Virtuelle LED Kanäle

In Arbeit
Under construction

Virtuelle Kanäle eignen sich, um versteckte Operationen auszuführen, die man für andere Funktionen wieder abfragen kann.
So können beispielsweise eigene Zeitabläufe auf einem virtuellen LED-Kanal erfolgen, ohne dafür einen echten WS2811 in der LED-Kette zu belegen.
Den Zeitablauf kann man wiederum mit einer Variable abrufen und zum Schalten echter WS2811/12 auf jedem anderen Kanal verwenden.

Einleitung

Was kann man damit machen?

Das Thema hört sich im ersten Moment hochkomplex an und ohne Beispiel fehlt oft der Bezug zur Verwendbarkeit.
Daher startet dieser Beitrag mit einem Beispiel:

Im konkreten Fall sollten drei Abschnitte à acht Neonlampen eines Bahnsteigs mit kurzer Zeitverzögerung nacheinander angehen.
Starten wir mit dem Ergebnis:

Hier werden zwei Funktionen der MobaLedLib miteinander vereint.

1. Der **Pattern Configurator** gibt den zeitlichen Ablauf vor, indem er mit jeweils einer Sekunde Verzögerung die Kanäle Rot, Grün und Blau an einem virtuellen WS2811 einschaltet.
   Dieser Ablauf wird bewusst auf einen virtuellen Kanal ausgelagert, damit man diesen WS2811 nicht wirklich einlöten muss. Er existiert nur in der virtuellen Welt.
   Die MobaLedLib kann im Programm jederzeit die Schaltzustände dieses WS2811 abrufen, obwohl er physisch gar nicht existiert.
2. Das **belebte Haus** zündet innerhalb weniger Millisekunden die jeweils acht LEDs eines Bahnsteigdachs.

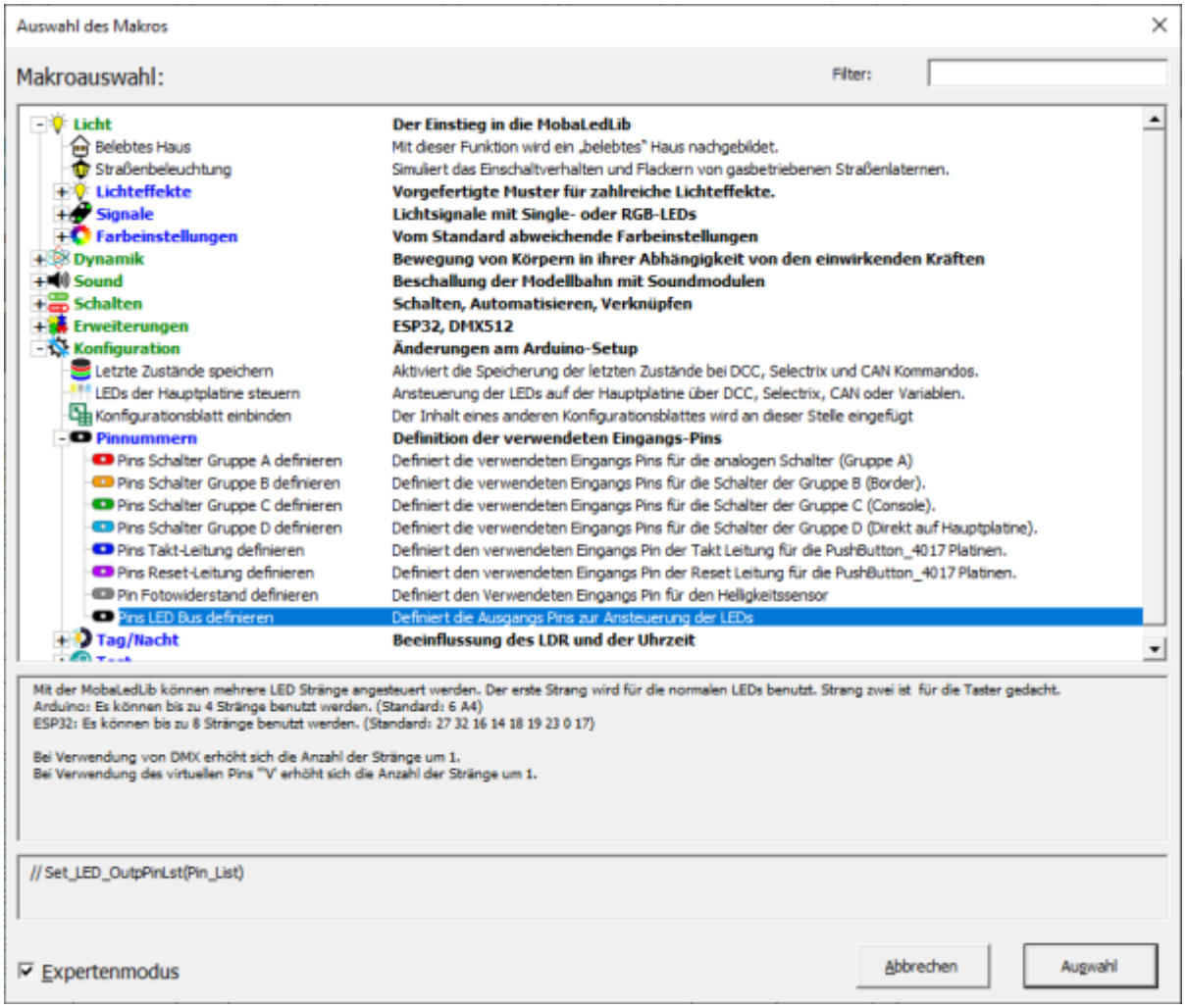
Vereint werden diese beiden Funktionen später mit dem Befehl LED-Werte als Variable.
Mit dieser Funktion kann man die Zustände anderer LEDs abfragen und bei bestimmten Zuständen Aktionen auslösen. Dazu später mehr.

Wie aktiviert man virtuelle Kanäle?

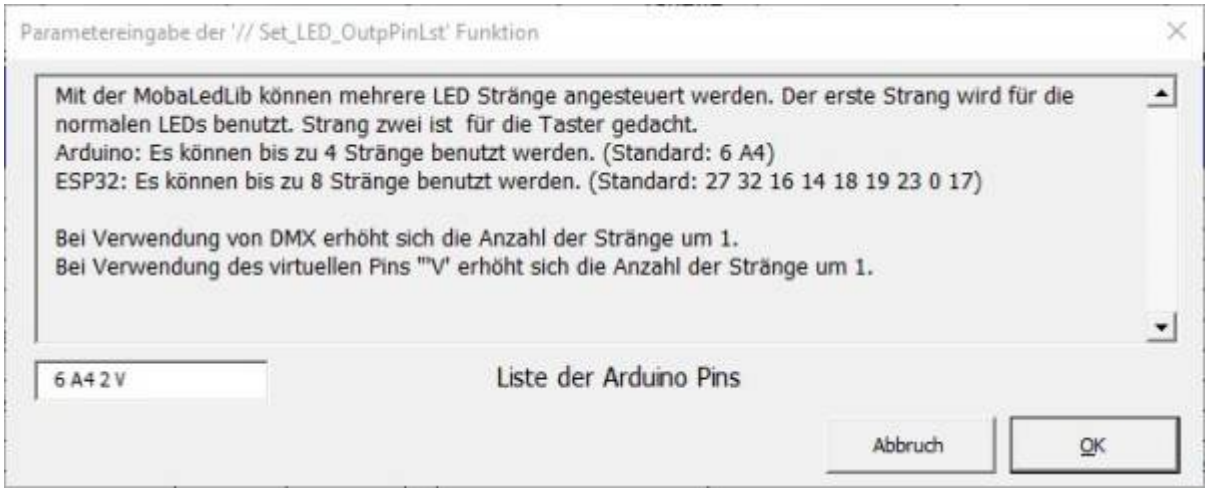
Virtuelle Kanäle kann man ganz einfach zusätzlich zu den echten Kanälen definieren. Das geht mit der Funktion Pins LED Bus definieren

Zunächst wählt man die erste Zeile im aktuellen Excel-Sheet. Dort sollte der Befehl **Pins LED Bus definieren** stehen:
Beim Arduino nano sollte in der Spalte rechts daneben folgender Eintrag stehen:
Set_LED_OutpPinLst(6 A4)
Beim ESP32/Pico sollte in der Spalte rechts daneben folgender Eintrag stehen:
Set_LED_OutpPinLst(27 32 16 14 18 19 23 0)
Die 6 steht für den digitalen Pin D6, A4 steht für den analogen Pin A4. Das sind die beiden Arduino Pins, an denen die Kanäle LED #0 und Push Button #1 der alten Hauptplatine hängen.
Diese Zeile bitte per Doppelklick öffnen.

Bei aktiviertem Expertenmodus findet man unter Konfiguration > Pin-Nummern nun den Eintrag **Pins LED-Bus definieren**.



Für den Fall, dass alle vorhandenen LED-Kanäle verwendet werden sollen, muss man zunächst alle verwendeten oder einfach alle möglichen LED-Kanäle des jeweiligen Arduinos/ESPs definieren und zusätzlich den virtuellen Kanal mit einem V bezeichnen.
Schauen wir uns das am besten am Beispiel eines Arduinos an. Im folgenden Bild wurden der LED-Kanal 0 (Arduino-Pin: 6), der PushButtonKanal 1 (Arduino-Pin: A4), der LED-Kanal2 (Arduino-Pin: 2) und der virtuelle Kanal V extra definiert.



Welcher Kanal ist der virtuelle Kanal?

Die Herleitung ist ganz einfach. Wie oben bereits beschrieben, liegt der LED-Kanal 0 am Arduino-Pin 6. Wenn wir im Programm Generator nach dem LED-Kanal gefragt werden, tragen wir dort standardmäßig die „0“ ein.
Für den Push-Button Kanal am Arduino-Pin A4 tragen wir die „1“ ein.
Für den zweiten LED-Kanal am Arduino-Pin 2 tragen wir die „2“ ein.

Der Programm Generator erwartet beim LED-Kanal also einfach die Nummer des Kanals, **NICHT** den Namen.
Wenn also wie in unserem Beispiel alle physischen Kanäle des Arduinos verwendet werden (LED #0, PushButton #1 & LED #2), dann ist der virtuelle Kanal eben Nummer #3.
Wird der LED-Kanal 2 des Arduinos nicht als echter Ausgang genutzt, dann wäre der virtuelle Kanal Nummer #2. Die Liste der Arduino Pins wäre dann „6 A4 V“. Ganz einfach.

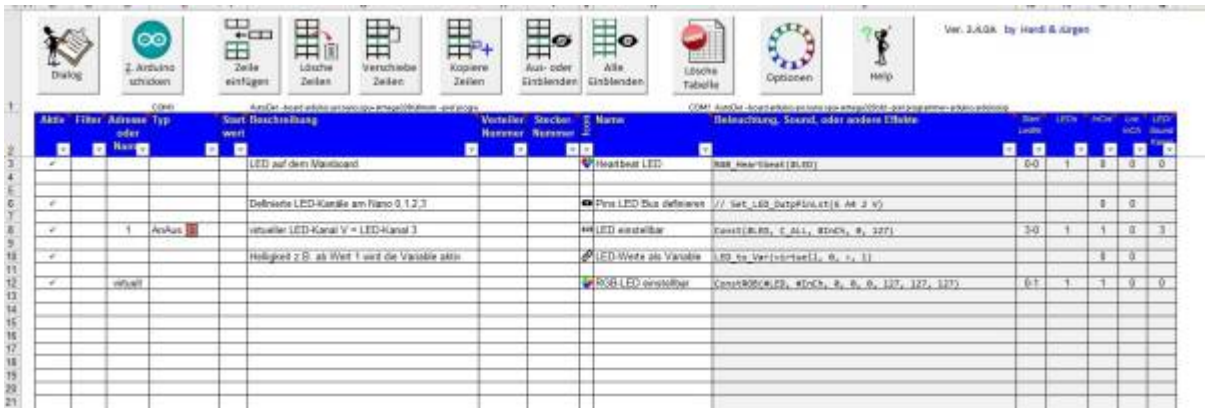
Wie verwendet man den virtuellen Kanal?

In Arbeit
Under construction

Die Einrichtung liegt nun hinter uns. Jetzt kommt der virtuelle Kanal in einem Beispiel als versteckter Schalter in einem einfachen Beispiel zum Einsatz.

In der Programmierung ist der virtuelle Kanal also Kanal 3.
Ablauf: Es wird mit der DCC Adresse 1 die LED auf dem virtuellen Kanal 3 eingeschaltet und sobald der Helligkeitswert größer als 1 ist dann wird die Variable „virtuell“ aktiv und die RGB-LED auf dem LED-Kanal 0 beginnt zu leuchten.

Für den ESP gilt das Gleiche, hier müssen auch die verwendeten Kanäle plus dem V-Kanal extra definiert werden.

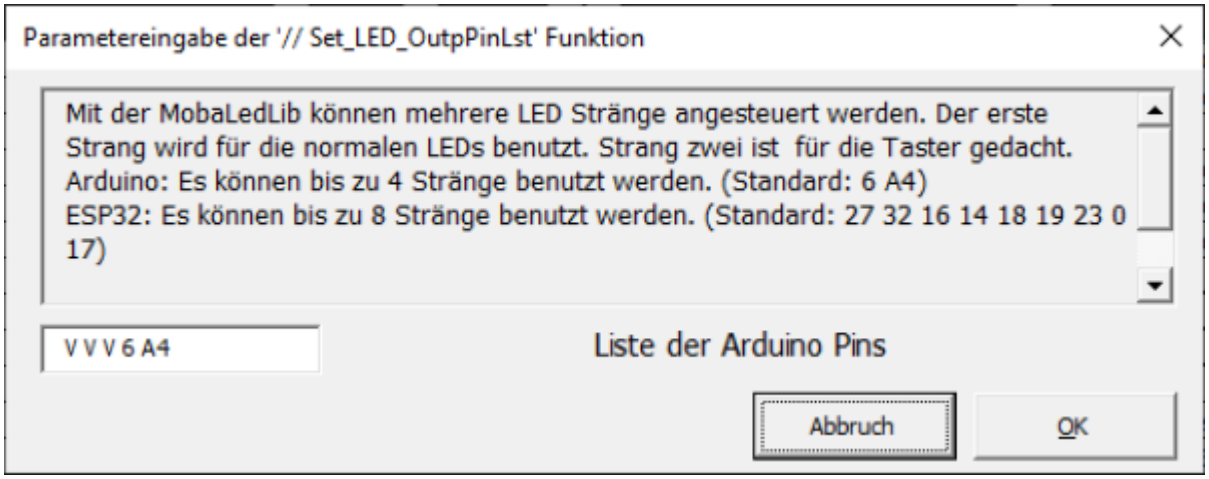


Virtuelle Kanäle zur Manipulation der Kanal-Nummer

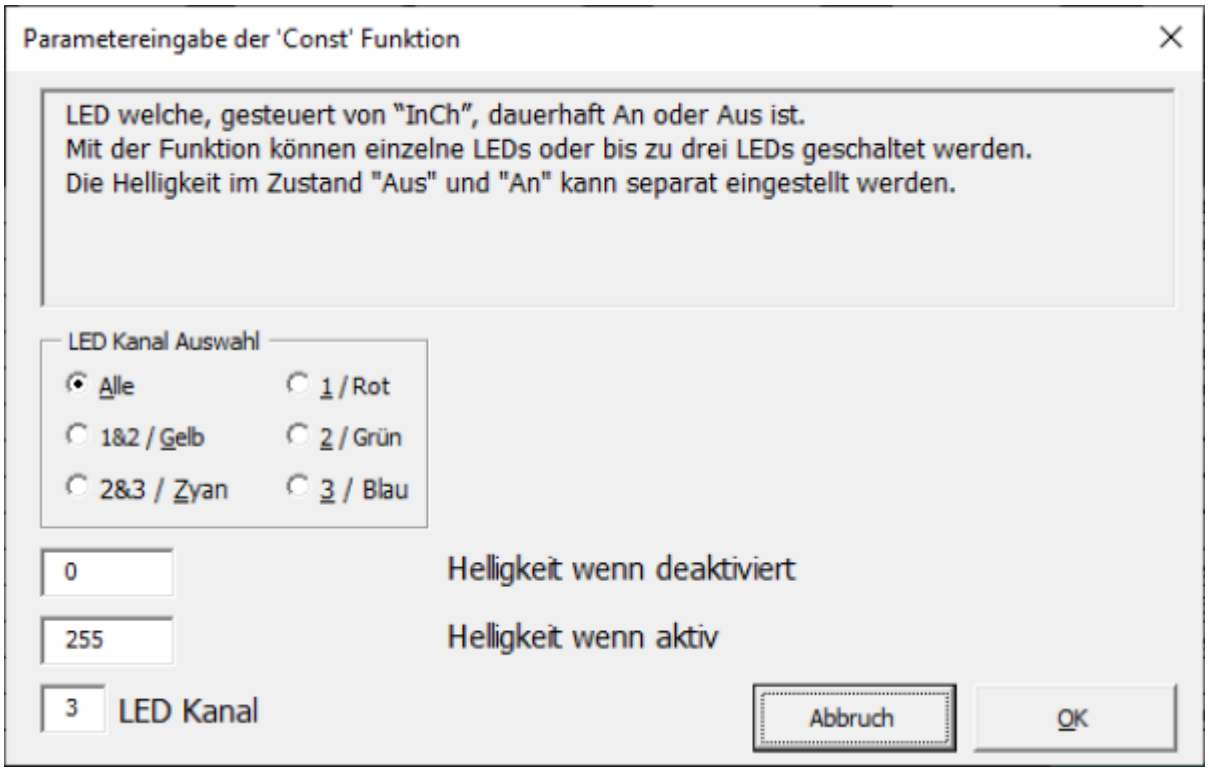
In Arbeit
Under construction

Wenn du weißt, dass du ein Haus für Kanal 3 bauen willst, musst du einfach die Kanäle 0, 1 und 2 an der alten Hauptplatine überspringen. Da diese Kanäle aber physisch nicht existieren, musst du vor dem eigentlichen Pin 6 (Das ist Kanal 0 an der Werkstatt-Platine) drei virtuelle Kanäle setzen. Klingt kompliziert, ist aber super easy.

Dort gibst du jetzt für jeden Kanal, den du an der LichtMaschine Pro später überspringen willst ein „V“ gefolgt von einem Leerzeichen ein, beginnend mit Kanal 0. Soll also Kanal 3 der LichtMaschine Pro verwendet werden, musst du Kanal 0, 1 und 2 überspringen. Das sind dann drei „V“irtuelle Kanäle. Erst dann folgen die Pins 6 (Kanal3) und Pin A4 (Kanal 4). Probleme könnte es hier mit Push Buttons geben, die durch die virtuellen Kanäle nicht mehr auf Kanal 1 sondern auf Kanal 4 liegen. Das ist aber wahrscheinlich ein Randthema.



Deine Werkbank-Platine belegt nun die Kanäle 0, 1 und 2 rein virtuell. Erst auf Kanal 3 lässt sie die Ausgabe am Pin D6 zu. Nun beginnst du mit deiner Programmierung des zu bauenden Objekts und definierst jeden Effekt auf Kanal 3, **beginnend mit der Heartbeat-LED deiner Hauptplatine!** Es folgt zum Beispiel eine konstante LED in weiß.



Obwohl du jetzt per Definition Kanal 3 verwendest, kannst du in der Werkstatt alle Effekte am

normalen LED-Kanal der Hauptplatine nutzen und somit testen, denn diese denkt ja, dass der Wannenstecker Kanal 3 ist. Wenn du dann mit dem Bau deines Häuschens und der Programmierung fertig bist, brauchst dur nur noch die entsprechenden Zeilen in das Programm deiner LichtMaschine Pro zu übertragen. Ohne Anpassungen, ohne Risiko und ohne zusätzliche Fehlerquellen.