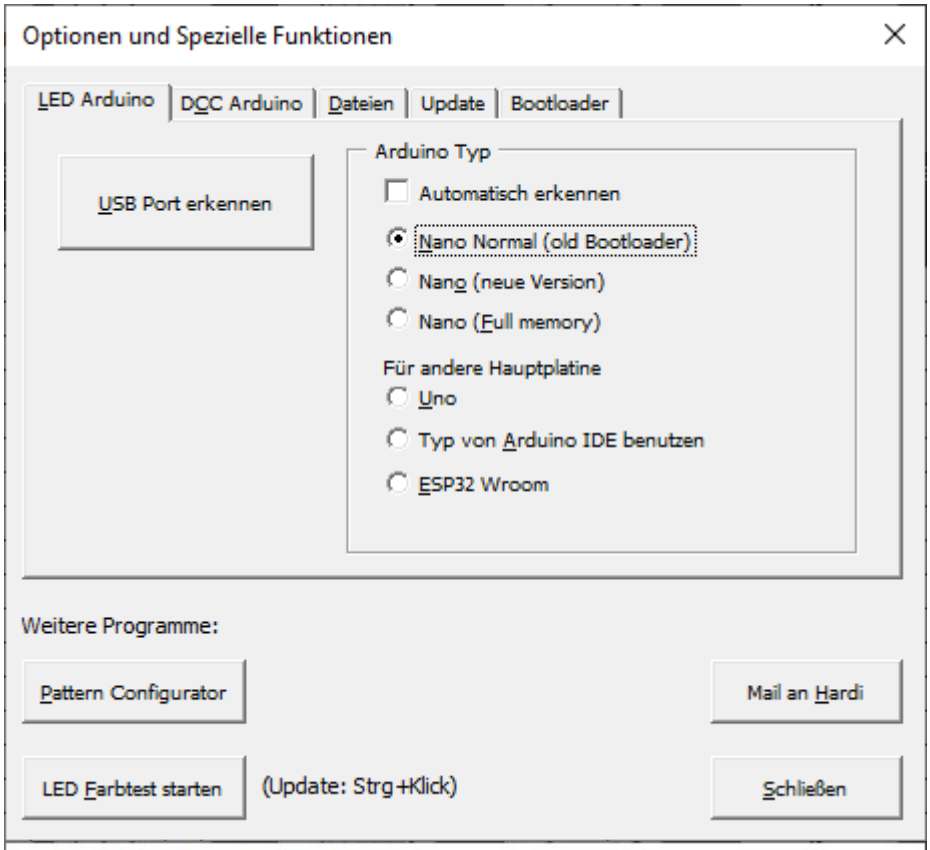
Im nächsten Schritt wird unsere Auswahl zum ARDUINO geschickt.
Beim ersten Sendeversuch erfolgt die Aufforderung, den benutzten COM-Port festzulegen. Einfach den Anweisungen folgen und anschließend den „Z.Arduino schicken“ Knopf drücken.

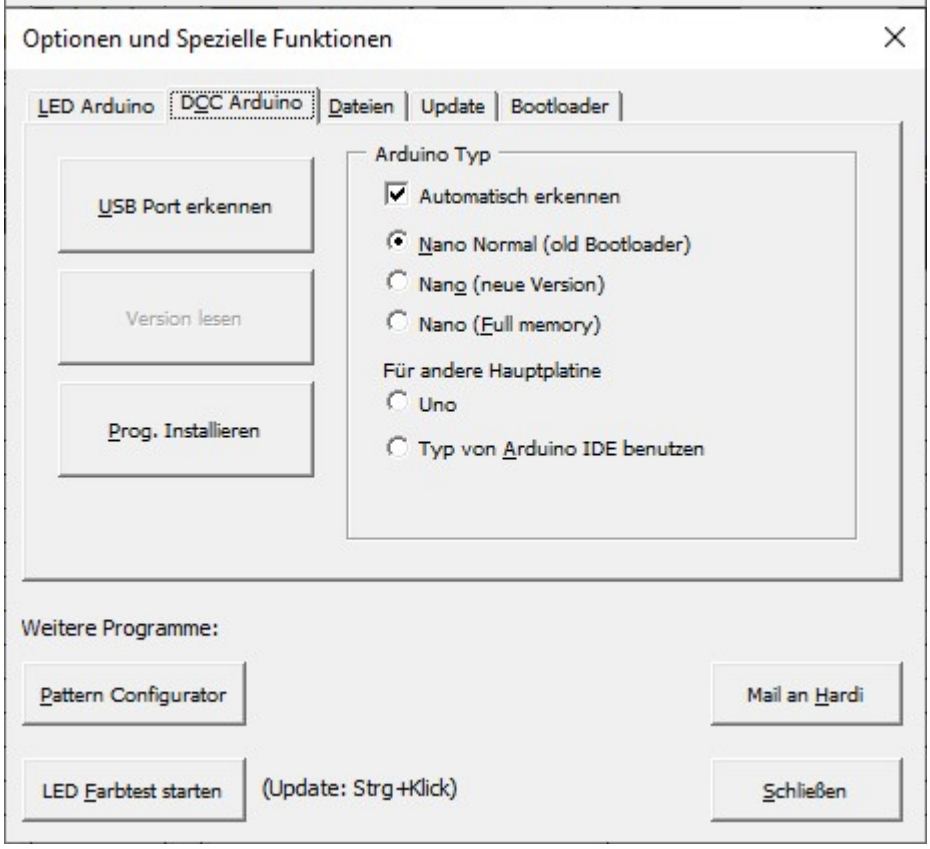
Wenn alles geklappt hat, sieht es so aus.




Bei Problem sieht es so aus.

Der Vorgang zum Erkennen des COM-Ports kann auch über „Optionen“ → „USB Port erkennen“ angestoßen werden.
Für eine fehlerfreie Übertragung zum Arduino muss im Auswahlmenü unbedingt der tatsächlich genutzte Typ eingetragen werden. Für Arduino Nano Clones i.d.R. „Nano Normal (old Bootloader)“ auswählen. Ist der Arduino Typ nicht aufgeführt, sollte der Punkt „Typ von Arduino IDE benutzen“ ausgewählt werden und hier übernommen werden. Die Auswahl des Arduino-Types muss für den LED-Arduino und für den Steuer-Arduino (DCC, Selectrix, LocoNet™,...) separat erfolgen.




# Config

In dem Tabellenblatt „Config“ können die Parameter direkt geändert werden.

| Zeile | Standardwert | Erklärung |
|---|---|---|
| Minimale Zeit bis zur nächsten Änderung | 50 | Definiert wie lange es minimal dauert bis eine Änderung der aktiven Räume eines Hauses auftritt. 0..255 [sek] Wenn nichts eingetragen ist wird 50 Sekunden verwendet.<br>Diesen und den folgenden Parameter kann man zu Testzwecken verkleinern wenn man die House() Beleuchtung im „Zeitraffer“ betrachten will. |
| Maximale Zeit bis zur nächsten Änderung | 150 | Definiert wie lange es maximal dauert bis eine Änderung der aktiven Räume eines Hauses auftritt. 1..255 [sek]<br>Wenn nichts eingetragen ist wird 150 Sekunden verwendet.<br>Wenn die Min und Max Zeit gleich groß sind, dann wird genau nach dieser Zeit die nächste Änderung ausgelöst. Die Maximale Zeit darf nicht kleiner als die Minimale Zeit sein. 0 ist auch nicht erlaubt. |
| Offset für DCC Adresse | | Über diesen Parameter kann ein Offset zur DCC Adresse addiert werden.<br>Ein Offset von 4 ist z.B. bei Roco Zentralen sinnvoll.<br>**Erklärung:**<br>Roco nummeriert die Weichen ab Modul 0 (mit jeweils 4 Weichen), andere DCC-Zentralenhersteller erst ab Modul 1. Diese unterschiedliche Zählweise ist historisch aus einer Schwäche der Spezifikation NMRA S-9.2.1 gewachsen, wo keine der beiden Zählweisen grundsätzlich als „falsch“ bezeichnet werden konnte.<br>Dieser Offset kann auch über das „ROCO Maintenance Tool“ angegeben werden. |
| Aktiviere LED Farbtest Mode | 1 | Wenn der Farbtest Mode aktiviert ist (1), dann können die Farben der LEDs von diesem Programm aus Live verstellt werden. |
| Anzeige der StartLed Nummer | 1 | Dieser Parameter bestimmt die Darstellung der Start Led Nummer. |
| Baumansicht zur Makroauswahl verwenden | | Wenn der Schalter aktiviert ist (1), dann wird die neue Baumansicht zur Auswahl der Makros verwendet. Mit (0) werden die Makros über die alte Listauswahl selektiert. Ist keine Zahl in der Config eingetragen, fragt der Programm Generator beim Anlegen der nächsten Funktion, welche Ansicht verwendet werden soll. |
| Expert Mode aktivieren | | Mit einer 1 wird der Expertenmodus bei der Makro Auswahl aktiviert. |
| Icons anzeigen | 1 | Zeigt zu jedem Makro das passende Bild an. Dadurch kann die Funktion schnell erkannt werden. Diese Funktion wird mit (1) aktiviert. |
| Einfachen Namen | 1 | Zeigt zu jedem Makro einen einfachen Namen in der aktiven Sprache an. |
| Ausführliche Makro Ansicht | 1 | Zeigt die ausführlichen Makros mit allen Parametern an. Bis zur Programm Version 3.0.0 gab es nur diese Spalte. |
| Schnelles Build und Upload verwenden | | Schnellere Methode zum Hochladen des Programms verwenden. Diese Eingabe sollte nur von Experten verändert werden. |
| Zusätzlich benötigte Bibliotheken | | Listen der zusätzlich benötigten Bibliotheken |

| Zeile | Standardwert | Erklärung |
|---|---|---|
| Bibliotheken an anderer Stelle installiert | | Wenn der Benutzer die Bibliotheken nicht im Standardverzeichnis installiert hat, dann wird hier „Ja“ eingetragen. Es erfolgt keine weitere Überprüfung der Bibliotheken innerhalb einer Sitzung. Beim nächsten Programmstart muss die Frage erneut beantwortet werden. |
| Umstellung der Sprache | | Ändert die Anzeigesprache. Das Programm muss nach einer Änderung neu gestartet werden. |
| Verwende SPI Kommunikation | | Aktiviert die SPI Kommunikation zwischen den beiden Arduinos. Dadurch treten keine Fehler mehr beim Flashen des Arduinos auf. Es werden aber zusätzliche Pins benötigt, die dann nicht mehr zum Einlesen von Tastern zur Verfügung stehen |
| Arduino Ausgaben in Excel Fenster | 0 | Die Ausgaben beim Kompilieren des Programms werden in einem Excel Fenster angezeigt. Diese Funktion sollte nur von Experten verwendet werden. |

# Effekte

Hier findet man ausführlichere Beschreibungen der Effekte.
Zur Beschreibung der einzelnen Effekte gelangt man auch durch das Klicken auf den Namen des Effekts.

## Lichteffekte

| @ | Name des Effekts | Funktion | Kurzbeschreibung |
|---|---|---|---|
| | **Licht** | | |
| | Belebtes Haus | House | Mit dieser Funktion wird ein „belebtes“ Haus nachgebildet. |
| | Straßenlaternen | GasLights | Simuliert das Einschaltverhalten bzw. das Flackern von gasbetriebenen Straßenlaternen, von Neonröhren, von Kerzen oder von modernen aber simplen LED-Lampen. |
| | LED einstellbar | Const | LED welche, gesteuert von “InCh”, dauerhaft An oder Aus ist. |
| | RGB-LED einstellbar | ConstRGB | RGB LED welche, gesteuert von “InCh”, dauerhaft An oder Aus ist. |
| | Heartbeat LED | RGB_Heartbeat | RGB LED welche als Funktionsindikator in wechselnden Regenbogenfarben blinkt. |
| | Heartbeat LED einstellbar | RGB_Heartbeat2 | RGB LED welche als Funktionsindikator in wechselnden Regenbogenfarben blinkt. Minimale und maximale Helligkeit kann angegeben werden. |

| @ | Name des Effekts | Funktion | Kurzbeschreibung |
|---|---|---|---|
| | Heartbeat LED einstellbare Farbe | RGB_Heartbeat_Color | RGB LED welche als Funktionsindikator mit einstellbarer Farbe blinkt. Minimale und maximale Helligkeit sowie die Farbe und Blinkgeschwindigkeit können angegeben werden. |
| | Leuchtfeuer | Leuchtfeuer | Dieses Makro generiert das Blinkmuster eines Windrads. |
| | Blitzlicht | Flash | Die „Flash()“ Funktion erzeugt ein zufälliges Blitzen eines Fotografen. |
| | Feuer | Fire | Mit der „Fire()“ Funktion können größere Feuer simuliert werden. |
| | Defekte Neonlampe | Def_Neon_Misha | Simmulation eines defekten Neonlicht von Misha |
| | **Ampel** | | |
| | Ampel | AmpelX | Damit wird das Muster zweier Ampeln für eine Kreuzung erzeugt. |
| | Ampel RGB | RGB_AmpelX | Damit wird das Muster zweier Ampeln für eine Kreuzung mit 6 RGB LEDs erzeugt. |
| | Ampel RGB Soft | RGB_AmpelXFade | Damit wird das Muster zweier Ampeln für eine Kreuzung mit 6 RGB LEDs erzeugt. (Langsames überblenden) |
| | Ampel RGB Österreich | RGB_AmpelX_A | Damit wird das Muster zweier Ampeln für eine Kreuzung in Österreich mit 6 RGB LEDs erzeugt. |
| | **Andreaskreuz** | | |
| | Andreaskreuz | Andreaskreuz | Generiert das abwechselnde Blinken der Lampen in Andreaskreuzen. |
| | Andreaskreuz RGB | AndreaskrRGB | Generiert das abwechselnde Blinken der Lampen in Andreaskreuzen zu Demonstrationszwecken mit zwei RGB LEDs. |
| | Andreaskreuz Lampentest | AndreaskrLT | Blinken eines Andreaskreuzes mit Lampentest am Anfang |
| | Andreaskreuz Lampentest RGB | AndreaskrLT_RGB | Blinken eines Andreaskreuzes mit Lampentest am Anfang zu Demonstrationszwecken mit zwei RGB LEDs. |
| | Andreaskreuz Bü 1 Signal | AndreaskrLT3 | Blinken eines Andreaskreuzes mit Lampentest am Anfang und Bü1 Signal für Zugführer |

| @ | Name des Effekts | Funktion | Kurzbeschreibung |
|---|---|---|---|
| | Andreaskreuz Bü 1 Signal RGB | AndreaskrLT3_RGB | Blinken eines Andreaskreuzes mit Lampentest am Anfang und Bü1 Signal für Zugführer zu Demonstrationszwecken mit drei RGB LEDs. |
| | **Baustellen-Lauflicht** | | |
| | Baustellenlicht 6x | ConstrWarnLightRGB6 | Baustellenlicht mit 6 RGB-LEDs (WS2812) |
| | Baustellenlicht 3-15x | ConstrWarnLight | Baustellenlicht mit 3 bis 15 einzelnen LEDs (WS2811) |
| | **Blaulicht** | | |
| | Blaulicht 1 | BlueLight1 | Diese Funktion generiert das typische doppelte Blitzen eines Blaulichts bei Einsatzfahrzeugen. |
| | Blaulicht 2 | BlueLight2 | Diese Funktion generiert das typische doppelte Blitzen eines Blaulichts mit geringfügig anderer Frequenz als BlueLight1. |
| | **Blinker** | | |
| | Blinker | Blinker | Blinker mit einstellbarer Periode. |
| | Blinker invers | BlinkerInvInp | Blinker mit inversem Ausgang und einstellbarer Periode. |
| | Blinker (Minimum) | BlinkerHD | Blinker mit einstellbarer Periode bei dem der Ausgang abwechselnd Hell und Dunkel aber nicht ganz aus geht. |
| | Blinker (Frequenz und Helligkeit) | Blink2 | Blinker mit einstellbaren Zeiten und Helligkeitswerten. |
| | Blinker komplett einstellbar | Blink3 | Blinker mit einstellbaren Zeiten und Helligkeitswerten. Zusätzlich kann die Helligkeit im deaktivierten Zustand bestimmt werden. |
| | **Schweißlicht** | | |
| | Schweißlicht dauerhaft | WeldingCont | Mit der „WeldingCont()“ Funktion kann ein dauerhaft aktives Schweißlicht simuliert werden. |
| | Schweißlicht einmalig | Welding | Mit der „Welding()“ Funktion kann ein Schweißlicht simuliert werden. |
| | Schweißlicht zufällig | RandWelding | Mit der „RandWelding()“ Funktion kann ein zufällig aktives Schweißlicht simuliert werden. |
| | **Signale** | | |
| | Einfahrsignal | EntrySignal3 | Einfahrsignal mit 3 einzelnen LEDs welche über ein WS2811 Modul angesteuert werden. (HP0, HP1, HP2). Es wird über drei Taster gesteuert. |

| @ | Name des Effekts | Funktion | Kurzbeschreibung |
|---|---|---|---|
| | Einfahrsignal RGB | EntrySignal3_RGB | Einfahrsignal mit 3 RGB LEDs zu Testzwecken (HP0, HP1, HP2). Es wird über drei Taster gesteuert. |
| | Einfahrsignal RGB (max) | EntrySignal3_RGB_B | Einfahrsignal mit 3 RGB LEDs zu Testwecken (HP0, HP1, HP2). Es wird über drei Taster gesteuert. |
| | Ausfahrsignal | DepSignal4 | Ausfahrsignal mit 6 einzelnen LEDs welche über zwei WS2811 Module angesteuert werden. (HP0, HP1, HP2, HP0+SH1). Es wird über vier Taster gesteuert. |
| | Ausfahrsignal RGB | DepSignal4_RGB | Ausfahrsignal mit 6 RGB LEDs zu Testzwecken (HP0, HP1, HP2, HP0+SH1). Es wird über vier Taster gesteuert. |
| | **KS-Signalsystem** | | |
| | KS-Vorsignal Zs3V | KS_Vorsignal_Zs3V | Einzel LED Ks-Vorsignalwiederholer: Ks 2 / Ks 1 / Ks 1+ Zs 3V / Kennlicht |
| | KS-Vorsignal Zs3V RGB | KS_Vorsignal_Zs3V_RGB | RGB Ks-Vorsignalwiederholer: Ks 2 / Ks 1 / Ks 1+ Zs 3V / Kennlicht |
| | KS-Hauptsignal Zs3 Zs1 | KS_Hauptsignal_Zs3_Zs1 | Einzel LED Ks-Hauptsignal: Hp 0 / Ks 1 / K2 1 + Zs 3 / Hp 0 + Zs 1 |
| | KS-Hauptsignal Zs3 Zs1 RGB | KS_Hauptsignal_Zs3_Zs1_RGB | RGB Ks-Hauptsignal: Hp 0 / Ks 1 / K2 1 + Zs 3 / Hp 0 + Zs 1 |
| | KS-Hauptsignal Zs3 Zs6 Zs1 | KS_Hauptsignal_Zs3_Zs6_Zs1 | Einzel LED Ks-Hauptsignal: HP 0 / Ks 1 / Ks 1 + Zs 3 + Zs 6 / HP 0 + Zs 1 |
| | KS-Hauptsignal Zs3 Zs6 Zs1 RGB | KS_Hauptsignal_Zs3_Zs6_Zs1_RGB | RGB Ks-Hauptsignal: HP 0 / Ks 1 / Ks 1 + Zs 3 + Zs 6 / HP 0 + Zs 1 |
| | **Signale WS2812 by Matthias** | | |
| | Hauptsignal links RGB | HS_5l_RGB | RGB Hauptsignal mit 5 WS2812 LEDs (links) |
| | Hauptsignal rechts RGB | HS_5r_RGB | RGB Hauptsignal mit 5 WS2812 LEDs (rechts) |
| | Haupt- und Vorsignal RGB | HS_5_Plus_RGB | RGB Hauptsignal und Vorsignal mit 9 WS2812 LEDs |
| | Gleissperrsignal RGB | Gleissperrsignal_RGB | RGB Gleissperrsignal mit 4 WS2812 LEDs |
| | **Signale (Trix)** | | |
| | Hauptsignal einfach RGB | HS_Einfach_RGB | RGB Hauptsignal Einfach mit 4 WS2812 LEDs |
| | Vorsignal RGB | Vorsignal_RGB | RGB Vorsignal mit 4 WS2812 LEDs |
| | Hauptsignal doppelt RGB | HP_2_2_RGB | RGB Hauptsignal mit 4 WS2812 LEDs (Doppelte Anzeige) |

| @ | Name des Effekts | Funktion | Kurzbeschreibung |
|---|---|---|---|
| | Einfahrsignal (Trix) | EntrySignal3Bin | Einfahrsignal mit 3 einzelnen LEDs welche über ein WS2811 Modul angesteuert werden. (HP0, HP1, HP2). Es wird über zwei binäre Eingänge gesteuert. Damit eignet es sich besonders für Selectrix. |
| | Einfahrsignal RGB (Trix) | EntrySignal3Bin_RGB | Einfahrsignal mit 3 RGB LEDs zu Testzwecken (HP0, HP1, HP2). Es wird über zwei binäre Eingänge gesteuert. Damit eignet es sich besonders für Selectrix. |
| | Ausfahrsignal (Trix) | DepSignal4Bin | Ausfahrsignal mit 6 einzelnen LEDs welche über zwei WS2811 Module angesteuert werden. (HP0, HP1, HP2, HP0+SH1). Es wird über zwei binäre Eingänge gesteuert. Damit eignet es sich besonders für Selectrix. |
| | Ausfahrsignal RGB (Trix) | DepSignal4Bin_RGB | Ausfahrsignal mit 6 RGB LEDs zu Testzwecken (HP0, HP1, HP2, HP0+SH1). Es wird über zwei binäre Eingänge gesteuert. Damit eignet es sich besonders für Selectrix. |
| | **KS-Signalsystem (Trix)** | | |
| | KS-Vorsignal Zs3V (Trix) | KS_Vorsignal_Zs3VBin | Einzel LED Ks-Vorsignalwiederholer (Binär): Ks 2 / Ks 1 / Ks 1+ Zs 3V / Kennlicht |
| | KS-Vorsignal Zs3V RGB (Trix) | KS_Vorsignal_Zs3VBin_RGB | RGB Ks-Vorsignalwiederholer (Binär): Ks 2 / Ks 1 / Ks 1+ Zs 3V / Kennlicht |
| | KS-Hauptsignal Zs3 Zs1 (Trix) | KS_Hauptsignal_Zs3_Zs1Bin | Einzel LED Ks-Hauptsignal (Binär): Hp 0 / Ks 1 / K2 1 + Zs 3 / Hp 0 + Zs 1 |
| | KS-Hauptsignal Zs3 Zs1 RGB (Trix) | KS_Hauptsignal_Zs3_Zs1Bin_RGB | RGB Ks-Hauptsignal (Binär): Hp 0 / Ks 1 / K2 1 + Zs 3 / Hp 0 + Zs 1 |
| | KS-Hauptsignal Zs3 Zs6 Zs1 (Trix) | KS_Hauptsignal_Zs3_Zs6_Zs1Bin | Einzel LED Ks-Hauptsignal (Binär): HP 0 / Ks 1 / Ks 1 + Zs 3 + Zs 6 / HP 0 + Zs 1 |
| | KS-Hauptsignal Zs3 Zs6 Zs1 RGB (Trix) | KS_Hauptsignal_Zs3_Zs6_Zs1Bin_RGB | RGB Ks-Hauptsignal (Binär): HP 0 / Ks 1 / Ks 1 + Zs 3 + Zs 6 / HP 0 + Zs 1 |
| | **Signale WS2812 by Matthias (Trix)** | | |
| | Hauptsignal links RGB (Trix) | HS_5lBin_RGB | RGB Hauptsignal (Binär) mit 5 WS2812 LEDs (links) |
| | Hauptsignal rechts RGB (Trix) | HS_5rBin_RGB | RGB Hauptsignal (Binär) mit 5 WS2812 LEDs (rechts) |

| @ | Name des Effekts | Funktion | Kurzbeschreibung |
|---|---|---|---|
| | Haupt- und Vorsignal RGB (Trix) | HS_5_PlusBin_RGB | RGB Hauptsignal (Binär) und Vorsignal mit 9 WS2812 LEDs |
| | Gleissperrsignal RGB (Trix) | GleissperrsignalBin_RGB | RGB Gleissperrsignal (Binär) mit 4 WS2812 LEDs |
| | Hauptsignal einfach RGB (Trix) | HS_EinfachBin_RGB | RGB Hauptsignal (Binär) Einfach mit 4 WS2812 LEDs |
| | Vorsignal RGB (Trix) | VorsignalBin_RGB | RGB Vorsignal mit 4 WS2812 LEDs |
| | Hauptsignal doppelt RGB (Trix) | HP_2_2Bin_RGB | RGB Hauptsignal mit 4 WS2812 LEDs (Doppelte Anzeige) |
| | **Farbeinstellungen** | | |
| | Farbe und Helligkeit einstellen | Set_ColTab | Mit dem Makro „Set_ColTab()“ kann man die Farben und Helligkeiten der Lampen individuell anpassen. |
| | Farb-TV Kanal 1 einstellen | Set_TV_COL1 | Definiert die Simulation des Farb TV Kanals 1 |
| | Farb-TV Kanal 2 einstellen | Set_TV_COL2 | Definiert die Simulation des Farb TV Kanals 2 |
| | S/W-TV Kanal 1 einstellen | Set_TV_BW1 | Definiert die Simulation des Schwarz-Weiß TV Kanals 1 |
| | S/W-TV Kanal 2 einstellen | Set_TV_BW2 | Definiert die Simulation des Schwarz-Weiß TV Kanals 2 |
| | Neonöhre Defekt einstellen | Set_Def_Neon | Definiert die Simulation der defekten Leuchtstoffröhre |
| | Kerzen einstellen | Set_CandleTab | Definiert die Parameter der Kerzen Funktion. |

## Dynamik

| @ | Name des Effekts | Funktion | Kurzbeschreibung |
|---|---|---|---|
| | **Servo** | | |
| | Servo mit 2 Positonen | servo2 | Servo Steuerung mit zwei Positionen welche über Taster angesteuert werden |
| | Servo mit 3 Positonen | servo3 | Servo Steuerung mit drei Positionen welche über Taster angesteuert werden |
| | Servo mit 4 Positonen | servo4 | Servo Steuerung mit vier Positionen welche über Taster angesteuert werden |
| | Servo mit 5 Positonen | servo5 | Servo Steuerung mit fünf Positionen welche über Taster angesteuert werden |
| | **Herzstückpolarisierung bistabil v1.1** | | |
| | Relaiskontakt A/B bis E/F | Herz_BiRelais | Polarisierung über definierten Anschluss (v1.1) mit verzögert angesteuertem bistabilen Relais |
| | Relaiskontakt A/B bis E/F, invers | Herz_BiRelais_I | Polarisierung über definierten Anschluss (v1.1) mit verzögert angesteuertem bistabilen Relais, invers |
| | **Herzstückpolarisierung bistabil v1.0** | | |

| @ | Name des Effekts | Funktion | Kurzbeschreibung |
|---|---|---|---|
| | Relaiskontakte A und B | Herz_BiRelais_V1_AB | Polarisierung über Anschluss A und B (v1.0) mit verzögert angesteuertem bistabilen Relais |
| | Relaiskontakte A und B, invers | Herz_BiRelais_I_V1_AB | Polarisierung über Anschluss A und B (v1.0) mit verzögert angesteuertem bistabilen Relais, invers |
| | Relaiskontakte C und D | Herz_BiRelais_V1_CD | Polarisierung über Anschluss C und D (v1.0) mit verzögert angesteuertem bistabilen Relais |
| | Relaiskontakte C und D, invers | Herz_BiRelais_I_V1_CD | Polarisierung über Anschluss C und D (v1.0) mit verzögert angesteuertem bistabilen Relais, invers |
| | Relaiskontakte E und F | Herz_BiRelais_V1_EF | Polarisierung über Anschluss E und F (v1.0) mit verzögert angesteuertem bistabilen Relais |
| | Relaiskontakte E und F, invers | Herz_BiRelais_I_V1_EF | Polarisierung über Anschluss E und F (v1.0) mit verzögert angesteuertem bistabilen Relais, invers |
| | **Herzstückpolarisierung monostabil** | | |
| | Relaiskontakt A/B bis E/F | Herz_MoRelais | Polarisierung über definierten Anschluss mit einem verzögert angesteuerten monostabilen Relais |
| | Relaiskontakt A/B bis E/F, invers | Herz_MoRelais_I | Polarisierung über definierten Anschluss mit einem verzögert angesteuerten monostabilen Relais, invers |
| | Relaiskontakt A bis F | Herz_2MoRelais | Polarisierung über definierten Anschluss mit zwei verzögert angesteuerten monostabilen Relais |
| | Relaiskontakt A bis F, invers | Herz_2MoRelais_I | Polarisierung über definierten Anschluss mit zwei verzögert angesteuerten monostabilen Relais, invers |

## Sound Funktionen für das MP3-TF-16P Sound Modul

| Name des Soundeffektes | Kurzbeschreibung |
|---|---|
| Sound_Prev | Wiedergabe der vorherigen Sound Datei |
| Sound_Next | Wiedergabe der nächsten Sound Datei |
| Sound_PausePlay | Hält die Sound Wiedergabe an |
| Sound_Loop | Sound Wiedergabe in einer Schleife |
| Sound_USDSPI | Umschalten der Quelle |
| Sound_PlayMode | Abspielmode Umschalten |
| Sound_DecVol | Lautstärke reduzieren |
| Sound_IncVol | Lautstärke erhöhen |
| Sound_Seq1 | Sounddatei 1 abspielen |
| Sound_Seq2 | Sounddatei 2 abspielen |
| Sound_Seq3 | Sounddatei 3 abspielen |
| Sound_Seq4 | Sounddatei 4 abspielen |
| Sound_Seq5 | Sounddatei 5 abspielen |

| Name des Soundeffektes | Kurzbeschreibung |
|---|---|
| Sound_Seq6 | Sounddatei 6 abspielen |
| Sound_Seq7 | Sounddatei 7 abspielen |
| Sound_Seq8 | Sounddatei 8 abspielen |
| Sound_Seq9 | Sounddatei 9 abspielen |
| Sound_Seq10 | Sounddatei 10 abspielen |
| Sound_Seq11 | Sounddatei 11 abspielen |
| Sound_Seq12 | Sounddatei 12 abspielen |
| Sound_Seq13 | Sounddatei 13 abspielen |
| Sound_Seq14 | Sounddatei 14 abspielen |
| Sound_PlayRandom | Zufällige Sounddatei abspielen |
| Sound_Next_of_N_Reset | Nächste Sounddatei von N abspielen mit Reset Eingang |
| Sound_Next_of_N | Nächste Sounddatei von N abspielen |

## Sound Funktionen für das JQ6500 Sound Modul

| Name des Soundeffektes | Kurzbeschreibung |
|---|---|
| Sound_JQ6500_Prev | Wiedergabe der vorherigen Sound Datei |
| Sound_JQ6500_Next | Wiedergabe der nächsten Sound Datei |
| Sound_JQ6500_DecVol | Lautstärke reduzieren |
| Sound_JQ6500_IncVol | Lautstärke erhöhen |
| Sound_JQ6500_Seq1 | Sounddatei 1 abspielen |
| Sound_JQ6500_Seq2 | Sounddatei 2 abspielen |
| Sound_JQ6500_Seq3 | Sounddatei 3 abspielen |
| Sound_JQ6500_Seq4 | Sounddatei 4 abspielen |
| Sound_JQ6500_Seq5 | Sounddatei 5 abspielen |
| Sound_JQ6500_PlayRandom | Zufällige Sounddatei abspielen |
| Sound_JQ6500_Next_of_N_Reset | Nächste Sounddatei von N abspielen mit Reset Eingang |
| Sound_JQ6500_Next_of_N | Nächste Sounddatei von N abspielen |

## Sound Funktionen für das JQ6500 Sound Modul mit anderem WS2811

| Name des Soundeffektes | Kurzbeschreibung |
|---|---|
| effekte_mll | *** Sound Funktionen für das JQ6500 Sound Modul mit anderem WS2811** * |
| Sound_JQ6500_BG_Prev | Wiedergabe der vorherigen Sound Datei |
| Sound_JQ6500_BG_Next | Wiedergabe der nächsten Sound Datei |
| Sound_JQ6500_BG_DecVol | Lautstärke reduzieren |
| Sound_JQ6500_BG_IncVol | Lautstärke erhöhen |
| Sound_JQ6500_BG_Seq1 | Sounddatei 1 abspielen |
| Sound_JQ6500_BG_Seq2 | Sounddatei 2 abspielen |
| Sound_JQ6500_BG_Seq3 | Sounddatei 3 abspielen |
| Sound_JQ6500_BG_Seq4 | Sounddatei 4 abspielen |
| Sound_JQ6500_BG_Seq5 | Sounddatei 5 abspielen |
| Sound_JQ6500_BG_PlayRandom | Zufällige Sounddatei abspielen |
| Sound_JQ6500_BG_Next_of_N_Reset | Nächste Sounddatei von N abspielen mit Reset Eingang |
| Sound_JQ6500_BG_Next_of_N | Nächste Sounddatei von N abspielen mit Reset Eingang |

## Schalten

Die Verwendung des Expertenmodus wir nur erfahrenen Benutzern empfohlen!

| @ | Expert | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|---|
| | | **Abhängigkeiten** | | |
| | | Logische Verknüpfung | Logic | Logische Verknüpfungen mehrerer Eingangsvariablen zu einer Ausgangsvariable |
| | | LED-Werte kopieren | CopyLED | Mit dem „CopyLED()“ Befehl wird die Helligkeit der drei Farben der „SrcLED“ in die „LED“ kopiert. |
| | | LED-Werte als Variable | LED_to_Var | Setzt eine Variable abhängig von einer LED Helligkeit. |
| | | Variable für 256 Zustände erstellen | New_Local_Var | Erzeugt eine Variable in der 256 verschiedene Zustände gespeichert werden können. |
| | | Verknüpfung zur MLL-Bibliothek | Use_GlobalVar | Mit der Funktion „Use_GlobalVar()“ können die eigenen Programmteile mit den bibliotheksinternen Funktionen Daten austauschen. |
| | | Temporäre 8bit Variable erstellen | InCh_to_TmpVar | Mit diesem Befehl wird eine temporäre 8 Bit Variable mit den Werten aus mehreren Logischen Variablen gefüllt. |
| | | Temporäre 8bit Variable erstellen, binär | Bin_InCh_to_TmpVar | Mit diesem Befehl wird eine temporäre 8 Bit Variable mit den binären Werten aus mehreren logischen Variablen gefüllt. |
| | | Eingangsvariablen definieren | Define Input | Definiert Variable aus einen DCC, Selectrix oder CAN Eingang (Bsp.: INCH_DCC_1_ONOFF) |
| | | **Automatisierung** | | |
| | | Zeitplan | Schedule | Mit dem „Schedule“ Makro kann ein Zeitplan für das Ein- und Ausschalten mehrerer Lichter erstellt werden. |
| | **Expert** | Zählwerk | Counter | Die „Counter()“ Funktion kann für die verschiedensten (Zähler) Aufgaben verwendet werden. |
| | **Expert** | Timer mit Abbruch | Button | Dieses Makro speichert ein Ereignis (z.B. Tastendruck) für eine bestimmte Zeit mit Abbruchmöglichkeit. |
| | | Timer ohne Abbruch | ButtonNOff | Dieses Makro speichert ein Ereignis (z.B. Tastendruck) für eine bestimmte Zeit ohne Abbruchmöglichkeit. |

| @ | Expert | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|---|
| | **Expert** | Treppenhausschalter | ButtonFunc | Dieses Makro entspricht einem Treppenhaus Lichtschalter. |
| | | Zufallsschaltung | RandMux | Die „RandMux()“ Funktion aktiviert zufällig einen der Ausgänge. |
| | | Zufallsschaltung 1 Ausgang | Random | Die Funktion „Random()“ aktiviert einen Ausgang nach einer zufälligen Zeit. |
| | **Expert** | Zufallsschaltung sequenziell | RandCntMux | Die „RandCntMux()“ Funktion aktiviert nach einer zufälligen Zeit den nächsten Ausgang. |
| | **Expert** | Multiplexer | Multiplexer | Kombiniert bis zu 8 verschiedenen Muster zu einer Gruppe und aktiviert diese zufällig. |
| | | **Charlieplexing** | | |
| | | Charlieplexing Taster | Charlie_Buttons | Steuert die Zustände eines Charlieplexing Moduls per Taster an. |
| | | Charlieplexing binär | Charlie_Binary | Steuert die Zustände eines Charlieplexing Moduls binär an. |
| | | **Manipulation** | | |
| | **Expert** | LED Nummer manipulieren | Next LEDs | Manipulation der LED Nummer |
| | | Reserve LEDs | LEDs reservieren | Mit diesem Eintrag können LEDs reserviert werden. |
| HSV | **Expert** | Speicher für HSV reservieren | New_HSV_Group | Speicher reservieren für eine neuen HSV Gruppe. |
| | | **Flip-Flop** | | |
| | **Expert** | RS FlipFlop | RS_Flip-Flop | Ein Flip-Flop kann zwei Zustande annehmen (0 oder 1) welche über zwei Eingänge gesteuert werden. |
| | **Expert** | RS Flip-Flop, Auszeit | RS_FlipFlopTimeout | Flip-Flop mit zwei Zuständen (0 oder 1) welche über zwei Eingänge gesteuert werden incl. Timeout. |
| | **Expert** | Toggle Flip-Flop | T_FlipFlopReset | Der Ausgang eines „Toggle Flip-Flops” wird bei jeder positiven Flanke an Eingang umgeschaltet. |
| | **Expert** | Toggle Flip-Flop, Auszeit | T_FlipFlopResetTimeout | „Toggle Flip-Flops” mit zusätzlichem Parameter „Timeout“. |
| | **Expert** | RS Flip-Flop invers | RS_FlipFlopInv | Dieses Flip-Flop ist zu Beginn aktiv. |
| | **Expert** | RS Flip-Flop invers, Auszeit | RS_FlipFlopInvTimeout | Dieses Flip-Flop ist zu Beginn aktiv und wird per Timeout zurückgesetzt. |
| | **Expert** | Toggle Flip-Flop invers | T_FlipFlopInvReset | Flip-Flop mit Umschalt- und Reseteingang welches zu Begin aktiv ist. |
| | **Expert** | Toggle Flip-Flop invers, Auszeit | T_FlipFlopInvResetTimeout | Flip-Flop mit Umschalt- und Reseteingang und Timeout welches zu Begin aktiv ist. |

| @ | Expert | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|---|
| | **Expert** | RS Flip-Flop bipolar | RS_FlipFlop2 | RS FlipFlop mit zwei Ausgängen, die Invers zueinander geschaltet sind. |
| | **Expert** | RS Flip-Flop bipolar, Auszeit | RS_FlipFlop2Timeout | RS FlipFlop mit zwei Ausgängen, die Invers zueinander geschaltet sind und Timeout. |
| | **Expert** | Toggle Flip-Flop bipolar | T_FlipFlop2Reset | RS FlipFlop mit zwei Ausgängen, die Invers zueinander geschaltet sind und zusätzlichen Reset Eingang. |
| | **Expert** | Toggle Flip-Flop bipolar, Auszeit | T_FlipFlop2ResetTimeout | RS FlipFlop mit zwei Ausgängen, die Invers zueinander geschaltet sind und zusätzlichem Reset Eingang und Timeout. |
| | | **Mono-Flop** | | |
| | **Expert** | Mono-Flop | MonoFlop | Ein Mono Flop ist eine Funktion welche den Ausgang für eine bestimmte Zeit aktiviert. |
| | **Expert** | Mono-Flop, Reset | MonoFlopReset | Mono Flop mit zusätzlichen Reset Eingang |
| | **Expert** | Mono-Flop, Reset nach Zeit | MonoFlopLongReset | Ist ein Mono Flop, der zurückgesetzt werden kann, wenn der Eingang länger als 1.5 Sekunden aktiv ist. |
| | **Expert** | Mono-Flop invers | MonoFlopInv | Dieser Mono Flop besitzt einen inversen Ausgang. |
| | **Expert** | Mono-Flop invers, Reset nach Zeit | MonoFlopInvLongReset | Dieser Mono Flop besitzt einen inversen Ausgang und kann mit langen Tastendruck gelöscht werden. |
| | **Expert** | Mono-Flop bipolar | MonoFlop2 | MonoFlop mit zwei Ausgänge, die Invers zueinander geschaltet sind. |
| | **Expert** | Mono-Flop bipolar, Reset | MonoFlop2LongReset | MonoFlop mit Reset bei langen Impuls und zwei Ausgänge, die Invers zueinander geschaltet sind. |
| | | **Taster beleuchtet** | | |
| | **Expert** | Taster beleuchtet, 1 Funktion | PushButton_w_LED_BL_0_1 | „Knopf Druck Aktion“ mit einen Zustand und einzelner LED im Taster |
| | **Expert** | Taster beleuchtet, 2 Funktionen | PushButton_w_LED_BL_0_2 | „Knopf Druck Aktion“ mit zwei Zuständen und einzelner LED im Taster |
| | **Expert** | Taster beleuchtet, 3 Funktionen | PushButton_w_LED_BL_0_3 | „Knopf Druck Aktion“ mit drei Zustanden und einzelner LED im Taster |
| | **Expert** | Taster beleuchtet, 4 Funktionen | PushButton_w_LED_BL_0_4 | „Knopf Druck Aktion“ mit vier Zustanden und einzelner LED im Taster |
| | **Expert** | Taster beleuchtet, 5 Funktionen | PushButton_w_LED_BL_0_5 | „Knopf Druck Aktion“ mit fünf Zustanden und einzelner LED im Taster |

| @ | Expert | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|---|
| | | **2 Taster beleuchtet** | | |
| | **Expert** | 2 Taster beleuchtet, 1 Funktion | PushButton2I_w_LED_BL_0_1 | „Knopf Druck Aktion“ mit einen Zustand, zwei Eingängen und einzelner LED im Taster |
| | **Expert** | 2 Taster beleuchtet, 2 Funktionen | PushButton2I_w_LED_BL_0_2 | „Knopf Druck Aktion“ mit zwei Zuständen, zwei Eingängen und einzelner LED im Taster |
| | **Expert** | 2 Taster beleuchtet, 3 Funktionen | PushButton2I_w_LED_BL_0_3 | „Knopf Druck Aktion“ mit drei Zustanden, zwei Eingängen und einzelner LED im Taster |
| | **Expert** | 2 Taster beleuchtet, 4 Funktionen | PushButton2I_w_LED_BL_0_4 | „Knopf Druck Aktion“ mit vier Zustanden, zwei Eingängen und einzelner LED im Taster |
| | **Expert** | 2 Taster beleuchtet, 5 Funktionen | PushButton2I_w_LED_BL_0_5 | „Knopf Druck Aktion“ mit fünf Zustanden, zwei Eingängen und einzelner LED im Taster |
| | | **RGB-Taster** | | |
| | **Expert** | RGB-Taster, 1 Funktion | PushButton_RGB_0_1 | „Knopf Druck Aktion“ mit einem Zustand und RGB LED für den Taster |
| | **Expert** | RGB-Taster, 2 Funktionen | PushButton_RGB_0_2 | „Knopf Druck Aktion“ mit zwei Zuständen und RGB LED für den Taster |
| | **Expert** | RGB-Taster, 3 Funktionen | PushButton_RGB_0_3 | „Knopf Druck Aktion“ mit drei Zustanden und RGB LED für den Taster |
| | **Expert** | RGB-Taster, 4 Funktionen | PushButton_RGB_0_4 | „Knopf Druck Aktion“ mit vier Zustanden und RGB LED für den Taster |
| | **Expert** | RGB-Taster, 5 Funktionen | PushButton_RGB_0_5 | „Knopf Druck Aktion“ mit fünf Zustanden und RGB LED für den Taster |
| | | **2 RGB-Taster** | | |
| | **Expert** | 2 RGB-Taster, 1 Funktion | PushButton2I_RGB_0_1 | „Knopf Druck Aktion“ mit zwei Eingängen und einem Zustand und RGB LED für den Taster |
| | **Expert** | 2 RGB-Taster, 2 Funktionen | PushButton2I_RGB_0_2 | „Knopf Druck Aktion“ mit zwei Eingängen und zwei Zuständen und RGB LED für den Taster |
| | **Expert** | 2 RGB-Taster, 3 Funktionen | PushButton2I_RGB_0_3 | „Knopf Druck Aktion“ mit zwei Eingängen und drei Zustanden und RGB LED für den Taster |
| | **Expert** | 2 RGB-Taster, 4 Funktionen | PushButton2I_RGB_0_4 | „Knopf Druck Aktion“ mit zwei Eingängen und vier Zustanden und RGB LED für den Taster |
| | **Expert** | 2 RGB-Taster, 5 Funktionen | PushButton2I_RGB_0_5 | „Knopf Druck Aktion“ mit zwei Eingängen und fünf Zustanden und RGB LED für den Taster |
| | | **Taster unbeleuchtet** | | |

| @ | Expert | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|---|
| | Expert | Taster unbeleuchtet, 1 Funktion | PushButton_0_1 | „Knopf Druck Aktion“ mit einem Zustand ohne LED Ansteuerung für Taster |
| | Expert | Taster unbeleuchtet, 2 Funktionen | PushButton_0_2 | „Knopf Druck Aktion“ mit zwei Zuständen ohne LED Ansteuerung für Taster |
| | Expert | Taster unbeleuchtet, 3 Funktionen | PushButton_0_3 | „Knopf Druck Aktion“ mit drei Zustanden ohne LED Ansteuerung für Taster |
| | Expert | Taster unbeleuchtet, 4 Funktionen | PushButton_0_4 | „Knopf Druck Aktion“ mit vier Zustanden ohne LED Ansteuerung für Taster |
| | Expert | Taster unbeleuchtet, 5 Funktionen | PushButton_0_5 | „Knopf Druck Aktion“ mit fünf Zustanden ohne LED Ansteuerung für Taster |
| | | **2 Taster unbeleuchtet** | | |
| | Expert | 2 Taster unbeleuchtet, 1 Funktion | PushButton2I_0_1 | „Knopf Druck Aktion“ mit einem Zustand und zwei Eingängen ohne LED Ansteuerung für Taster |
| | Expert | 2 Taster unbeleuchtet, 2 Funktionen | PushButton2I_0_2 | „Knopf Druck Aktion“ mit zwei Zuständen und zwei Eingängen ohne LED Ansteuerung für Taster |
| | Expert | 2 Taster unbeleuchtet, 3 Funktionen | PushButton2I_0_3 | „Knopf Druck Aktion“ mit drei Zustanden und zwei Eingängen ohne LED Ansteuerung für Taster |
| | Expert | 2 Taster unbeleuchtet, 4 Funktionen | PushButton2I_0_4 | „Knopf Druck Aktion“ mit vier Zustanden und zwei Eingängen ohne LED Ansteuerung für Taster |
| | Expert | 2 Taster unbeleuchtet, 5 Funktionen | PushButton2I_0_5 | „Knopf Druck Aktion“ mit fünf Zustanden und zwei Eingängen ohne LED Ansteuerung für Taster |
| | | **Status-LED** | | |
| | Expert | LED eines Tasters, 1 Funktion | Status_Button_0_1 | Steuert die LED in einem Taster mit einem Zustand (Hintergrundbeleuchtung und Blinken) |
| | Expert | LED eines Tasters, 2 Funktionen | Status_Button_0_2 | Steuert die LED in einem Taster mit zwei Zuständen (Hintergrundbeleuchtung und Blinken) |
| | Expert | LED eines Tasters, 3 Funktionen | Status_Button_0_3 | Steuert die LED in einem Taster mit drei Zuständen (Hintergrundbeleuchtung und Blinken) |
| | Expert | LED eines Tasters, 4 Funktionen | Status_Button_0_4 | Steuert die LED in einem Taster mit vier Zuständen (Hintergrundbeleuchtung und Blinken) |

| @ | Expert | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|---|
| | Expert | LED eines Tasters, 5 Funktionen | Status_Button_0_5 | Steuert die LED in einem Taster mit fünf Zuständen (Hintergrundbeleuchtung und Blinken) |

## Konfiguration

Die Verwendung des Expertenmodus wird nur erfahrenen Benutzern empfohlen!

| @ | Expert | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|---|
| | | Letzte Zustände speichern | ENABLE_STORE_STATUS() | Aktiviert die Speicherung der letzten Zustände bei DCC, Selectrix und CAN Kommandos. |
| | | LEDs der Hauptplatine steuern | Mainboard_LED | Ansteuerung der LEDs auf der Hauptplatine über DCC, Selectrix, CAN oder Variablen. |
| | | **Pinnummern** | | |
| | Expert | Pins Schalter Gruppe A definieren | Set_SwitchA_InpLst | Definiert die verwendeten Eingangs Pins für die analogen Schalter (Gruppe A) |
| | Expert | Pins Schalter Gruppe B definieren | Set_SwitchB_InpLst | Definiert die verwendeten Eingangs Pins für die Schalter der Gruppe B (Border). |
| | Expert | Pins Schalter Gruppe C definieren | Set_SwitchC_InpLst | Definiert die verwendeten Eingangs Pins für die Schalter der Gruppe C (Console). |
| | Expert | Pins Schalter Gruppe D definieren | Set_SwitchD_InpLst | Definiert die verwendeten Eingangs Pins für die Schalter der Gruppe D (Direkt auf Hauptplatine). |
| | Expert | Pins Takt-Leitung definieren | Set_CLK_Pin_Number | Definiert den verwendeten Eingangs Pin der Takt Leitung für die PushButton_4017 Platinen. |
| | Expert | Pins Reset-Leitung definieren | Set_RST_Pin_Number | Definiert den verwendeten Eingangs Pin der Reset Leitung für die PushButton_4017 Platinen. |
| | Expert | Pin Fotowiderstand definieren | Set_LDR_Pin_Number | Definiert den Verwendeten Eingangs Pin für den Helligkeitssensor |
| | Expert | Pins LED Bus definieren | Set_LED_OutpPinLst | Definiert die Ausgangs Pins zur Ansteuerung der LEDs |
| | | **Tag/Nacht** | | |
| | | Fotowiderstand aktivieren | READ_LDR | Aktiviert das Einlesen der Helligkeitswerte über einen LDR. |
| | | Helligkeitswerte anzeigen | READ_LDR_DEBUG | Anzeige der Helligkeitswerte zu Testzwecken. |

| @ | Expert | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|---|
| | | Tag/Nacht-Modus aktivieren | DayAndNightTimer | Aktiviert den Uhrzeit gesteuerten Tag/Nacht Modus für die Schedule-Funktion. |
| | | Uhrzeit beibehalten bei Tag/Nacht | KeepDarknessCtr | Uhrzeit wird nicht neu initialisiert beim Umschalten von Tag auf Nacht |
| | | Tageszeiten anzeigen | DayAndNightTimer_Debug | Anzeige der Tageszeiten zu Testzwecken |
| | | Interface-Telegramme anzeigen | COMMANDS_DEBUG | Anzeige von Interface Telegrammen und Aktionen zu Testzwecken. |

## Erweiterungen

Die Verwendung des Expertenmodus wird nur erfahrenen Benutzern empfohlen!

| @ | Expert | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|---|
| | **Expert** | ESP32 Erweiterung | ESP_Extentions | Erweiterungen für den ESP32 (Experimental) |
| | **Expert** | DMX512 Protokoll verwenden | Use DMX512 | LED Kanal mit DMX512 Protokoll |

## Verwendung der Testtaster und Funktionen

Diese Funktionen stehen ab Version 3.0.0 der MobaLedLib nicht mehr zur Verfügung.

| Name der Tasterfunktion | Kurzbeschreibung |
|---|---|
| TEST_PUSH_BUTTONS | Test Taster: Mit den drei Tasten des Mainboards können DCC/Selectrix, ... Befehle simuliert werden. Die Taster werden als Taster betrieben. |
| TEST_TOGGLE_BUTTONS | Test Schalter: Mit den drei Tasten des Mainboards können DCC/Selectrix, ... Befehle simuliert werden. Die Taster werden als Ein-/Ausschalter betrieben. |
| TEST_BUTTONS_INCH | Änderung der zugeordneten „DCC Zeile“: Ändert die erste zugeordnete „DCC Zeilen“. |

### Noch nicht zugeordnet

| @ | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|
| | Sound_ADKey | Manuelle Steuerung | Manuelles Abrufen einer Sound Funktion anhand individueller Helligkeitswerte. |
| | Sound_JQ6500_ADKey | Manuelle Steuerung | Manuelles Abrufen einer Sound Funktion für das JQ6500 Modul anhand individueller Helligkeitswerte. |

| @ | Name des Effekts | Befehl | Kurzbeschreibung |
|---|---|---|---|
| | DEBUG_SOUND_CHANNEL BitField | wie oben plus serielle Soundmodule | Anzeige von zusätzlichen Meldungen bei der Verwendung von seriellen Sound Modulen, die an Hautpplatine angeschlossen sind. |

1)

Apache OpenOffice, LibreOffice, usw.

2)

https://support.office.com/de-de/article/arbeiten-mit-vba-makros-in-excel-f%C3%BCr-das-web-98784ad0-898c-43aa-a1da-4f0fb5014343

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

Permanent link:
**https://wiki.mobaledlib.de/anleitungen/programmgenerator?rev=1710712340**

Last update: **2024/03/17 21:52**