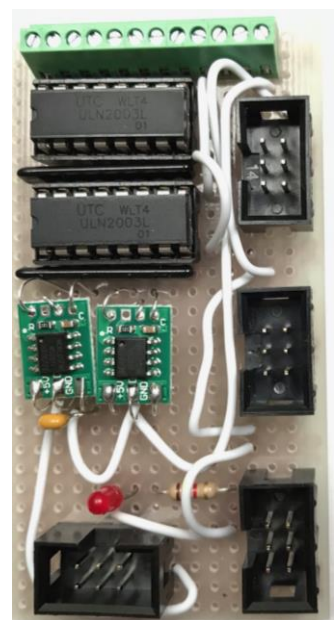
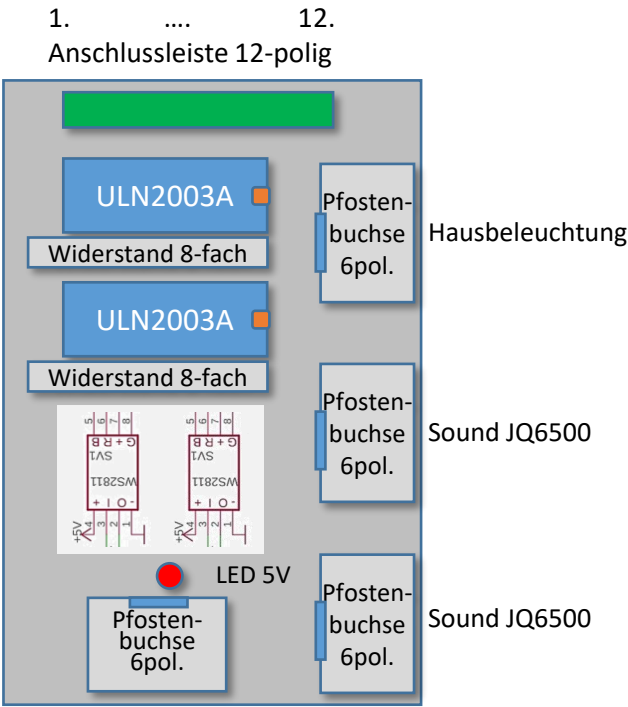


Verteilerplatine VP-3 (Straßenbeleuchtung, WS2812-Spezialeffekte, Sound)

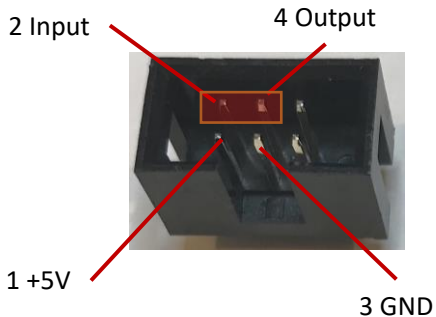
- Anschlüsse:**
- 1. Leuchte 1
 - 2. Leuchte 2
 - 3. Leuchte 3
 - 4. Leuchte 4
 - 5. Leuchte 5
 - 6. Leuchte 6
 - 7. -
 - 8. +15 V
 - 9. +15 V
 - 10. +15V
 - 11. +15V
 - 12. 0V / GND



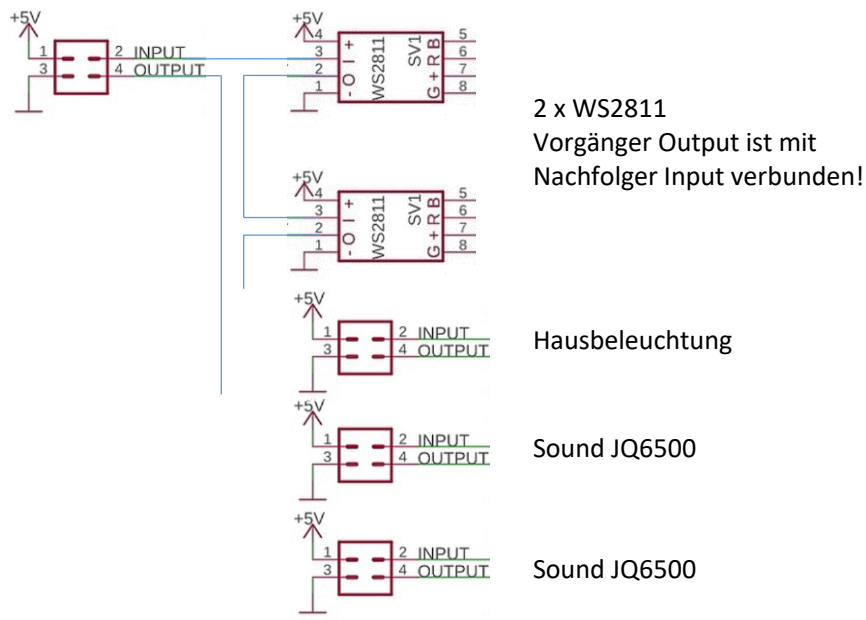
200er Verteiler



- Pfostenbuchse 6polig**
- 1. +5V
 - 2. Input
 - 3. GND
 - 4. Output
 - 5. -
 - 6. -

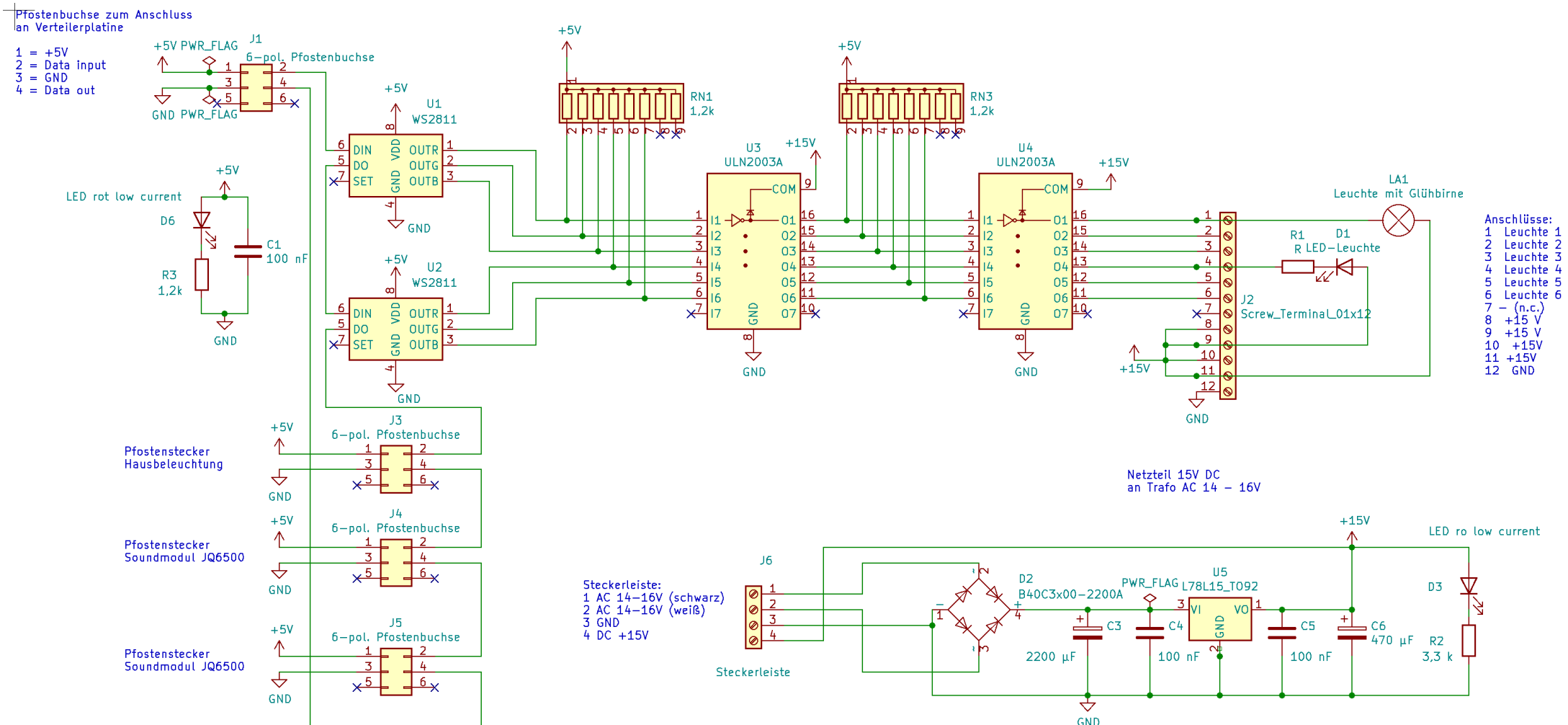


Nicht benötigte Ausgänge:
In Pfostenbuchse Kontakte 2 und 4 mit Jumper überbrücken!



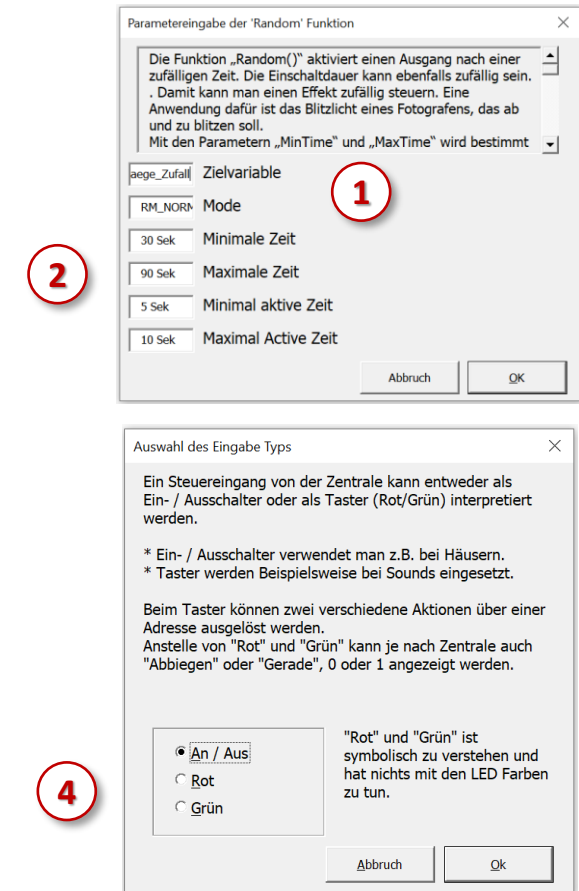
Output des 2. WS2811 an Pfostenbuchse für Hausbeleuchtung,
Von dort zur Buchse des 1. Soundanschlusses usw. und dann zurück zum Verteiler und den nächsten WS2812/WS2811

Verteilerplatine VP-3 (Straßenbeleuchtung, WS2812-Spezialeffekte, Sound)



MobaLedLib - Zufällige Aktionen über DCC steuern

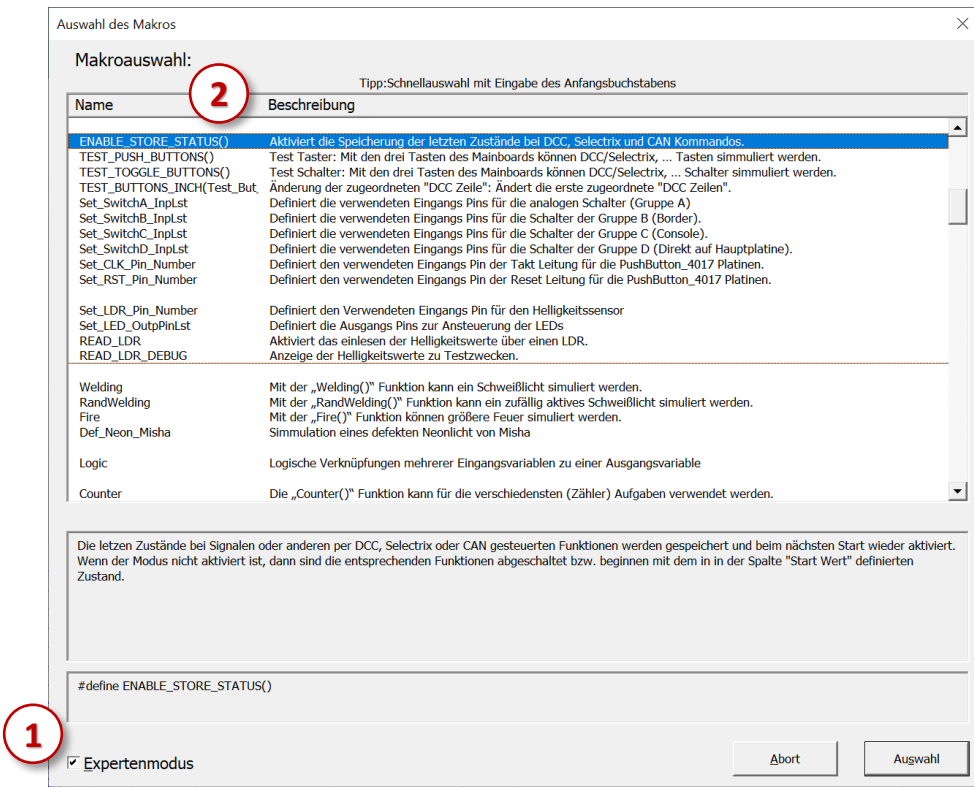
- Use Case: Ein WS2811-Kanal soll über DCC innerhalb eines Zeitintervalls (Min, Max) zufällig eingeschaltet werden und zwar für eine zufällige Zeitdauer (Min, Max). Damit soll z.B. das motorisierte Sägegatter im Sägewerk zufällig eine bestimmte (ebenfalls zufällige) Zeit lang laufen. Danach schaltet es sich aus und wird anschließend wieder zufällig eingeschaltet usw. Solange die DCC-Adresse nicht abgeschaltet wird, wiederholt sich das zufällig Ein- und Ausschalten.
- Hierfür sind zwei Zeilen im Programm Generator Excel notwendig.
- Verwendung des Random-Befehls in der 1. Zeile (hierfür muss man im Dialogfenster den Expertenmodus auswählen); in der Zielvariablen des Makros den Namen einer Variablen eingeben (Saeger_Zufall) **1**:
 - Bei mehreren zufälligen Aktionen in der LED-Kette im Excel, müssen verschiedene Namen gewählt werden!
 - Mode = „RM_NORMAL“
 - Der Variablenname muss der Arduino-Nomenklatur entsprechen.
- Die vier Sekundenwerte des Random-Befehls geben an, in welchem Zeitintervall das Einschalten ausgelöst wird und für wie lange. Einschaltzeitpunkt und –dauer werden dann zufällig berechnet **2**. Minimale und maximale Zeit können auf den gleichen Wert gesetzt werden, damit erzeugt man eine konstante Frequenz. Das gleiche gilt auch für die beiden aktiven Zeiten; hier ist die Einschaltdauer dann konstant.
- In Spalte „Adresse oder Name“ die DCC-Adresse eingeben, hier „301“ **3**.
- Im Folgenfenster den Typ auf „An/Aus“ einstellen **4**. Die rote Null bedeutet, dass die Funktion standardmäßig aus ist.
- In der 2. Zeile bei „Adresse oder Name“ den oben vergebenen Variablennamen eingeben, hier „Saeger_Zufall“ **5**.
- Die Const()-Funktion schaltet den Vorgang dann ein bzw. aus, wenn die Variable entsprechend gesetzt ist. Das #InCh wird im Arduino-Programm später durch den Variablennamen „Saeger_Zufall“ ersetzt **6**.



Aktiv	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte	Start LED Nr	LEDs	InCnt	Loc InCh	LED Kanal	Start Taster LED	Start LED G2	Start LED G3
✓	VP-1	301	AnAus 0		Zufälliges Einschalten Sägegatter für zufällige Zeitdauer	1		Random(Saeger_Zufall, #InCh, RM_NORMAL, 30 Sek, 90 Sek, 5 Sek, 10 Sek)	2	C2-2	1	0	0			
✓	VP-1	Saeger_Zufall			Zufälliges Einschalten Sägegatter für zufällige Zeitdauer; wichtig: Input-Variable muss manuell in Spalte D eingetragen werden, ersetzt dann #InCh in Spalte J.	1		Const(#LED, C2, #InCh, 0, 255)								

MobaLedLib – Speichern der LED-Zustände im EEPROM

- Use Case: Wie kann ich den jeweiligen Schaltzustand einer Beleuchtung etc. im EEPROM des Arduinos speichern, damit beim nächsten Einschalten des Arduinos die Funktionen den gleichen Status haben wie vor dem letzten Ausschalten? Dies ist v.a. bei DCC-Ansteuerung interessant, damit die Zentrale und der LED-Arduino den gleichen Stand haben (die Zentrale speichert die Zustände normalerweise ab).
- Hierfür muss man im Dialogfenster zur Auswahl des Makros den Expertenmodus auswählen ①.
- Dann den Eintrag `ENABLE_STORE_STATUS()` auswählen ②.
- Im Excel wird dann „`#define ENABLE_STORE_STATUS`“ eingetragen ③. Der Status aller Folge-Funktionen wird dann abgespeichert und beim Start oder Reset des LED-Arduinos werden die Funktionen entsprechend aktiviert oder bleiben ausgeschaltet.
- Der Befehl „`#define ENABLE_STORE_STATUS`“ ist nur einmal im Excel erforderlich. Am besten gleich am Anfang der Konfigurationen.



Aktiv	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte	Start LedNr	LEDs	InCnt	Loc InCh	LED Kanal	Start Taster LED	Start LED G2	Start LED G3
✓	ALL				Zeigt an, dass die LEDs angesteuert werden	1		RGB_Heartbeat(#LED)		1	0	0				
✓	ALL				Speichern der Zustände aktivieren, damit beim erneuten Einschalten des Arduinos die ECOS-Anzeige und die LED-Funktionen übereinstimmen. Macht insbesondere bei DCC-gesteuerten Effekten Sinn.			#define ENABLE_STORE_STATUS			0	0				

MobaLedLib – LED-Kanäle anpassen

- Use Case: Die auswählbaren Makros im Excel setzen automatisch die Nummer der LED für den nächsten Effekt (= Zeile im Excel) hoch. Verwendet man einzelne Kanäle von WS2811-Chips, so kann es sein, dass bereits die nächste LED gewählt wird, obwohl noch ein Kanal auf dem WS2811 frei ist, z.B. der B-Kanal.
- Hat man z.B. 4 Straßenbeleuchtungen an 2 x WS2811 angeschlossen, so sind der G- und der B-Kanal des zweiten WS2811 noch frei verfügbar. Im Programm Generator wird aber für Folgefunktionen gleich die 3. LED verwendet.
- Man muss vor dem nächsten Effekt den LED-Zähler manuell anpassen. Hierfür muss man im Dialogfenster zur Auswahl des Makros den Expertenmodus auswählen ①.
- Dann den Eintrag „Next_LEDs“ auswählen ②. Im nächsten Fenster die StartLedNr um eins verringern („-1“) ③.
- Im Excel wird dann „// Next_LED(-1)“ eingetragen ④.
- Die Zählung geht dann im G-Kanal von LED 2 weiter („C2-2“) ⑤ anstatt mit LED Nummer 3.

Auswahl des Makros

Makroauswahl:

Tipp:Schnellauswahl mit Eingabe des Anfangsbuchstabens

Name	Beschreibung
Set_LED_OutPinLst	Definiert die Ausgangs Pins zur Ansteuerung der LEDs
READ_LDR	Aktiviert das Einlesen der Helligkeitswerte über einen LDR.
READ_LDR_DEBUG	Anzeige der Helligkeitswerte zu Testzwecken.
Welding	Mit der „Welding()“ Funktion kann ein Schweißlicht simuliert werden.
RandWelding	Mit der „RandWelding()“ Funktion kann ein zufällig aktives Schweißlicht simuliert werden.
Fire	Mit der „Fire()“ Funktion können größere Feuer simuliert werden.
Def_Neon_Misha	Simulation eines defekten Neonlicht von Misha
Logic	Logische Verknüpfungen mehrerer Eingangsvariablen zu einer Ausgangsvariable
Counter	Die „Counter()“ Funktion kann für die verschiedensten (Zähler) Aufgaben verwendet werden.
RandMux	Die „RandMux()“ Funktion aktiviert zufällig einen der Ausgänge.
Random	Die Funktion „Random()“ aktiviert einen Ausgang nach einer zufälligen Zeit.
RandCntMux	Die „RandCntMux()“ Funktion aktiviert nach einer zufälligen Zeit den nächsten Ausgang.
CopyLED	Mit dem „CopyLED()“ Befehl wird die Helligkeit der drei Farben der „SrcLED“ in die „LED“ kopiert.
Schedule	Mit dem „Schedule“ Makro kann ein Zeitplan für das Ein- und Ausschalten mehrerer Lichter erstellt werden.
New_HSV_Group	Speicher reservieren für eine neuen HSV Gruppe.
New_Local_Var	Erzeugt eine Variable in der 256 verschiedene Zustände gespeichert werden können.
Use_GlobalVar	Mit der Funktion „Use_GlobalVar()“ können die eigenen Programmtile mit den bibliotheksinternen Funktionen Daten austausch.
InCh_to_TmpVar	Mit diesem Befehl wird eine temporäre 8 Bit Variable mit den Werten aus mehreren Logischen Variablen gefüllt.
Bin_InCh_to_TmpVar	Mit diesem Befehl wird eine temporäre 8 Bit Variable mit den binären Werten aus mehreren logischen Variablen gefüllt.
Reserve_LEDs	Mit diesem Eintrag können LEDs reserviert werden.
Next_LEDs	Manipulation der LED Nummer

Damit kann die StartLedNr für den folgenden Befehl verändert werden. Es sind positive und negative Werte möglich.

// Next_LED(LedCnt_Nxt)

☒ Expertenmodus

Parametereingabe der '// Next_LED' Funktion

Damit kann die StartLedNr für den folgenden Befehl verändert werden. Es sind positive und negative Werte möglich.

-1

Änderung von StartLedNr

0

LED Kanal

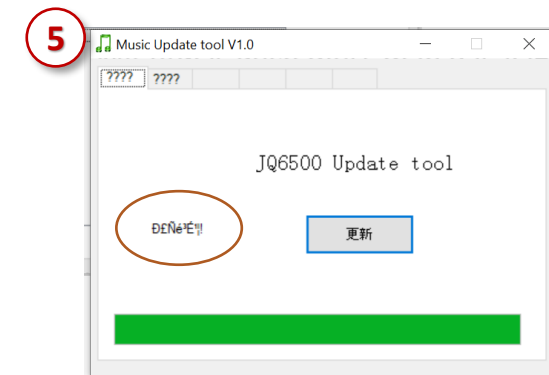
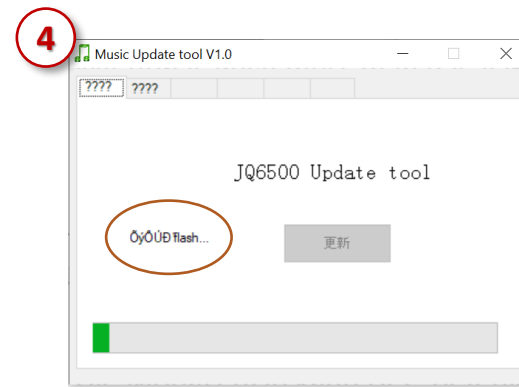
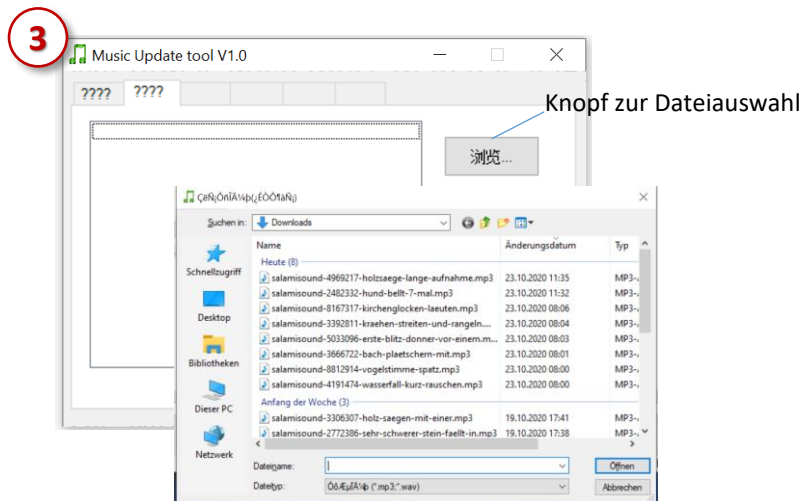
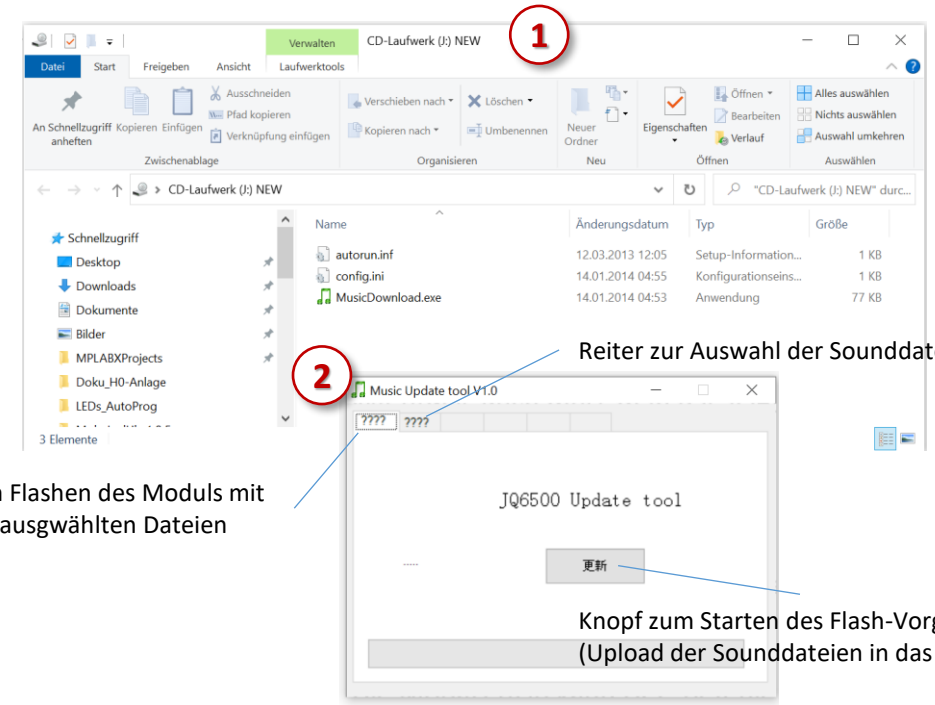
Abbruch

OK

Aktiv	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte	Start LedNr	LEDs	InCnt	Loc InCh	LED Kanal	Start Tact LED	Start LED G2	Start LED G3
✓	VP-1	300	AnAus	0	Straßenbeleuchtung, 2 Leuchten bei der Mühle + 2 Reserve	1		GasLights(#LED, #InCh, GAS_LIGHT1, GAS_LIGHT2, GAS_LIGHT3, GAS_LIGHT1)		2	1	0	0			
✓					LED-Zähler korrigieren, B + G-Kanal sind noch frei			// Next_LED(-1)		-1	0	0	0			
✓	VP-1	301	AnAus	0	Zufälliges Einschalten Sägegatter für zufällige Zeitdauer	1		Random(Saage_Zufall, #InCh, RM_NORMAL, 30 Sek, 90 Sek, 5 Sek, 10 Sek)			1	0				
✓	VP-1	Saage_Zufall			Zufälliges Einschalten Sägegatter für zufällige Zeitdauer; wichtig: Input-Variable muss manuell in Spalte D eingetragen werden, ersetzt dann #InCh in Spalte J.	1		Const(#LED, C2, #InCh, 0, 255)	0	C2-2	1	0	0			

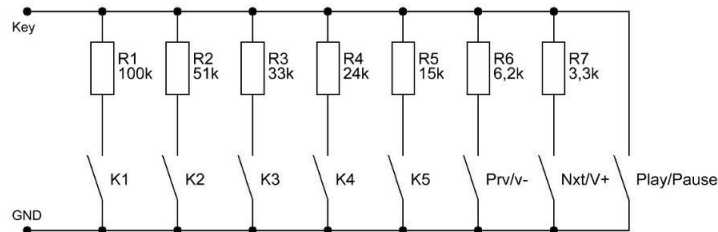
MobaLedLib - Sound-Dateien auf JQ6500 speichern

- Mit etwas Pech muss man das JQ6500 zuerst einmal initialisieren, da es sonst nicht vom PC erkannt wird. Siehe auch MobaLedLib-Thread und diverse Internet-Seiten (Video des Italieners und VMWare).
- JQ6500 via USB an PC anschließen. Wenn man Glück hat, wird das Modul als neues CD-Laufwerk erkannt und kann mit dem Explorer angezeigt werden ①.
- Nach Start der **MusicDownload.exe** erscheint ein spartanisches chinesisches Menü ②.
- Nach Drücken der rechten „????“ können die Sound-Files vom PC ausgewählt werden ③.
- Das Modul hat normalerweise 2 MB Speicher. Überschreitet man diesen mit den ausgewählten Dateien, kann der Upload nicht durchgeführt werden. Es erscheint in chinesische Fehlermeldung.
- Zurück auf dem linken Reiter „????“ den Knopf in der Mitte drücken, um den Upload-Vorgang zu starten. Es erscheint unten ein grüner Balken. In der Mitte links steht sogar das Wort „flash“ ④.
- Wenn der Upload fertig ist, ändert sich der Text und der grüne Balken ist komplett gefüllt ⑤.
- Dann sollten alle Files auf dem JQ6500 gespeichert sein. Dann die App schließen und das USB-Kabel entfernen.
- Welche Dateien aufgespielt wurden, lässt sich hinterher nicht mehr feststellen. Daher am besten die Dinger durchnummerieren (z.B. mit permanent Stift auf dem USB-Anschluss) und die Namen der Sounds irgendwo dokumentieren.

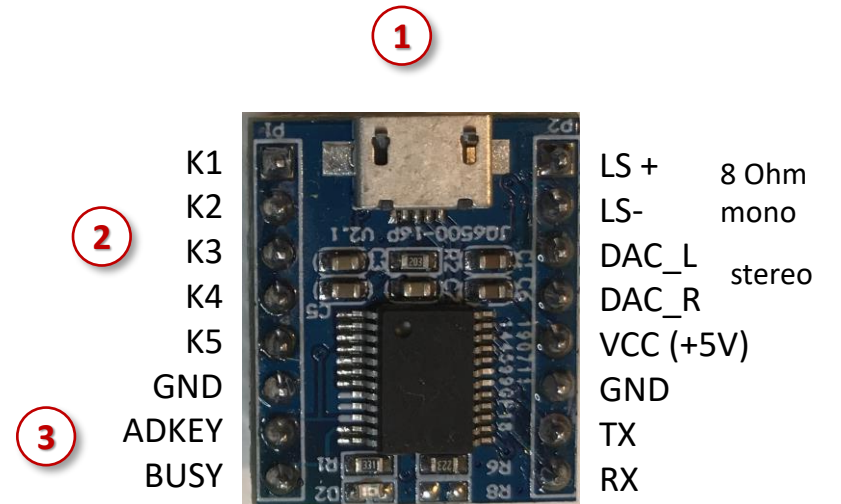


MobaLedLib – Stand alone Test JQ6500

- Nach dem Initialisieren des JQ6500 als CD-Laufwerk und Bespielen mit Soundfiles kann man das Modul stand alone testen. Einfach auf dem Breadboard wie folgt verdrahten ①.
- VCC: Die Stromversorgung des Moduls (5 Volt)
- GND: Dies sind die Masseanschlüsse des Soundmoduls
- LS+ und LS-: Diese Kontakte dienen zum Anschluss eines Lautsprechers (8 Ohm, 3 Watt).
- Die Eingänge K1 bis K5 spielen die ersten 5 Sound-Dateien ab, wenn sie kurz mit GND verbunden werden ②.
- Der Eingang Key oder Adkey reagiert auf einen Spannungsteiler mit unterschiedlichen Widerständen ③.

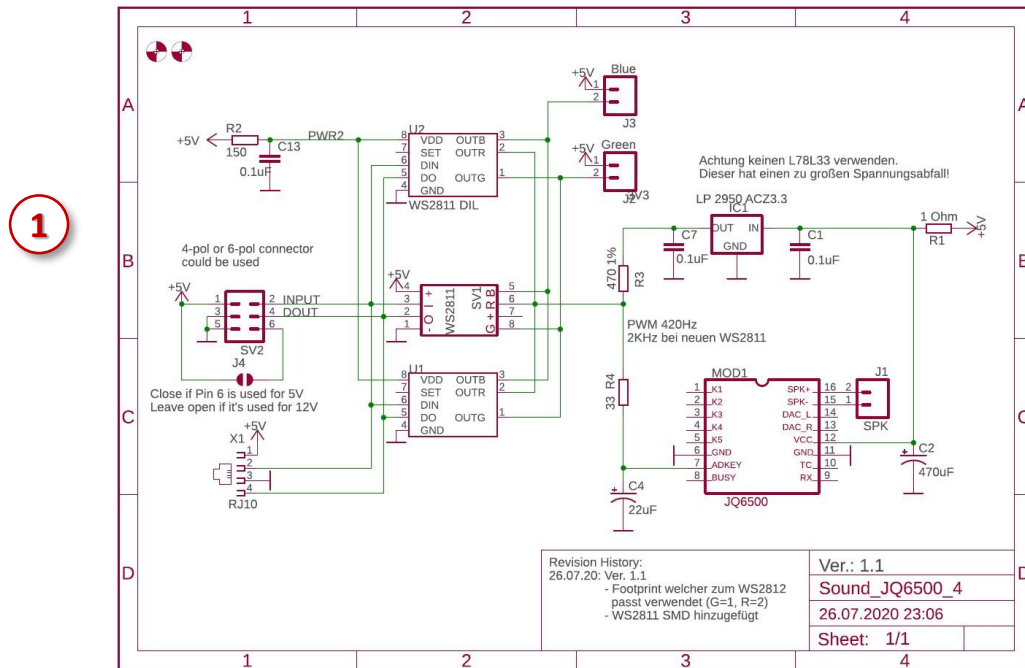


- Mit der Taste Prv/V- lässt sich der vorherige Titel abspielen (kurzer Tastendruck) oder die Lautstärke verringern (langer Tastendruck > 1 sec).
- Die Taste mit der Bezeichnung Nxt/V+ schaltet mit einem kurzen Tastendruck auf den nächsten Titel, bei einem längeren Tastendruck wird die Lautstärke erhöht.
- Beim Abspielen der Datei leuchtet eine eingebaute rote LED auf der JQ6500 Platine (D2 am unteren Rand der Platine)).
- Evtl. prüfen, ob die Lautstärke passt oder zu leise ist, und man deshalb nichts hört.
- Weitere Anschlüsse (nicht benötigt):
 - BS (Busy): Hier kann eine LED angeschlossen werden, die aufleuchtet, während eine Audiodatei abgespielt wird.
 - DAC_L und DAC_R i.V.m. GND: Die Ausgänge für den rechten und linken Kanal (Line-Out oder Kopfhörer), z.B. für den Anschluss an einen externen Verstärker.
 - RX und TX: Serielle Schnittstelle zur Kommunikation mit Mikrocontrollern (zum Beispiel einem Arduino).

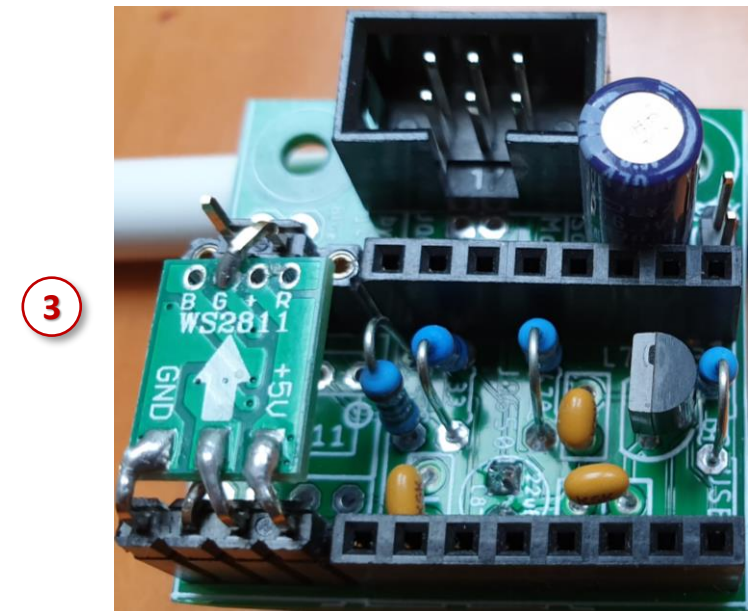


MobaLedLib - JQ6500 auf Platine

- Bestücken der Platine lt. Anleitung. Schaltplan siehe ①. Hier mit WS2811 aus einer Schokoladentafel.
- In die 2 x 4 Buchsenleiste wird das WS2811 gesteckt. Achtung bei manchen WS2811 sind die Anschlüsse vertauscht oder die aufgedruckte Reihenfolge ist falsch ②!
- Ggf. muss man die WS2811 Module separat testen, um herauszufinden, welcher Output R, G oder B ist. Hierzu mit dem Programm Generator einfach ein WS2811 unterschiedlich ansteuern, einmal mit dem R-, dann B- und G-Kanal, Dann sieht man schon, welcher Ausgang was ist. „Const(#LED, C2, #InCh, 0, 255)“ steuert z.B. den 2. Kanal = G(green) an.
- An die WS2811 müssen selbstgebastelte Winkelstecker angelötet werden.
- Die Platine mit dem sichtbaren weißen Pfeil auf der Oberseite verbauen ③.
- B, G und +5V an der oberen Anschlussleiste nicht anschließen. Für Sound ist nur der R-Kanal notwendig.
- Man kann auch ein WS2811 DIL oder die SMD-Variante verwenden. Je nach Verfügbarkeit und Lötkünsten.



Gezeigt wird hier die Unterseite der WS2811-Platine!



MobaLedLib – Test JQ6500 auf Platine

- Für einen ersten Test den Arduino-sketch „24. DCC und Sound“ aufspielen.
- Dazu JQ6500-Modul auswählen (#define-Befehl am Anfang des Programmes auskommentieren).
- Achtung: Das Programm funktioniert in der Originalversion nur, wenn die RGB-LED (Heartbeat) auf der Hauptplatine durch einen R1 = 100 Ohm Widerstand ersetzt wird.
- Ich habe das Programm angepasst, da auf der Hauptplatine LED0 eine RGB_Heartbeat LED eingesteckt ist (LED0).
- Der Sound liegt an LED1, siehe die geänderten Soundanweisungen im Sketch („Sound_JQ6500_Seq1(1, 0)“ etc.).
- Die 5 Kanäle + weitere Funktionen können über den seriellen Arduino-Monitor angesteuert und ggf. kalibriert werden. Das Arduino-Programm sendet PWM-Impulse an das JQ6500 und simuliert damit den Spannungsteiler am ADKEY-Eingang.
 - Zur Aktivierung des Test-Modus gibt man in der „Senden“ Zeile das „seriellen Monitors“ das Paragraph „\$“ Zeichen ein.
 - Mit „L1“ die erste LED wählen. Da wir bei 0 mit der Zählung starten (LED0 = Heartbeat-RGB-LED), ist die LED1 dann das Soundmodul, das direkt an der Hauptplatine angeschlossen ist!
 - Über „P“ + Zahl kann das Tastverhältnis des PWM-Signal eingestellt werden. „P29“ sollte den ersten Sound abrufen.
 - Vordefiniert sind folgende Schwellwerte und ihre Zuordnung zu den JQ6500-Funktionen:

```
// ADKey1 400ms 470 Ohm
#define SOUND_JQ6500_ADKEY1 29 // 1
#define SOUND_JQ6500_ADKEY2 40 // 2
#define SOUND_JQ6500_ADKEY3 50 // 3
#define SOUND_JQ6500_ADKEY4 60 // 4
#define SOUND_JQ6500_ADKEY5 80 // 5
#define SOUND_JQ6500_ADKEY6 162 // Prev/Vol-
#define SOUND_JQ6500_ADKEY7 231 // Next/Vol+
```

- Diese Tabelle steckt in der Datei MobaLedLib.h. Die Standardwerte passen für mein Modul.
- Man kann die Soundtest auch direkt im Program Generator über den LED-Farbtest durchführen. Auch lassen sich die Schwellwerte im Excel definieren (Expertenmodus). Eine Anpassung ist nur bei exotischen WS2811 Modulen notwendig!

Folgende Kommandos im seriellen Monitor sind verwendbar, um die LEDs anzusteuern:

- **L10 => LED Nr**
- L+ => Next LED
- L- => Previous LED
- **ENTER => Repeat last input (Useful with +, -, T or impulse)**
- X => Exit test mode
-
- h45 => Hue = 45
- v100 => Value = 100
- H20 => Hue = 20
- H+4 => Hue+=4
- H- => Hue--
- H+ => Hue++
-
- R30 => Red LED value
- G+ => Increment Green value
- B-3 => Decrement green value by 3
-
- **P50 => ImPulse with a value of 50 (Sound module tests)**
- **i100 => Impulse length [2ms] i100 = 200 ms**
- M2 => Impulse mask 1 = Red, 2 = Green, 4 = Blue, 7 = white
-
- A => All LEDs on (255)
- C => All LEDs off
-
- e3 => Enable input 3
- d3 => Disable input 3
- t3 => Toggle input 3

Das Testprogramm eignet sich auch gut für die Einstellung von LED-Farben.

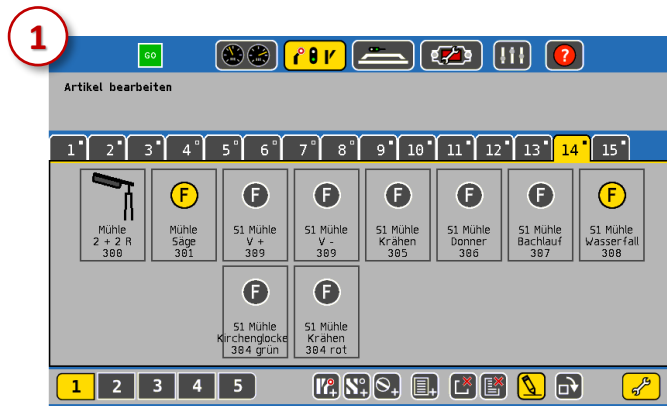
MobaLedLib – JQ6500 Endlosschleife Soundfiles über DCC

- Wenn man ein Soundfile endlos nacheinander abspielen möchte, muss man 3 Zeilen im Excel eingeben.
- Die erste Zeile aktiviert das Soundfile erstmalig bzw. einmalig beim Senden des DCC-Befehls (hier Adresse „308“). Macht man das nicht, würde der Sound erst nach Ablauf der Mindestzeit starten.
- Die zweite Zeile aktiviert die Random-Funktion. Diese setzt hier alle 20,3 Sekunden eine Variable „Sound-Loop“ für 1 Sekunde. Die beiden Wertepaare Min/Max sind identisch.
- Die Variable wiederum triggert dann in der 3. Zeile das erneute Abspielen des Soundfiles.
- Zwischen 2 Sound-Durchgängen ist eine kurze Pause für ein paar ms, was im Anlagenbetrieb nicht auffallen dürfte.

Aktiv	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte	Start LedNr	LEDs	InCnt	Loc InCh	LED Kanal
✓	VP-1	308	AnAus	0	Loop aktivieren für Sound 4, erstmalig abspielen			Sound_JQ6500_Seq4(#LED, #InCh)	3	^ C1-1	1	0	0
✓	VP-1	308	AnAus	0	Loop aktivieren für Sound 4			Random(Sound_Loop, #InCh, RM_NORMAL, 20.3 Sek, 20.3)			1	0	
✓	VP-1	Sound_Loop			Loop aktivieren für Sound 4			Sound_JQ6500_Seq4(#LED, #InCh)	3	^ C1-1	1	0	0

MobaLedLib – Soundmodul JQ6500 über DCC mit der ESU-ECoS ansteuern

- Für jede DCC-Adresse muss ein Schaltartikel in der ECOS-Zentrale eingerichtet werden ①.
- Die MobaLedLib kann DCC-Befehle auf zwei Arten schalten:
 - Ein-/Ausschalter:** wird z.B. für die Hausbeleuchtung oder die Steuerung eines Motors verwendet. Hierfür ist in der ECOS die Tastenfunktion „Umschalten“ zu markieren ②. Das Gleisbildsymbol kann dauerhaft ein- und ausgeschaltet werden.
 - Taster:** Pro DCC-Adresse können zwei Schaltvorgänge ausgelöst werden (z.B. rot/grün oder Abzweig/Gerade). Dies ist insbesondere für Sounds sinnvoll. Hier ist die Tastenfunktion „Impuls“ ② zu wählen und eine Farbe (Rot oder Grün) ③. Eine Schaltdauer ist hier nicht anwendbar; die Funktion wird solange ausgelöst, wie man auf das Symbol drückt.
 - Im Programm Generator kann dann eine DCC-Adresse zwei Aktionen/Sounds auslösen (Typ-Spalte) ④.



Aktiv	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte
✓	VP-1	304	Grün		Soundmodul Mühle 1 Kirchenglocken	1		Sound_JQ6500_Seq1(#LED, #InCh)
✓	VP-1	304	Rot		Soundmodul Mühle 2 Krähen	1		Sound_JQ6500_Seq2(#LED, #InCh)
✓	VP-1	305	Grün		Soundmodul Mühle 3 Donner	1		Sound_JQ6500_Seq3(#LED, #InCh)
✓	VP-1	305	Rot		Soundmodul Mühle 4 Bachlauf	1		Sound_JQ6500_Seq4(#LED, #InCh)
✓	VP-1	306	Grün		Soundmodul Mühle 5 Wasserfall	1		Sound_JQ6500_Seq5(#LED, #InCh)
✓	VP-1	307	Grün		Soundmodul Mühle V+	1		Sound_JQ6500_DecVol1(#LED, #InCh, 2)
✓	VP-1	307	Rot		Soundmodul Mühle V-	1		Sound_JQ6500_IncVol1(#LED, #InCh, 2)

