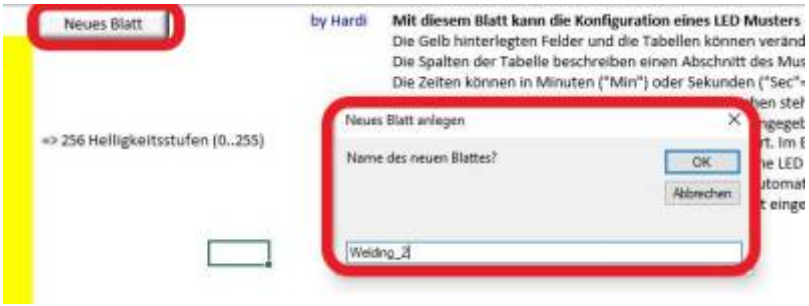
Schweißlicht mit Sound

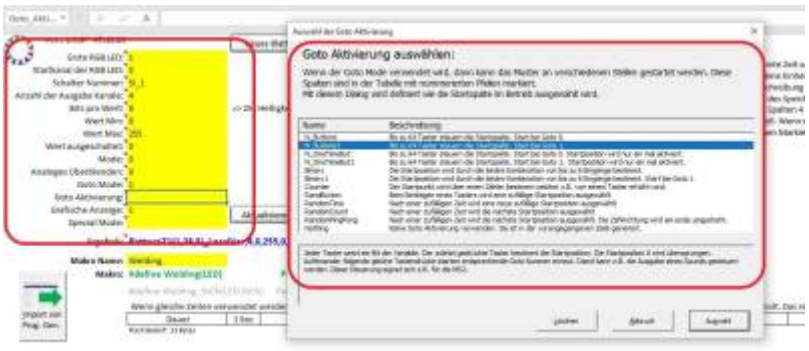
https://vimeo.com/466220989

Mit diesem Beispiel soll die Erzeugung eines schweißlichtähnlichen Lichteffekts mit einer RGB-LED und das zeitgleiche Abspielen der entsprechenden MP3-Sounddatei gezeigt werden. Der Sound liegt als MP3-Datei mit einer Dauer von ca. 3 Sekunden auf ein JQ6500 Modul vor. Details zum JQ6500 Modul im WIKI Link sound_jq6500_500de. Entsprechend wird das Schweißlicht ebenfalls eine Dauer von ca. 3 Sekunden haben. Längere Sounddateien können über entsprechende MP3-Bearbeitungsprogramme gekürzt und an eigene Vorstellungen angepasst werden. Harte Schnitte sollten im Fall des Schweißgeräuschs kein Problem sein.

Zunächst den Pattern_Configurator öffnen und ein neues Blatt anlegen, die Einstellungen nicht übernehmen und einen Namen nach eigener Wahl vergeben, hier Welding_2.



In dem neuen Blatt sind im gelb unterlegten Teil die Einstellungen wie im folgenden Bild gezeigt vorzunehmen.

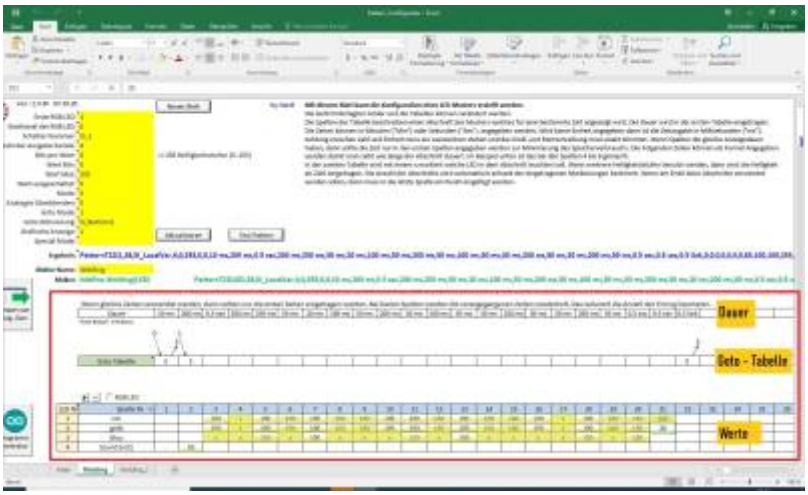


Anmerkungen dazu:

- Ausgabekanäle 4: 3 x für die RGB-LED, der vierte für die Ansteuerung des Sound-Moduls. Beim JQ6500-Modul wird der rote Kanal für den Sound gebraucht, gelb und blau sind herausgeführt und können für LEDs, natürlich auch weiße, genutzt werden. Das wird in diesem Beispiel im ProgGen auch so genutzt.
- Bits pro Wert 8: mit der 8 Bit Auflösung lassen sich die 256 Helligkeitsstufen darstellen. Für ein Schweißlicht mit harten Übergängen und wenigen Helligkeitsstufen würde eine geringere Auflösung zur Darstellung völlig ausreichen. Mit 4 Bit ergibt sich nach meiner Einschätzung keine wesentlicher Unterschied im erzeugten LED-Licht. Damit ließen sich ein paar Byte kostbarer Speicher einsparen. Allerdings brauchen wir für die Ansteuerung des Sound-Modules einen exakten Wert, der sich evtl. mit einer geringeren Auflösung nicht erzeugen lässt.
- Analoges Überblenden: ist für ein Schweißlicht mit kurzen Lichtimpulsen nicht notwendig.

- Goto Aktivierung: definiert die Methode zum Einlesen der Eingänge im GOTO-Modus. Durch einen Doppelklick auf das gelb unterlegte Feld öffnet sich das im Bild gezeigte Fenster. Wir wählen N_Buttons1 aus, da wir mit der Aktion bei „1“ starten wollen also erst bei der Betätigung eines Tasters oder der Ausführung eines Befehls aus einem Programm heraus und nicht bereits bei„0“ ohne eine Eingabe.
- Grafische Anzeige: eine „1“ schaltet die grafische Anzeige in den Tabellen ein und erzeugt die Goto-Tabelle.

Nun können wir die drei rot umrandeten Tabellen befüllen.



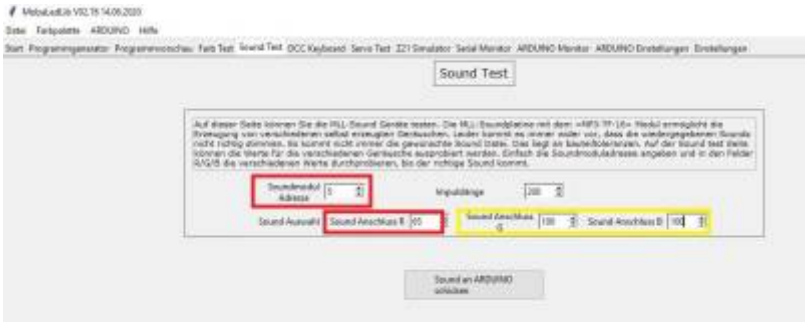
Die von mir eingetragenen Werte für die Zeiten und die Werte sind nur ein Beispiel und können natürlich nach eigenen Vorstellungen fast beliebig angepasst werden.

- In der untersten Tabelle für die Werteeingabe muss die Spalte Nr 1 leer bleiben, in der entsprechenden Spalte der Goto-Tabelle tragen wir ein „E“ als „GoEnd-Anweisung“ ein, die Funktion wartet auf ein Ausführungskommando, der logische Zustand ist „0“. Die Dauer ist unbedeutend daher „10 ms“. Bitte beachten: zwischen Zahlenwert und Einheit muss ein Leerzeichen eingegeben werden.
- In der zweiten Spalte der Goto-Tabelle geben wir ein „S“ als Startspalte ein. Der Einsprung in der Programmausführung erfolgt mit der logischen „1“ = Tastendruck, wie grafisch dargestellt.
- In der vierten Spalte der Wertetabelle habe ich den Wert 65 eingegeben, da bei meinem Sound-Modul die entsprechende MP3-Datei über diesen Wert sicher aufgerufen wird. Andere Werte können als Anhalt der folgenden Tabelle entnommen werden.

Anhaltswerte für den **JQ6500**

- SOUND 5 Wert 37
- SOUND 4 Wert 49
- SOUND 3 Wert 70
- SOUND 2 Wert 134
- SOUND 1 Wert 255

Die Werte können wegen der Bauteiltoleranzen für jedes Sound-Modul unterschiedlich ausfallen und müssen evtl. durch Tests mit der Sound Test-Funktion im LED-Farbtest des Programm_Generators ermittelt werden.

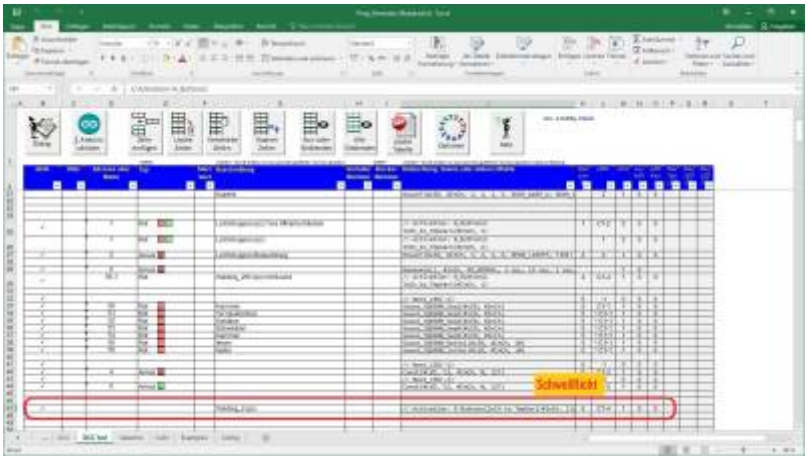


Die Sound-Modul-Adresse einstellen, bei mir die „5“, die Impulsdauer kann bei 200 ms bleiben. Der Sound Anschluss ist bei JQ6500 der rote Kanal, folglich hier den Wert einstellen und zum Arduino schicken. Ich konnte das Schweißgeräusch im Wertebereich 61 bis 71 aufrufen und „65“ als gewählt. Die Anschlüsse „G“elb und „B“lau haben beim JQ6500 keine Bedeutung für den Sound. Falls am Sound-Modul LEDs angeschlossen sind, können hier zur Kontrolle Helligkeitswerte eingestellt werden. Die beiden angeschlossenen LEDs leuchten dann dem Wert entsprechend.

- Die Dauer von 200 ms habe ich in dieser Spalte eingestellt, damit das Sound-Modul den Befehl sicher erkennt und reagiert, wie im vorigen Schritt bereits getestet. Der Sound wird nur einmal ausgelöst und läuft dann über 3 Sekunden ab. Ein Abbruch wäre nur über den Aufruf eines anderen Sounds bzw. eines „leeren“ Sounds möglich.
- In den folgenden Spalten sind Werte für die Ansteuerung der RGB-LED eingetragen. Die Werte können zwischen 0 und 255 liegen, x = Maximalwert. Zunächst ist das Licht etwas bläulich, zum Ende des Schweißvorgangs ist das rötliche Nachglühen zu sehen. Der Umsetzung eigener Vorstellungen und Wünsche steht hier (fast) nichts im Wege.
- Am Ende muss in die Goto-Spalte der Wert „E“ für „Ende“ eingegeben werden.
- Sound und Licht können leicht über die Verlängerung der Brenndauer der LED synchronisiert werden. Hier muss man etwas experimentieren.

Nun können wir die Konfiguration mit einem Klick auf die entsprechende Taste an den Programm_Generator schicken.

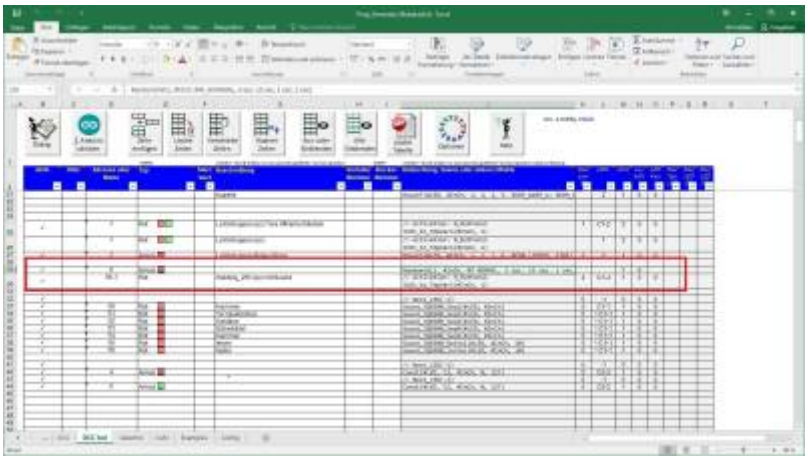
Wir wollen jetzt das Programm nicht direkt zum Arduino schicken und auch nicht direkt zurück zum Pattern_Configurator zurückkehren. Also wie im Text aufgefordert nur eine Zeile im ProgGen anklicken, in die das Muster geschrieben werden soll. Im nächsten Fenster übernehmen wir den vorgegebenen Wert „0“, Standard LEDs. Nach der Bestätigung mit „OK“ sollte das Ergebnis wie in Zeile 47 aussehen:



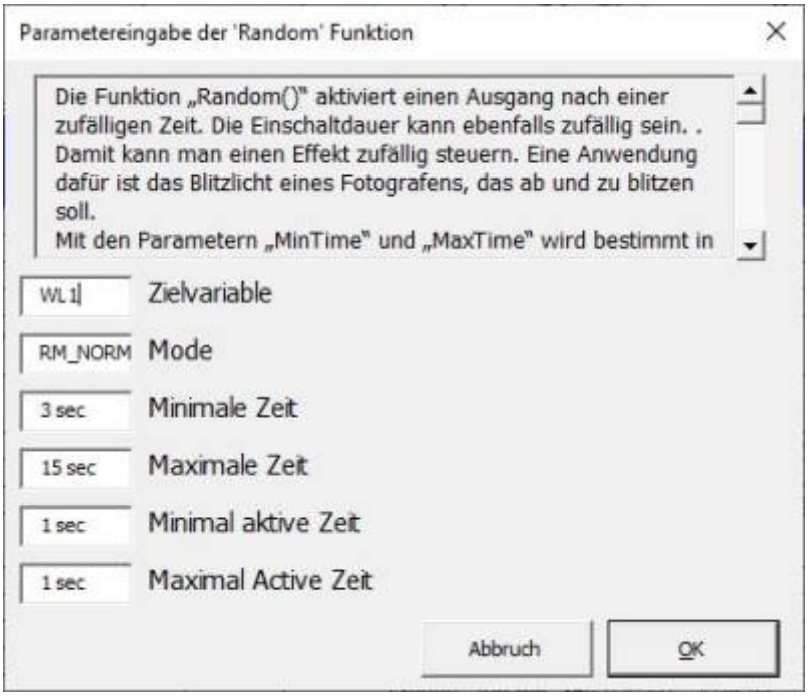
Nach der Eingabe einer Adresse und Auswahl eines Tasters können wir unser Ergebnis nun wie

gewohnt zum Arduino schicken und testen.

Mit jedem Aufruf wird die Funktion einmal ausgeführt. Ich habe die Funktion noch mit der Random-Funktion verknüpft, die über einen Ein/Aus-Befehl aktiviert wird und Schweißlicht und Geräusch in unregelmäßigen Abständen aufruft. Dazu muss der Expertenmodus aktiviert werden.



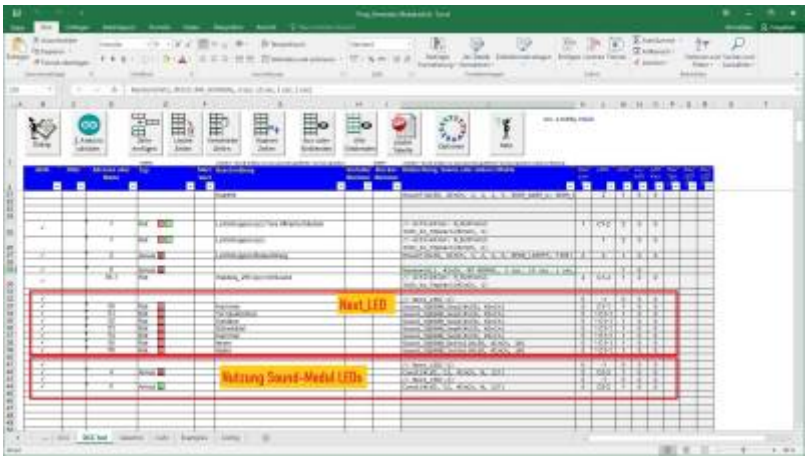
In der Random-Funktion habe ich folgende Werte eingegebe:



Über die Zielvariable WL1 wird unsere Schweißlichtfunktion aufgerufen.

In meinem Beispiel kann über die DCC-Befehle 50 bis 56 weiterhin das JQ6500-Modul wie gewohnt gesteuert werden. Dazu muss über die NEXT_LED-Funktion der Sprung zur nächsten LED in der Kette rückgängig gemacht werden – NEXT_LED **-1**!

Wie anfangs erwähnt können die beiden verbleibenden Ausgänge auf dem Sound-Modul für LEDs, zum Beispiel die Außenbeleuchtung des Lokschuppens, unter Verwendung von „NEXT_LED -1“ genutzt werden. In der Programmzeile 41 bis 44 ist das realisiert.



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

Permanent link:
https://wiki.mobaledlib.de/anleitungen/spezial/codevorlagen/schweisslicht

Last update: **2020/10/11 11:08**