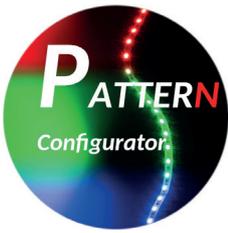


Heft 2



# Servosteuerung mit der MobaLedLib

Jochem Heinen

# Inhaltsverzeichnis

Vorwort.....	3
Meine Vorgehensweise.....	4
Voraussetzung.....	4
In eigener Sache.....	5
Bemerkungen zu meiner Vorgehensweise.....	6
Wie funktioniert die Servoplatine.....	6
Platine löten.....	7
Programmierung ATTiny.....	10
Einstellen der Servos.....	16
Servos im Prog_Generator anlegen.....	23
Ansteuerung der Tore in Traincontroller.....	28
Links.....	29
Danksagung.....	29

## Vorwort

In Folge 1 ([hier downloaden](#)) habe ich das Projekt „MobaLedLib“ beschrieben. Außerdem erzählte ich dort, wie ich dazu kam und erklärte auch die Installation.

Doch die MobaLedLib kann noch viel mehr als LEDs in den verschiedensten Farben ein- und auszuschalten. Wenn man das im Stummiforum verfolgt, dann erfährt man sehr schnell, dass man mit der MobaLedLib auch Servos ansteuern, Multiplex-Signale (z. B. von Viessmann) schalten, Schrittmotoren steuern, Sounds abspielen, Relaiskarten ansprechen und noch viel mehr kann.

Ich möchte mich in dieser Folge mit der Ansteuerung von Servos beschäftigen. Da ich absolut kein Elektronikfachmann bin mögen mir die fachlich Kundigen einige einfache Formulierungen verzeihen aber ich bitte darum, mir Fehler in meinem Vorgehen mitzuteilen damit ich dies korrigieren kann.

Jetzt zu meinem Projekt: aus der Vor-MobaLedLib-Ära steht auf meiner Modellbahn ein 3-ständiger Lokschuppen von Faller. Dieser ist weder beleuchtet noch öffnen bzw. schließen sich die Tore automatisch. Eigentlich stehen die Tore immer auf - und die Preiserlein holen sich bei kaltem Wetter eine kräftige Erkältung. Dies möchte ich mit einer Servoansteuerung der Tore ändern, allerdings sind mir die von Faller dafür erhältlichen Servos etwas zu kostspielig.

Da ich mich seit kurzem mit der MobaLedLib beschäftige möchte ich jetzt diese Servos mit selbiger ansteuern. Nachfolgend erkläre ich mein Vorgehen und zum Schluss funktioniert dies alles auf meiner Moba und ich bin mehr als zufrieden damit.

Viel Spass mit der MobaLedLib

*Jochem Heinen*

## Meine Vorgehensweise

Folgende Punkte muss man nacheinander durchführen um zu einem Ergebnis zu kommen:

1. Platine für Servoansteuerung bestellen (Wiki)
2. elektronische Teile für diese Platine bestellen (Reichelt)
3. Platine löten
4. ATTiny mit „Betriebssystem“ bespielen
5. Min.- und Max.-Position und Stellgeschwindigkeit der Servos einstellen
6. Im Programm\_Generator diese Servos anlegen und auf die Hauptplatine übertragen
7. Einbindung in Traincontroller

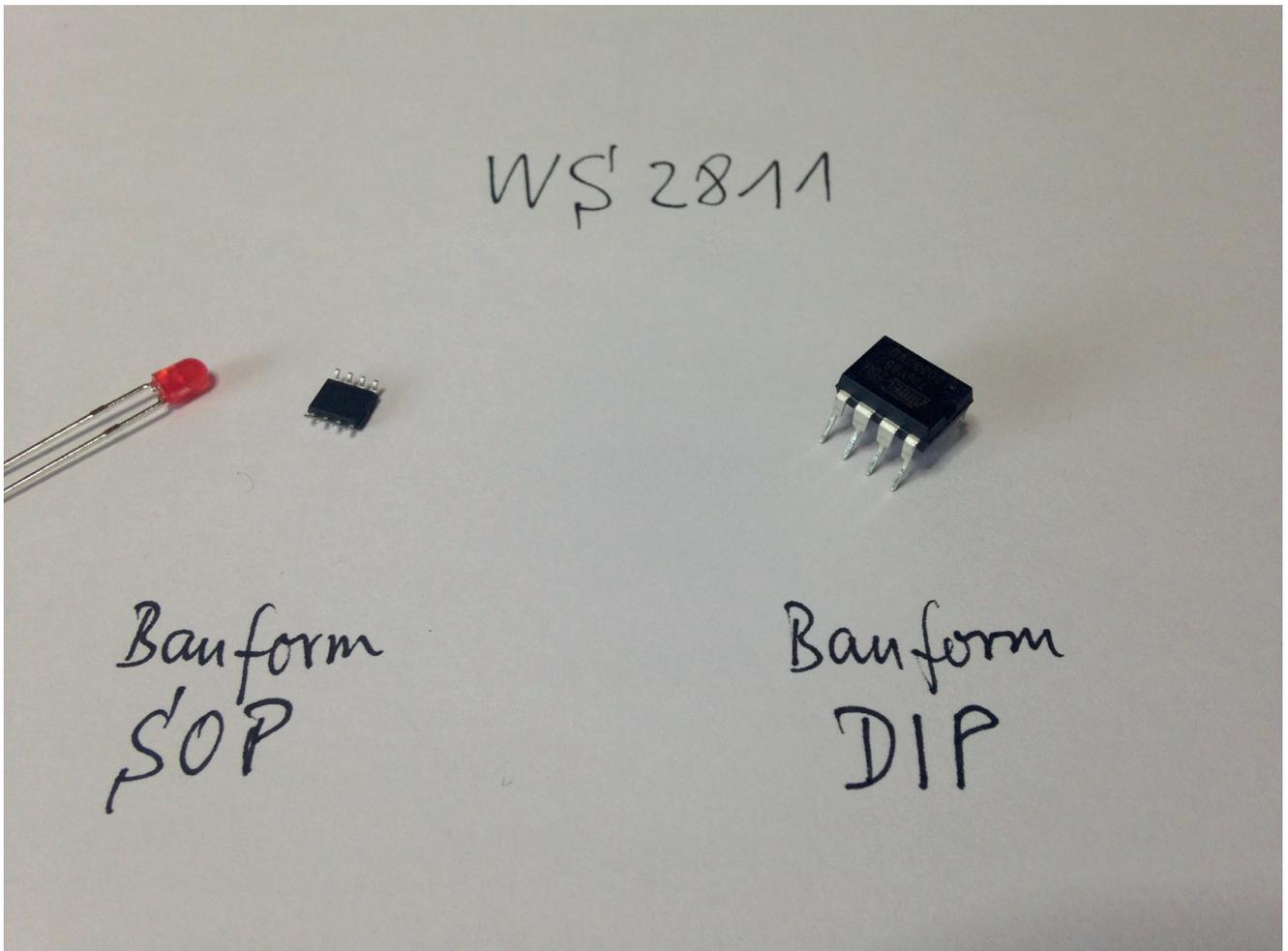
## Voraussetzung

Um die Servoplatine ansteuern zu können benötigt man natürlich die MobaLedLib-Hauptplatine, eventuell Verteilerplatinen und zur Programmierung des ATTiny die Programmierplatine und einen Arduino Uno. Alle weiteren Informationen hierzu findet man auf den Seiten des [MobaLedLib-Wiki](#). Hier werden auch die entsprechenden Warenkörbe für Reichelt angezeigt. Bitte bei der Übernahme der Artikel unbedingt prüfen, ob diese auch vorrätig sind. Eventuell - so habe ich das gemacht - einige der Artikel zur Sicherheit in größerer Stückzahl bestellen.

**ACHTUNG:** der von mir genutzte WS2811 in DIP-Bauform ist NICHT im Reichelt-Warenkorb enthalten. Soweit ich weiß bietet Reichelt diesen gar nicht an. Ich hatte meine bei Alf bestellt, der auch die MLL-Platinen verschickt (eMail: LedLib@yahoo.com).

## In eigener Sache

Den auf der Servoplatine aufzubringenden „WS2811“-IC kann man wahlweise als SOP- oder als DIP-Version auflöten. Da ich trotz Lupe und Brille den superkleinen WS2811 in SOP-Ausführung nicht mehr richtig aufgelötet bekomme habe ich mich für die DIP-Version entschieden. Beides funktioniert natürlich tadellos, man muss sich halt nur für eine von beiden Varianten entscheiden und diesen dann auch entsprechend bei Reichelt bestellen.



*Größenvergleich der beiden Bauformen eines WS2811. Zum Vergleich eine 3mm LED.*

## Bemerkungen zu meiner Vorgehensweise

„Viele Wege führen nach Rom“ sagt ein altes Sprichwort. Und so ist es auch mit der MobaLedLib: es gibt mehrere Wege zum Endergebnis.

### 1. Programmierung ATTiny:

Den ATTiny kann man entweder mit der „Programmierplatine“ oder mittels eines zusätzlichen Arduinos (also ohne Programmierplatine) programmieren. Ich habe mich für die Programmierplatine entschieden

### 2. Einstellung der Servos:

Den linken und den rechten Anschlag und die Stellgeschwindigkeit kann man entweder mittels des Programms „LED Farbtest“ einstellen oder man nutzt dazu - falls man diese bestückt hat - die Taster auf der Hauptplatine. Da ich aber die Hauptplatine nur in der „DCC Grundversion“ bestückt habe nutze ich zur Einstellung der Servos den LED-Farbtest.

## Wie funktioniert die Servoplatine

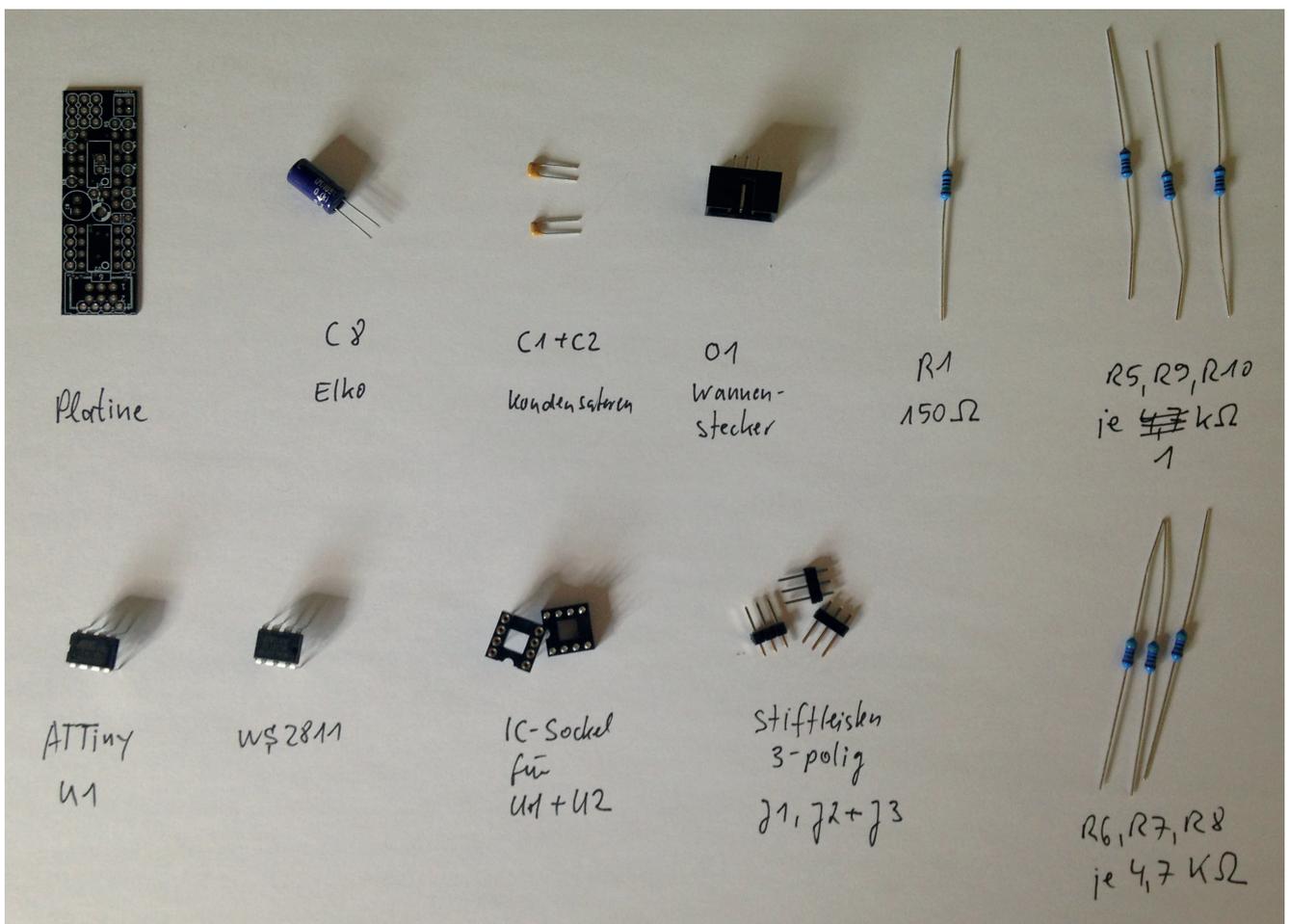
Ich kann dies nur laienhaft erklären und ich hoffe, dass ich damit einigermaßen richtig liege:

Die Hauptplatine der MobaLedLib erzeugt grundsätzlich nur Digitalsignale, mit denen die Helligkeit von LEDs geändert bzw. angezeigt werden kann. Auf der Servoplatine befindet sich ein „WS2811“, der diese Informationen von der Hauptplatine empfängt, in Helligkeitswerte pro RGB-Kanal umwandelt und an den ATTiny weiterreicht. Der ATTiny wiederum wandelt diese Helligkeitswerte in Stellimpulse für die Servos um und sagt dann dem Servo so etwas wie: „Ich habe für dich den Helligkeitswert 157 empfangen und somit stellst du jetzt die Position 140 Grad an.“ Da der WS2811 die Helligkeitswerte für die 3 Farbkanäle (Rot, Grün, Blau) empfängt kann der ATTiny damit 3 verschiedene Servos ansprechen.

Die Elektronikfachleute mögen mir meine Ausdrucksweise verzeihen.

## Platine löten

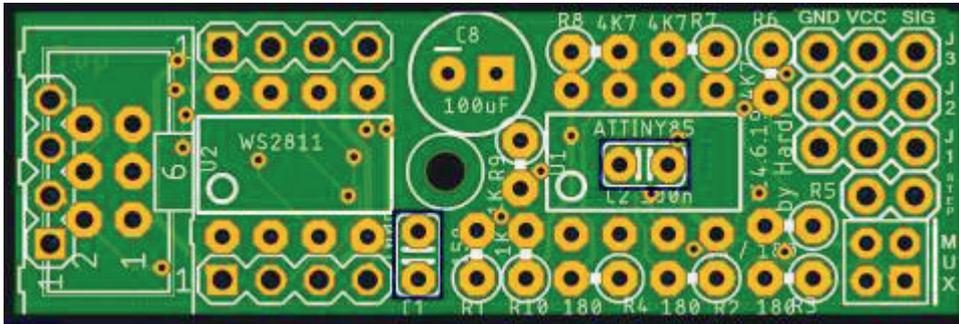
Hierzu gibt es nicht viel zu sagen. Es sind einige wenige Bauteile, die verlötet werden müssen. Wie bereits gesagt habe ich mich für die DIP-Variante des WS2811 entschieden. Tip: nach dem Verlöten jedes einzelnen Bauteils kontrolliere ich die Löstellen auf der Rückseite durch eine Lupe. So kann ich direkt feststellen, ob die Verlötung korrekt ist oder ob eventuell ein direkt daneben befindlicher Kontakt / Durchkontaktierung fälschlicherweise mit Lötzinn verbunden wurde. Außerdem habe ich mir angewöhnt, die verschiedenen Widerstände vor dem Einlöten nochmals mit einem Multimeter nachzumessen. Es ist mir tatsächlich schon einmal passiert, dass ich keinen 4,7 Kilo-Ohm-Widerstand sondern einen 47 Ohm Widerstand aus meiner Bastelkiste zurechtgelegt habe. Das sind Fehler, die später nicht mehr so leicht auffindbar sind!



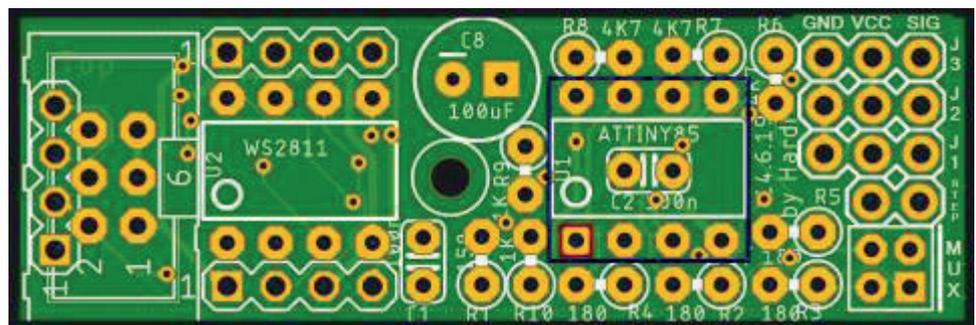
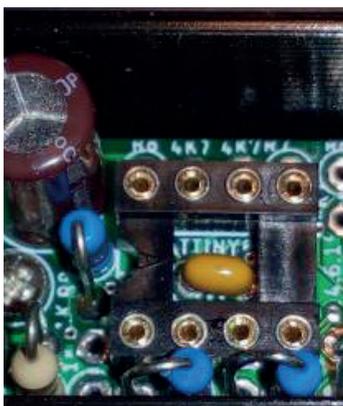
*Benötigte Teile für die Servoplatine*

Im MobaLedLib-Wiki wird die Bestückung der Platine wie folgt beschrieben:

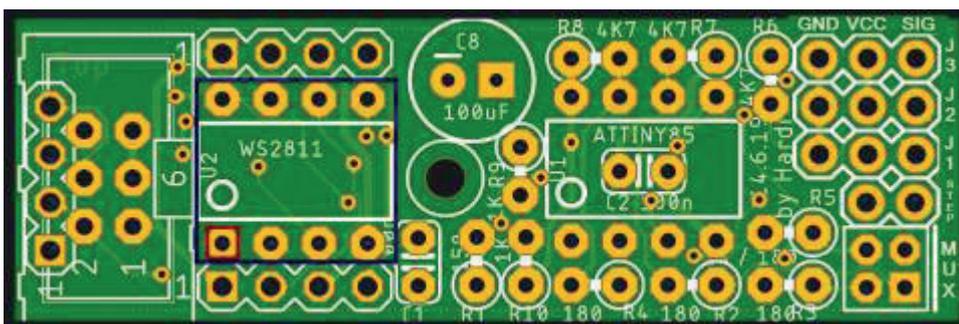
Den Anfang der Bestückung machen die beiden Keramikkondensator C1 und C2 mit jeweils 100nF.



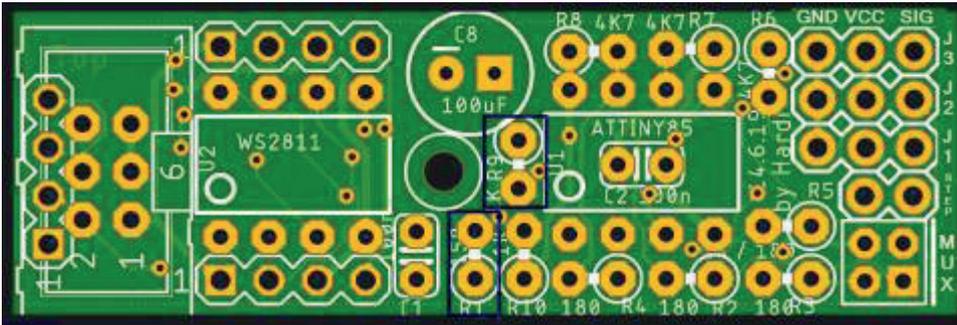
Als nächstes kommt der Sockel für U1 an die Reihe. Dieser wird über den Kondensator C2 gesteckt, sowie auf dem Bild zu sehen ist. Die Markierung am Sockel/IC ist auf dem Bild hervorgehoben.



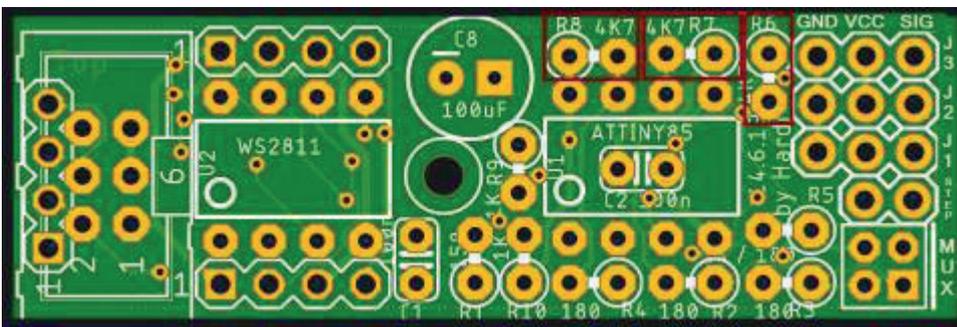
Nun folgt, wenn man sich für die DIP-Variante entschieden hat, das Einlöten von dem 8-Pin-Sockel für U2. Die Markierung am Sockel/IC ist auf dem Bild wieder hervorgehoben. Sollte man sich für die SMD-Variante entschieden haben, bleibt der Sockel leer. U3 wird auf der Unterseite der Platine aufgelötet. Dieser Vorgang wird im Wiki beschrieben.



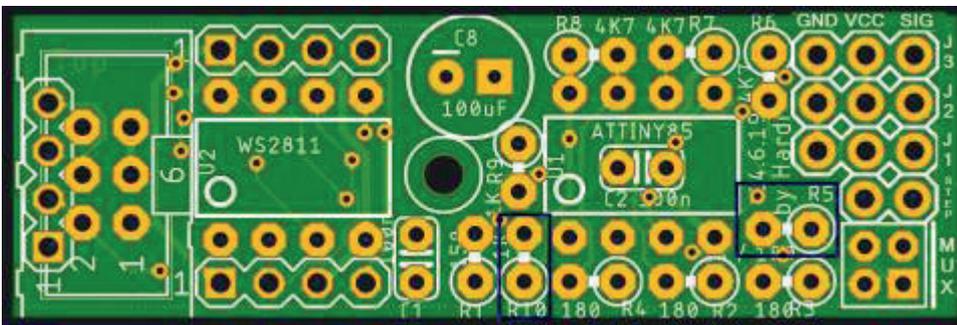
Als nächstes folgen nun die Widerstände R1 (150  $\Omega$ ) und R9 (1 K $\Omega$ ),



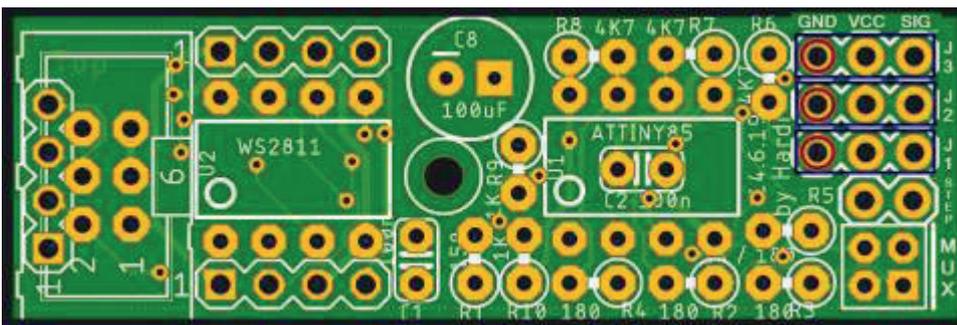
die Widerstände R6, R7 und R8 (jeweils 4,7 K $\Omega$ )



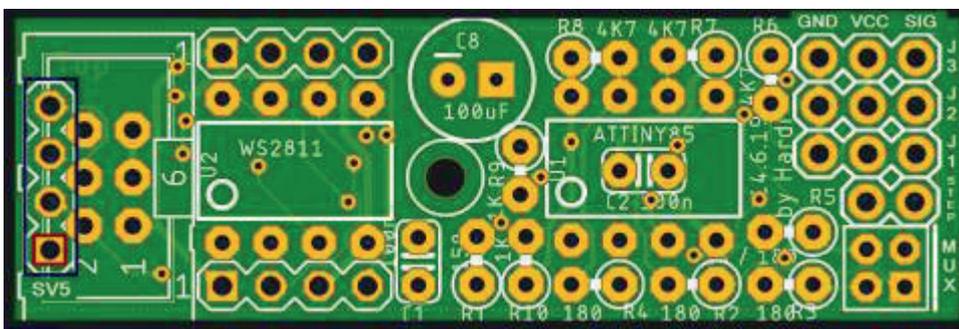
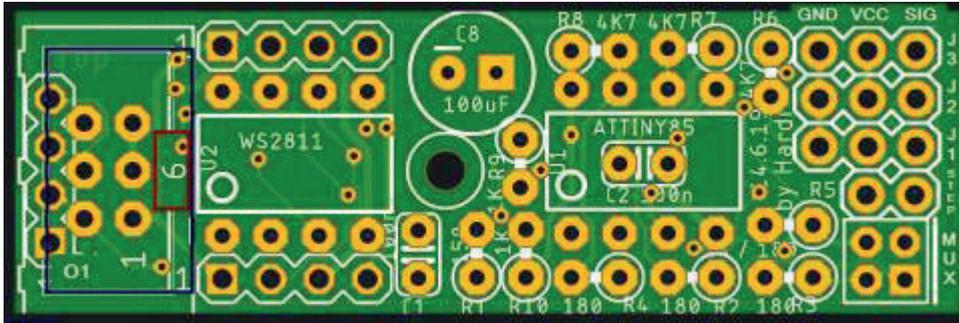
sowie die Widerstände R5 und R10 (jeweils 1 K $\Omega$ )



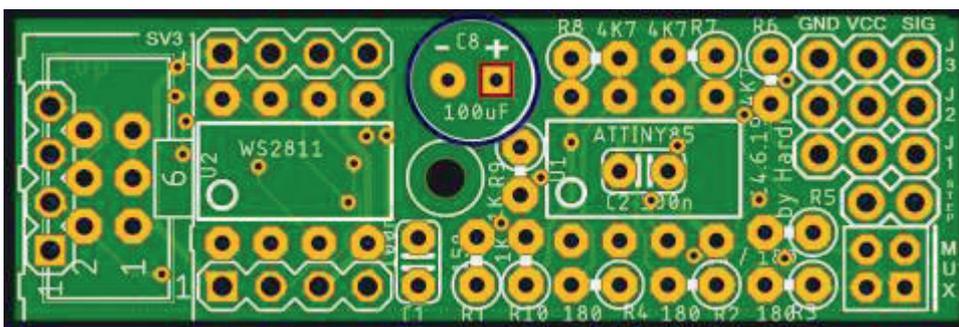
Die Anschlüsse für die drei Ausgänge J1 - J3 kommen als nächstes dran. Die Beschriftung auf der Platine ist leider fehlerhaft. Auf dem Bild ist die richtige Beschriftung der Pins erkennbar.



Je nachdem ob man den Wannenstecker (O1) oder die 4-polige Stiftleiste (SV5) benötigt, kann eine der beiden Varianten nun eingelötet werden. Wenn man mehrere Platinen miteinander verbinden möchte, bitte vorab die Rubrik „Signalbus und Anschluss“ (im Wiki) beachten.



Das letzte Bauteil das noch fehlt ist der Elektrolytkondensator C8. Bei diesem bitte auf die Polung achten. Der Minuspol ist am Kondensator und auf der Platine markiert.



## Lötjumper

Auf der Unterseite der Platine befinden sich sieben Lötjumper. Diese haben die folgenden Funktionen

Als letztes müssen auf der Rückseite der Platine noch folgende Lötjumper mittels Lötzinn verbunden werden:

**SERVO, SERVO1, SERVO2, SERVO3:** verbinden die notwendigen, zusätzlichen Datenleitungen zwischen dem WS2811 und dem Attiny85 und aktiviert die Servofunktionalität.

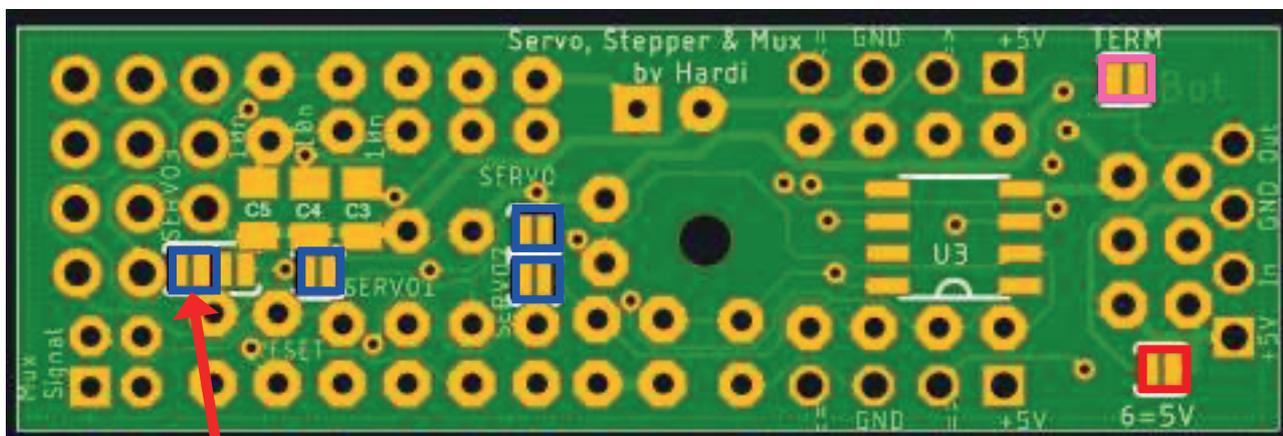
**6=5V:** verbindet die 5V-Leitung mit dem Pin6 vom Wannenstecker „O1“. Nicht verbinden wenn an dem Pin6 am Wannenstecker O1 mehr als 5V vorhanden sind.

**RESET:** bleibt bei der Nutzung als Servoplatine offen!

**TERM:** Wenn dies die letzte Platine in der Reihe ist oder die einzige Platine ist, muss dieser Lötjumper geschlossen werden, andernfalls wird die Kette unterbrochen.

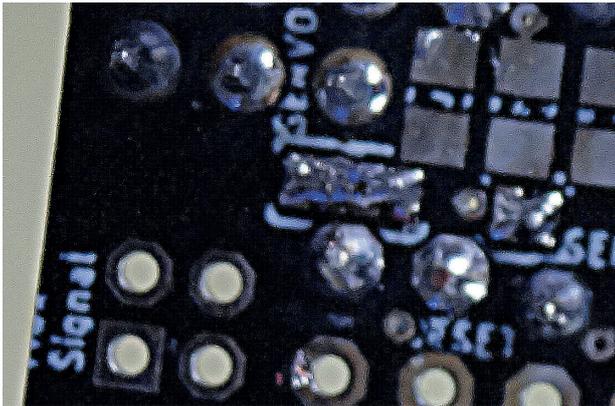
Mein Tip: da ich jede Server-Platine einzeln an Verteilerplatinen anschließen werde, muss ich bei jeder Server-Platine den Lötjumper „TERM“ mit Lötzinn schließen.

Die notwendigen Lötjumper für den Servo-Betrieb sind auf unterm Bild blau markiert. Die rote Markierung zeigt den Lötjumper für die Verbindung Pin6 und 5V (Pin1). Der Lötjumper oben rechts in rosa ist für die Abschlusstermierung, bei der letzten Platine oder bei Einzelplatinen.

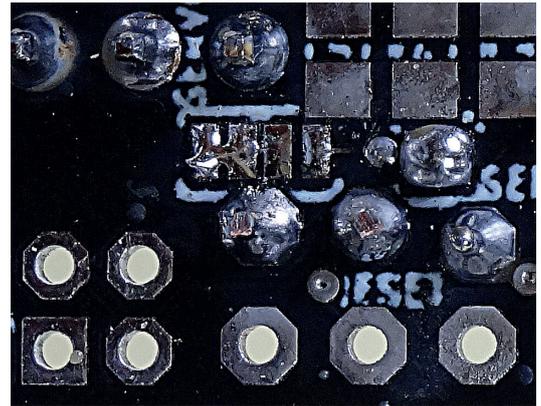


*Siehe Bermerkung auf nächster Seite*

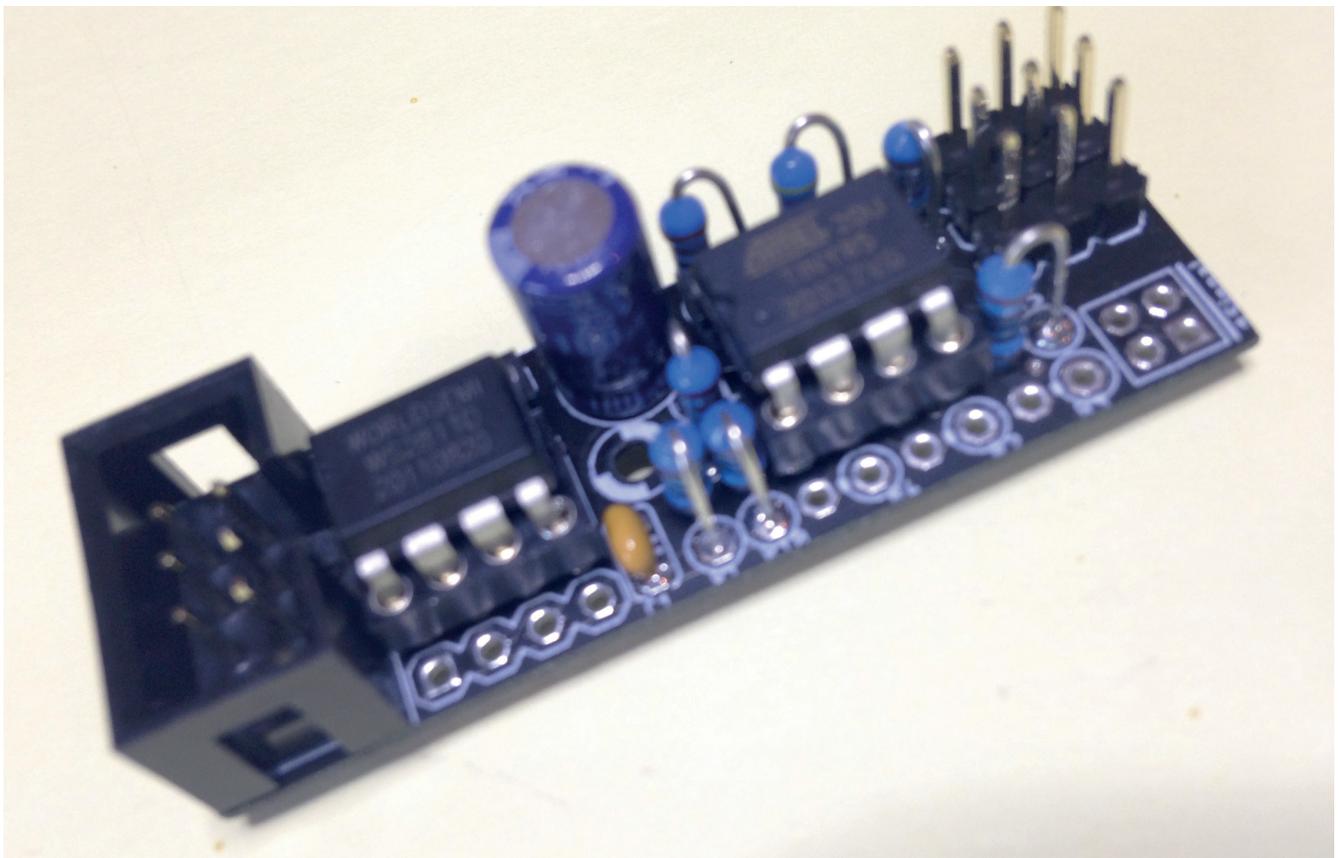
ACHTUNG: an der mit Pfeil markierten Stelle bitte darauf achten, dass nur die beiden linken Pads mit Lötzinn verbunden werden. Fälschlicherweise hatte ich zuerst alle 4 Pads verbunden, damit funktionierte aber das 3. Servo nicht. Nach Entlöten der 4 Pads und sorgfältigem überbrücken der beiden linken Pads funktioniert alles prima!!! :-)



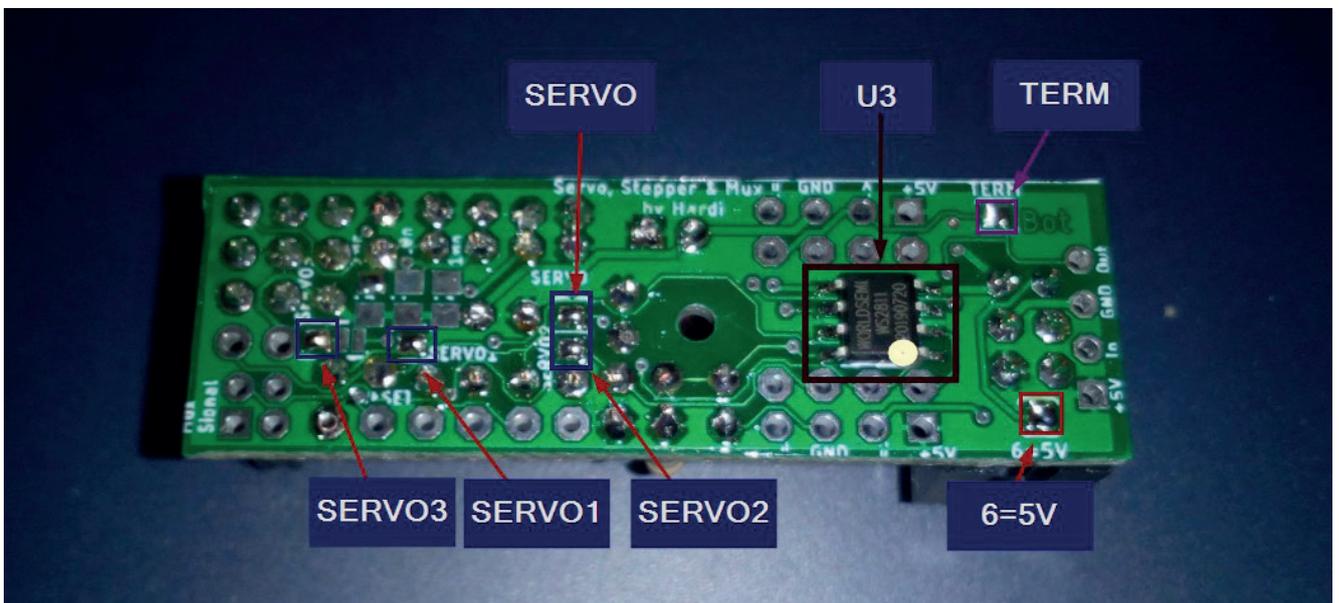
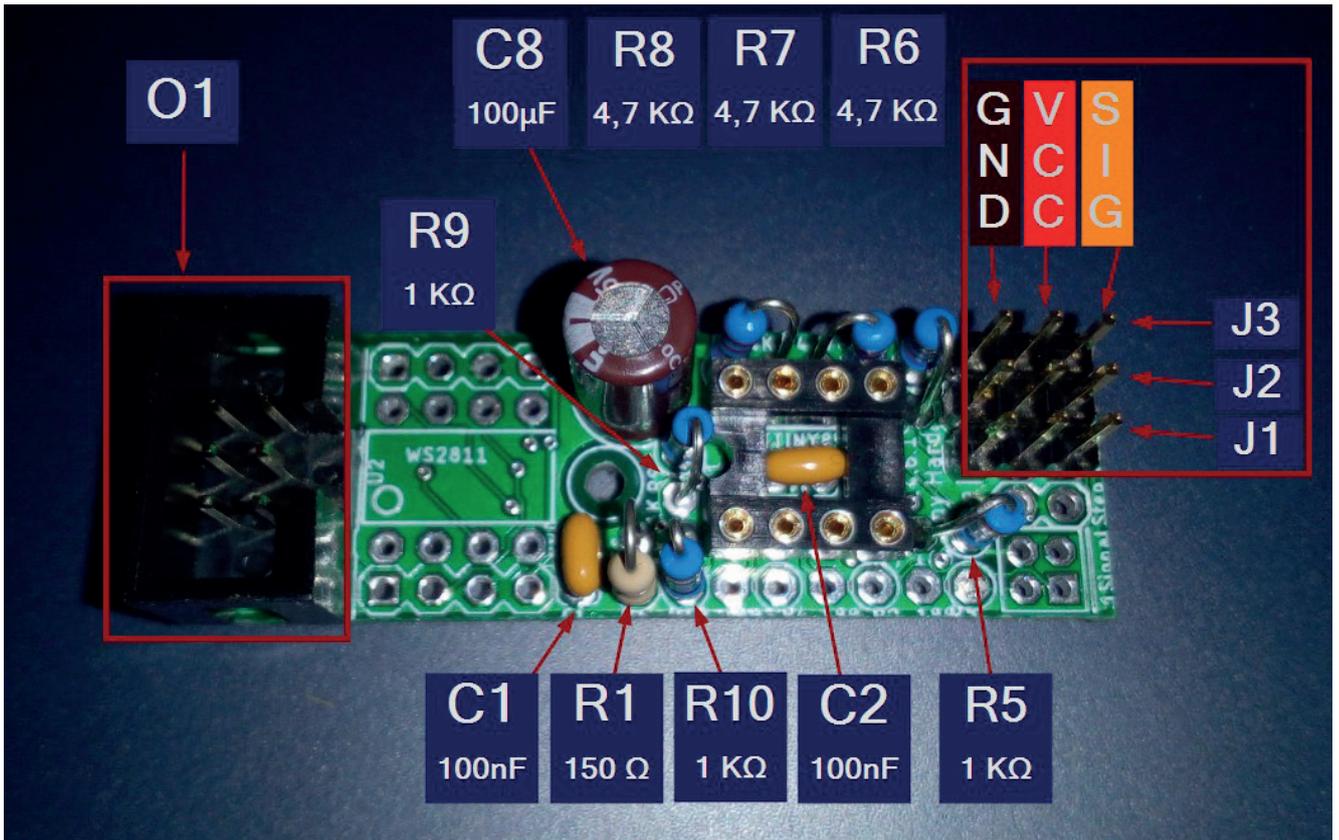
*FALSCH: 4 Pads verbunden*



*RICHTIG: nur die beiden linken Pads verbunden*



*Meine fertig bestückte Servoplatine*



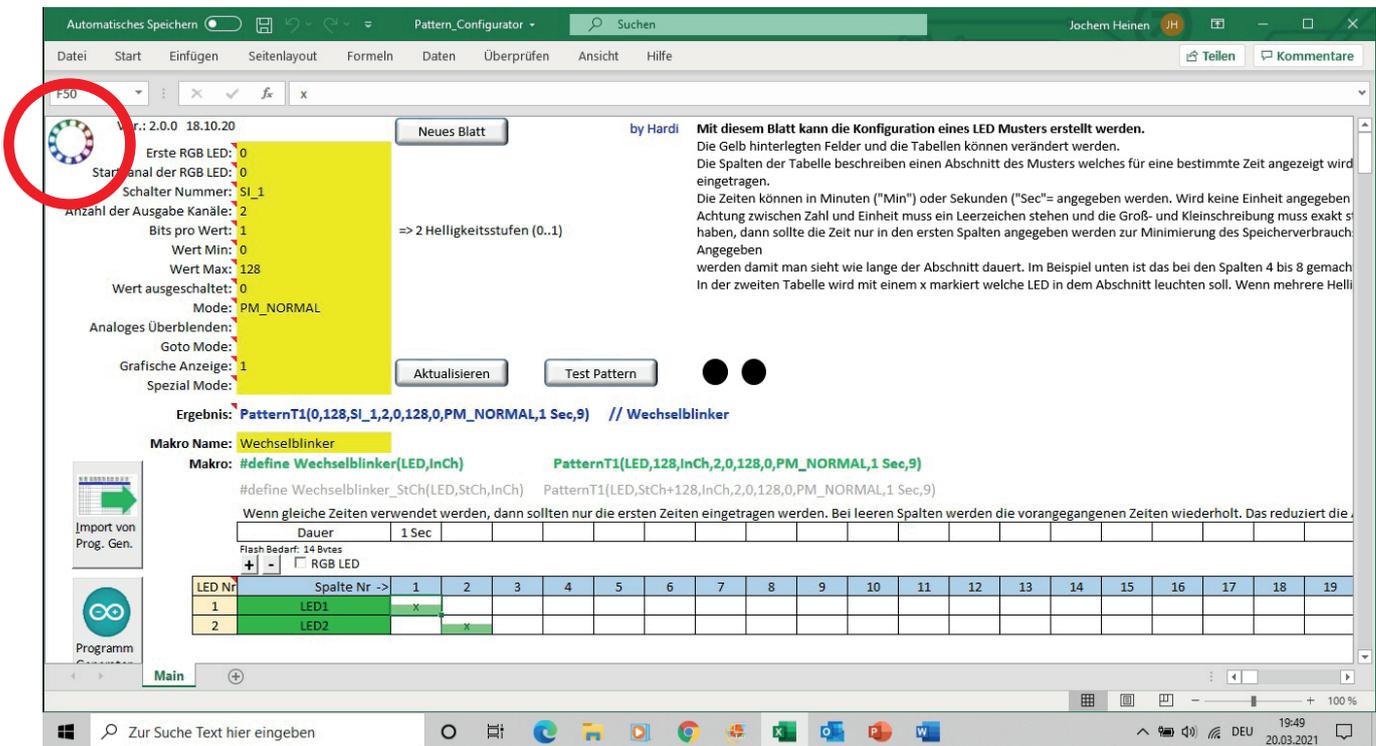
So sieht die fertige Platine (oben Vorderseite, unten Rückseite) im Wiki aus. Abweichend zu meiner Platine wurde hier der WS2811 (U3) als SOP-Bauform auf der Rückseite aufgelötet, so dass dieser auf der Vorderseite rechts neben dem Wannenstecker O1 entfällt.

# Programmierung ATTiny

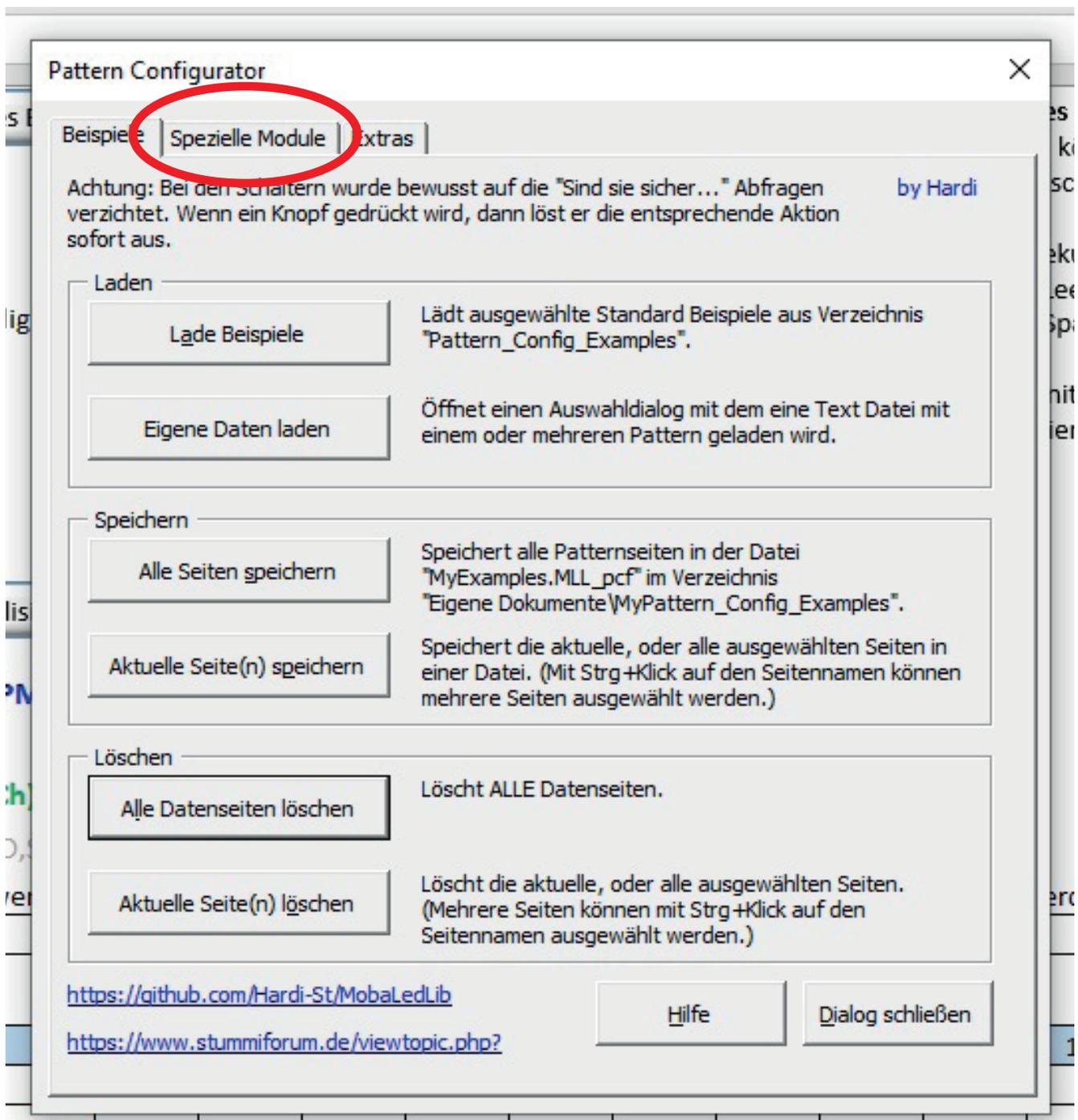
Als nächstes muss man dem ATTiny beibringen, was er überhaupt machen soll. Dafür muss dieser mit einem „Betriebssystem“ (also ähnlich wie bei einem Computer) programmiert werden. Ich habe dies mittels der „Programmier-Platine“ gemacht. Diese wird auf einen Arduino-Uno gesteckt und auf der Platine befindet sich ein IC-Sockel, in den der zu programmierende ATTiny gesteckt wird. Jetzt wird der Uno mittels USB-Kabel mit dem Computer verbunden und das Programm „MobaLedlib Pattern\_Configurator“ gestartet.

Anschließend führte ich folgende Schritte durch:

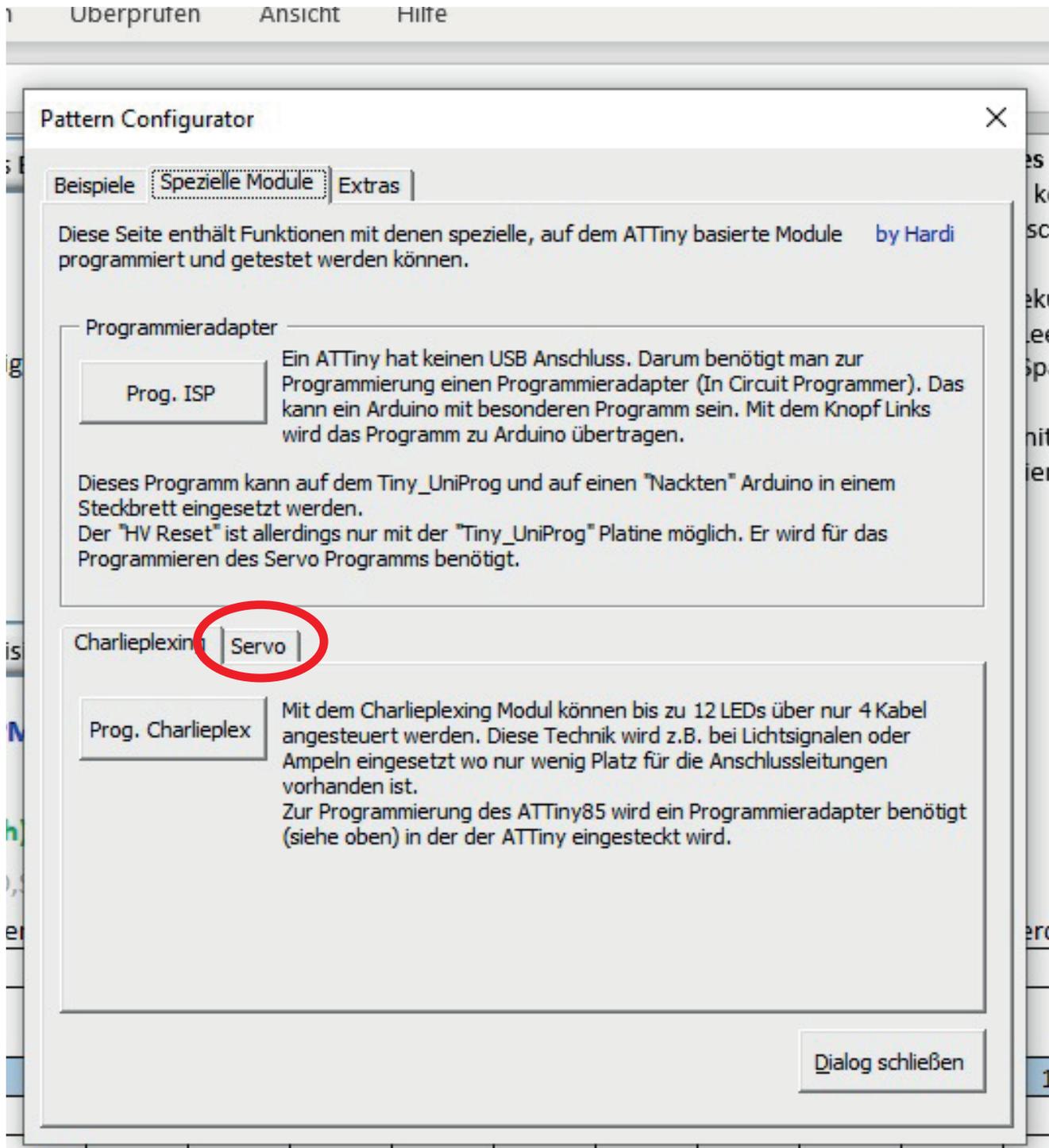
Klick auf den „Farbkreis“



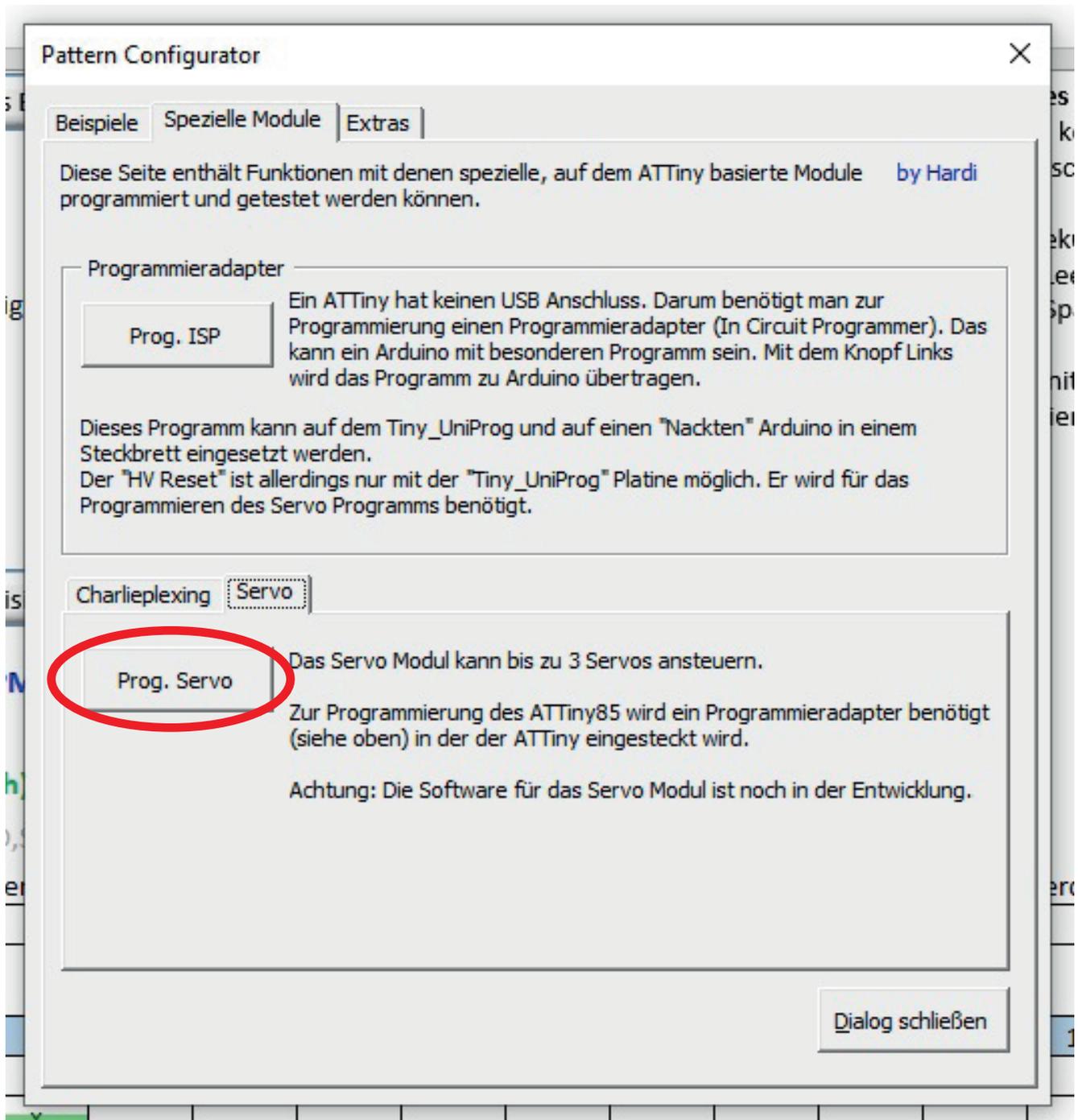
Es öffnet sich ein Fenster „Pattern Configurator“. Hier auf den Reiter „Spezielle Module“ klicken.



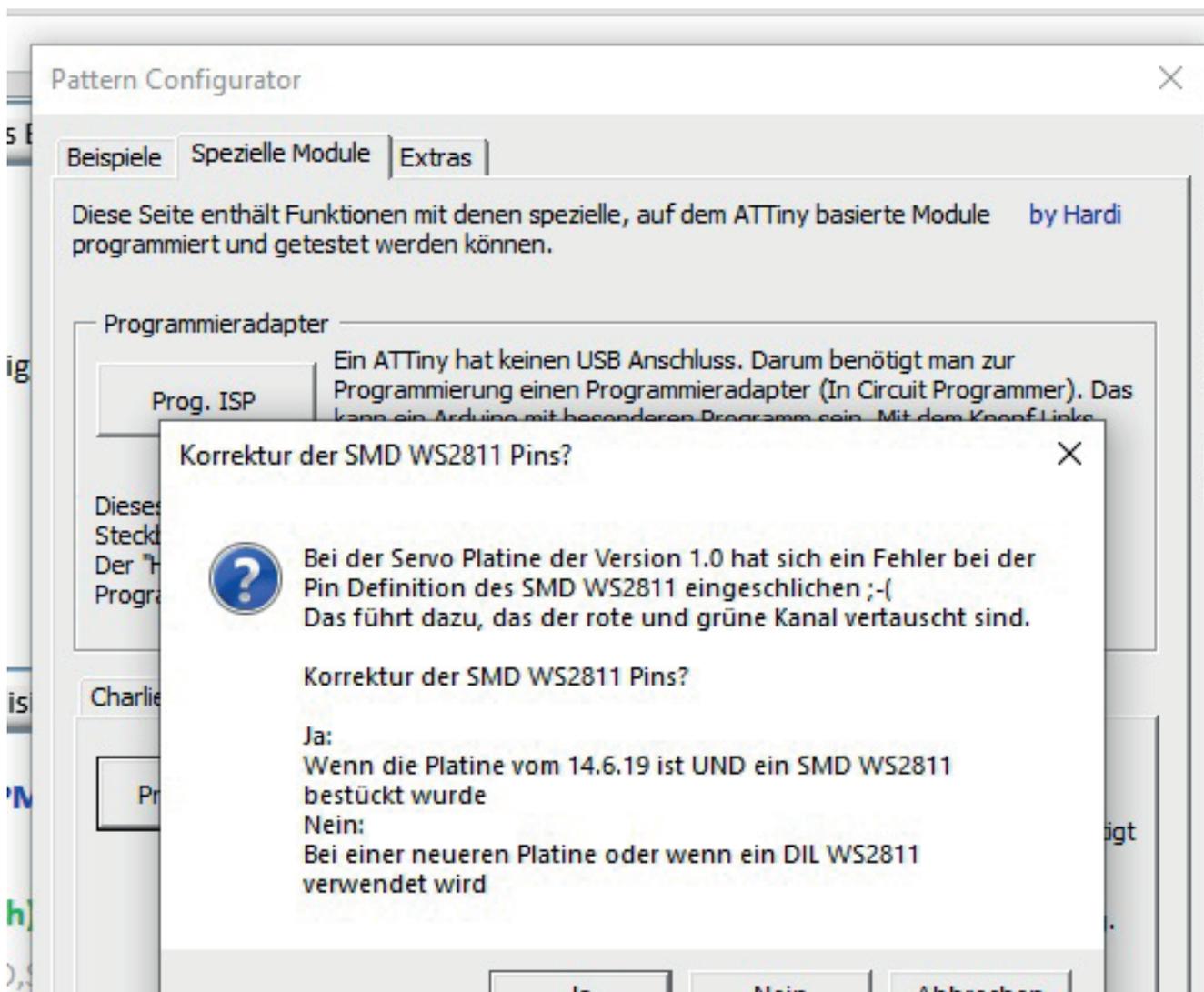
Im unteren Teil auf den Reiter „Servo“ klicken.



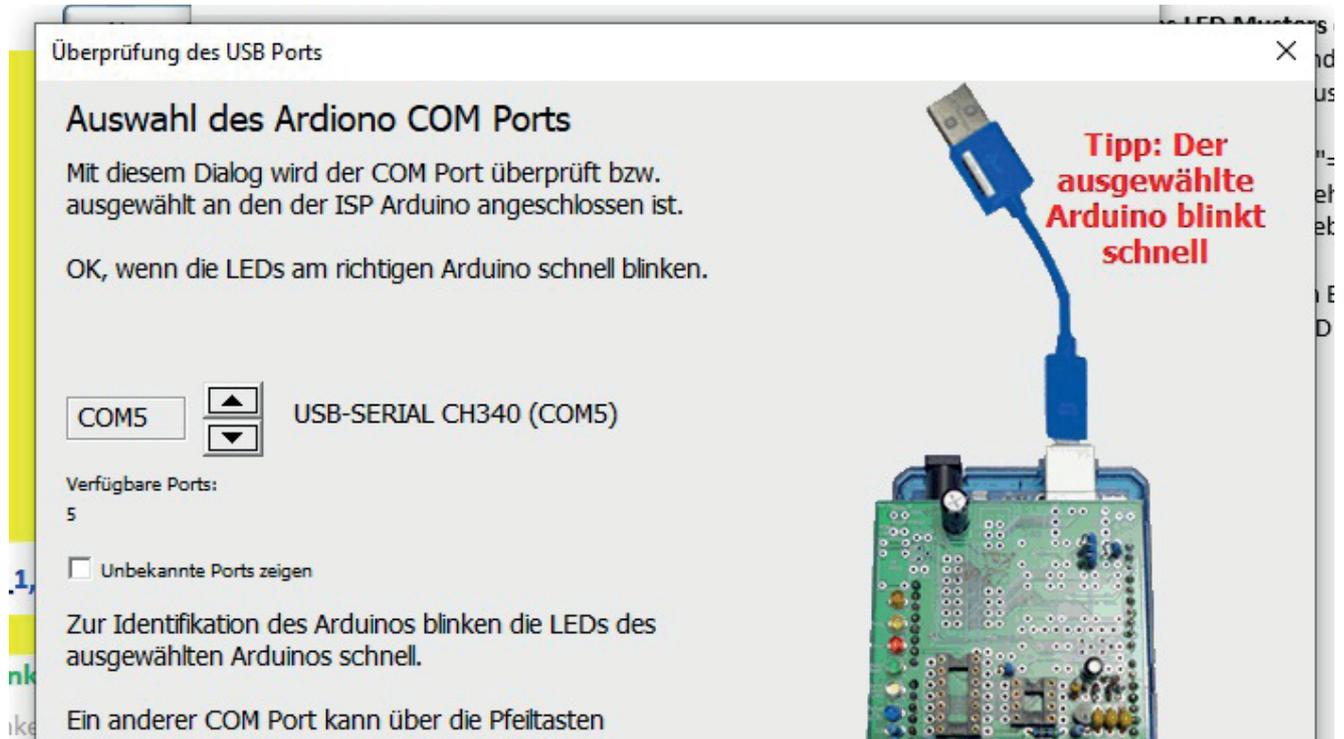
Dann auf die Schaltfläche „Prog. Servo“ klicken.



Als nächstes wird die Bauform des WS-2811 abgefragt. Da ich einen in der DIL-Version eingelötet habe muss ich hier auf „Nein“ klicken.



Wie wir das schon von der Programmierung der Hauptplatine kennen wird jetzt der COM-Port des Arduinos mit der Programmier-Platine gefragt. Entweder wird diese schon korrekt angezeigt oder man muss diese manuell auswählen. Hier dann einfach auf „Weiter“ klicken.

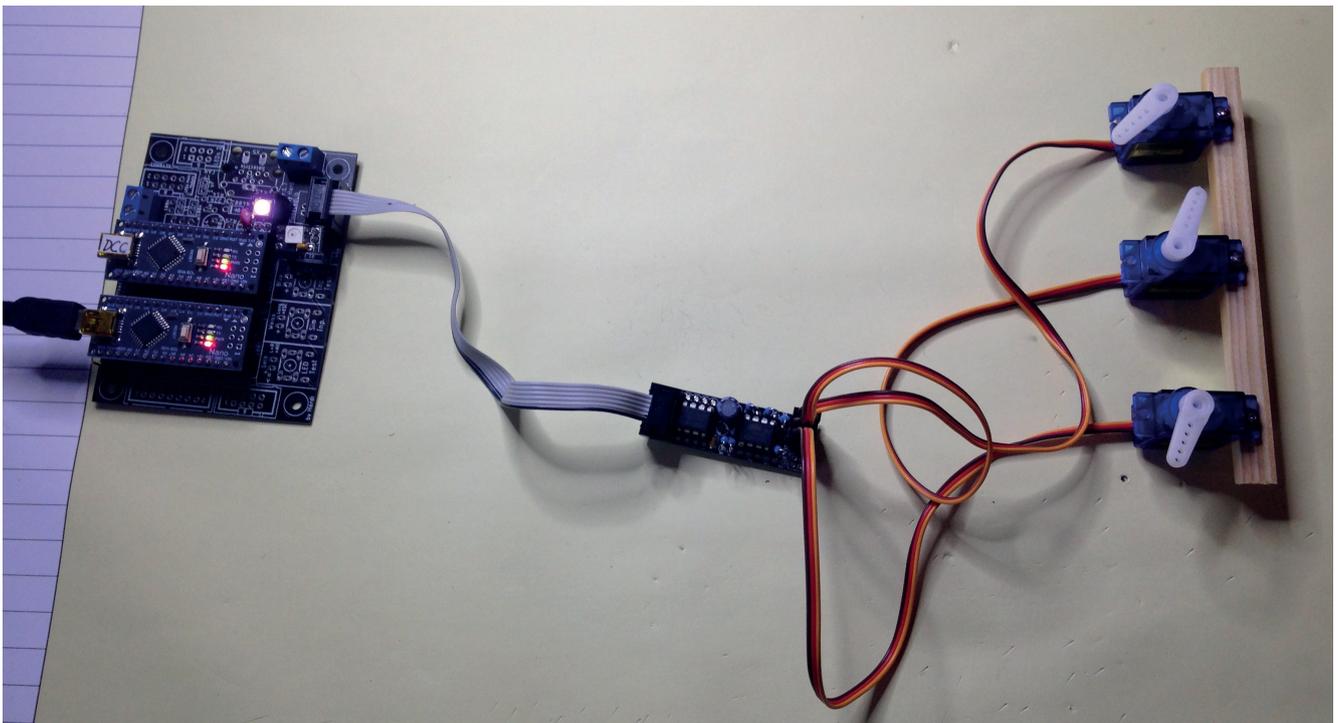


In Folge erscheinen jetzt mehrere Fenster, die die Programmierung des ATTiny anzeigen. Hierzu habe ich keine Screenshots gemacht. Nach kurzer Zeit sollte eine Fertigmeldung ohne Fehler angezeigt werden. Dann kann man den Pattern-Generator schließen. Der ATTiny ist jetzt für die Servo-Platine fertig programmiert.

## Einstellen der Servos

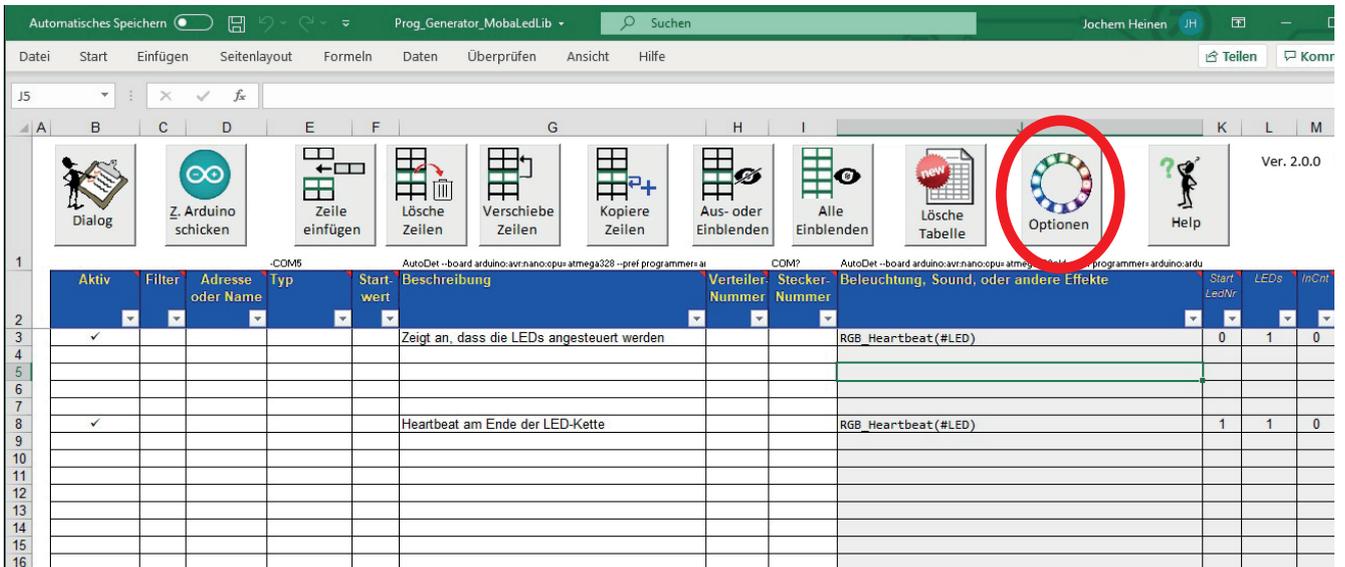
Jetzt kann der fertig programmierte ATTiny von der Programmier-Platine entfernt werden und in den Sockel auf der Servoplatine eingesteckt werden.

Dann wird ein Grundaufbau der MobaLedLib aufgebaut: Die Hauptplatine kommt mittels USB-Kabel an den Computer, in den Wannenstecker der LED-Kette wird das Verbindungskabel zur Servoplatine gesteckt. An die Servoplatine selber können bis zu 3 Servos angeschlossen werden.

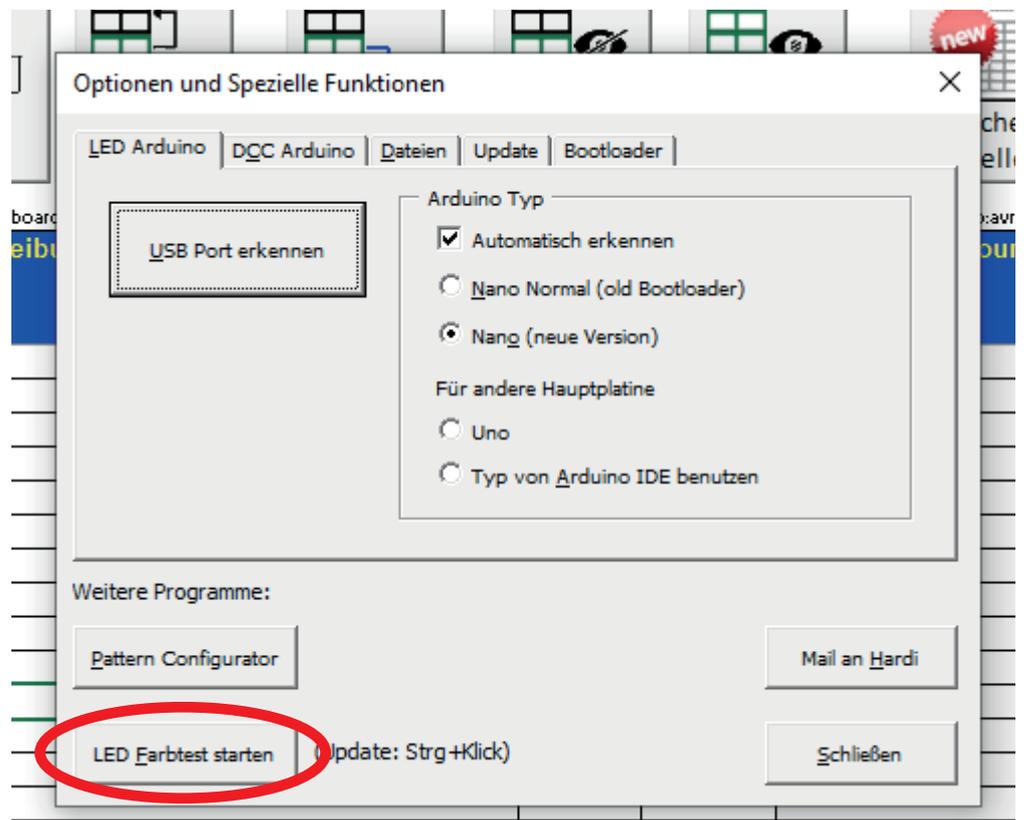


*Mein Testaufbau Hauptplatine - Servoplatine - Servos*

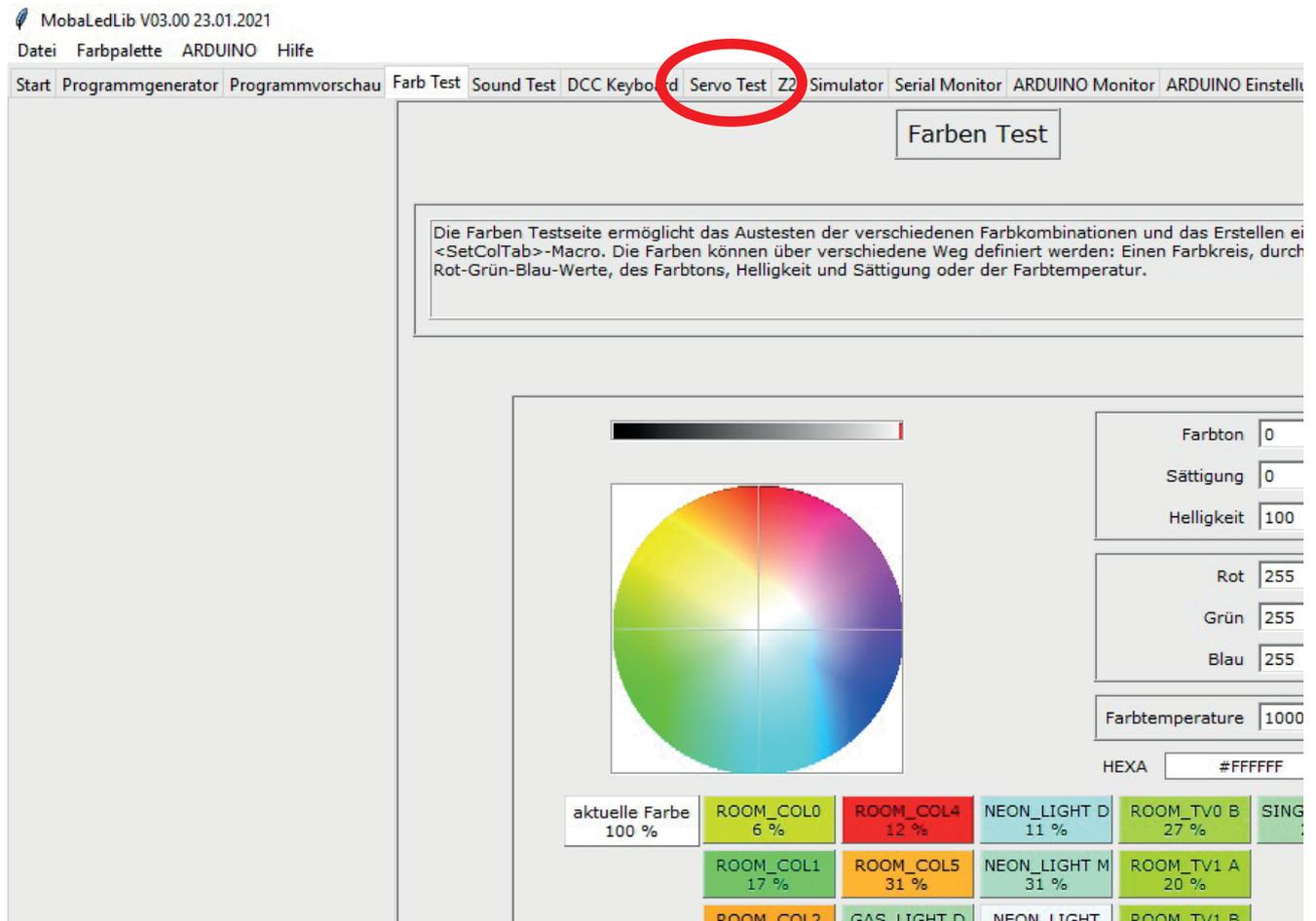
Als nächstes wird das Programm „Prog\_Generator“ gestartet und dort auf die Schaltfläche „Optionen“ geklickt.



Dann bitte auf „LED Farbtest starten“ klicken.



Es öffnet sich jetzt das „Farbtest“-Fenster. Hier bitte auf den Reiter „Servo Test“ klicken.



Jetzt wird das Fenster „Servotester“ angezeigt. Hier können nun die Servos eingestellt werden.

Zuerst muss das richtige Servo angesteuert werden. Im Feld „Servo Adresse“ muss die Nummer der Servoplatine in der „LED-Kette“ eingetragen werden. Ist auf der Hauptplatine eine „Heartbeat-LED“, dann hat diese die Nummer 0. Ist dann - so wie bei meinem Testaufbau - die Servoplatine direkt an die Hauptplatine angeschlossen, so hat diese die Nummer 1. Diese 1 wird dann bei „Servo Adresse“ eingegeben. Befinden sich zwischen „Heartbeat“ und Servoplatine noch andere LEDs, so hat die Servoplatine natürlich eine abweichende Nummer.

Bei „Servo Anschluss“ kann das jeweils einzustellende Servo an der vorher definierten Servoplatine ausgewählt werden. Da bis zu 3 Servos an einer Servoplatine eingesteckt werden, haben diese die Nummern 0, 1, oder 2 (Achtung: immer bei 0 anfangen zu zählen!).

13.00 23.01.2021

ARDUINO Hilfe

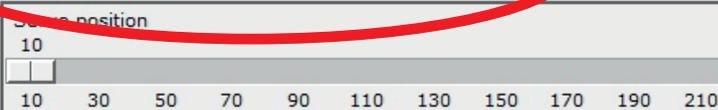
nerator Programmvorschau Farb Test Sound Test DCC Keyboard Servo Test Z21 Simulator Serial Monitor ARDUINO Monitor ARDUINO Einstellungen

## MobaLedLib LED Servotester

Mit der Servoplatine von Hardi können über die MLL auch Servos angesteuert werden. Auf dieser Seite können Sie den Servo testen und konfigurieren.  
Mit dem Schieberegler kann die Position des Servos eingestellt werden. Die Konfiguration des Servos erfolgt mit den Funktionen <Min und Max Position programmieren> und <Servo Geschwindigkeit programmieren>

Servo Adresse 0 Servo Anschluss 0

Servo STOP



Programmierung von Min/Max Position und Geschwindigkeit

Starte Min-Max Pos/Speed programmieren

Beende Programmierung ohne Speichern

Dec &lt;&lt;

Dec &lt;

Inc &gt;

Inc &gt;&gt;

mit COM5 - EFFECT Mode

Ist das richtige Servo ausgewählt wird die mittlere Position des Servos bestimmt. Hierzu den Schieber auf ca. 105 in die Mitte schieben. Das Servo müsste sich jetzt bewegen.

13.00 23.01.2021

ARDUINO Hilfe

nerator Programmvorschau Farb Test Sound Test DCC Keyboard Servo Test Z21 Simulator Serial Monitor ARDUINO Monitor ARDUINO Einstellungen

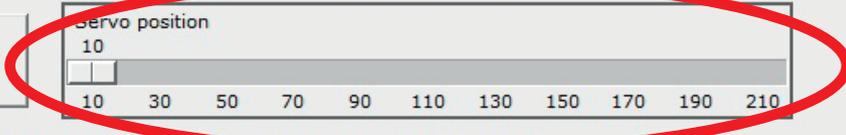
### MobaLedLib LED Servotester

Mit der Servoplatine von Hardi können über die MLL auch Servos angesteuert werden. Auf dieser Seite können Sie den Servo testen und konfigurieren.  
Mit dem Schieberegler kann die Position des Servos eingestellt werden. Die Konfiguration des Servos erfolgt mit den Funktionen <Min und Max Position programmieren> und <Servo Geschwindigkeit programmieren>

Servo Adresse 0 Servo Anschluss 0

Servo STOP

Servo position  
10



10 30 50 70 90 110 130 150 170 190 210

Programmierung von Min/Max Position und Geschwindigkeit

Starte Min-Max Pos/Speed programmieren

Beende Programmierung ohne Speichern

Dec << Dec < Inc > Inc >>

mit COM5 - EFFECT Mode

Ist die Mittelposition richtig werden die linken und rechten Maximal-Aus-schläge definiert. Hierzu bitte einmal auf die breite Taste „Starte Min-Max Pos/Speed programmieren“ klicken. Nach dem Klicken ändert sich die Tastenbe-schriftung und jetzt kann man mit den unteren Tasten „Dec <<“, „Dec <“, „Inc >“ und „Inc >>“ die „linke“ Maximalposition anfahren. Die Tasten mit den Doppelpfeilen bewegen das Servo schneller, die Tasten mit den Einfachpfeilen ganz langsam (kaum sichtbar). Ist dieser Anschlag fertig angefahren, wieder auf die obere große Taste klicken und dann mit den Pfeiltasten die „rechte“ Maximalposition anfahren. Mit einem nochmaligem Klick auf die obere große Taste kann man die Stellgeschwindigkeit des Servos mittels der unteren Pfeil-tasten einstellen. Zum Abschluss nochmals die obere große Taste anklicken und schon ist man fertig. Für das nächste Servo an gleicher Platine dann den Eintag bei „Servo Anschluss“ um 1 erhöhen und die Prozedur wiederholen. Wenn man fertig ist, einfach das Fenster schließen.

13.00 23.01.2021

ARDUINO Hilfe

nerator Programmvorschau Farb Test Sound Test DCC Keyboard Servo Test Z21 Simulator Serial Monitor ARDUINO Monitor ARDUINO Einstellungen

## MobaLedLib LED Servotester

Mit der Servoplatine von Hardi können über die MLL auch Servos angesteuert werden. Auf dieser Seite können Sie den Servo testen und konfigurieren.  
Mit dem Schieberegler kann die Position des Servos eingestellt werden. Die Konfiguration des Servos erfolgt mit den Funktionen <Min und Max Position programmieren> und <Servo Geschwindigkeit programmieren>

Servo Adresse  Servo Anschluss 

Servo STOP

Servo position

10

10 30 50 70 90 110 130 150 170 190 210

Programmierung von Min/Max Position und Geschwindigkeit

Starte Min-Max Pos/Speed programmieren

Beende Programmierung ohne Speichern

Dec &lt;&lt;

Dec &lt;

Inc &gt;

Inc &gt;&gt;

it COM5 - EFFECT Mode

In meinem Lokschuppen habe ich bausatzbedingt einen Servo spiegelverkehrt eingebaut. Hierfür muss ich die Drehrichtung - gegenüber den anderen beiden Servos - ändern.

Das ist eigentlich ganz einfach - wenn man weiß wie es geht. Das wird auch mit der Definition der Min. und Max.-Stellungen gemacht. Ich musste nur umdenken. Statt maximaler „linker“ und maximaler „rechter“ Position habe ich - gedanklich - als erste einzustellende Min.-Position die Servostellung „Tor geschlossen“ angefahren und als zweite einzustellende Max-Position die Servostellung „Tor geöffnet“ eingestellt. Bei zwei Servos konnte ich die erste Position durch Klicken auf die Linkspfeil-Tasten und die zweite Position durch Klicken auf die Rechtspfeil-Tasten anfahren. Bei dem andersherum eingebauten Servo klickte ich für die erste Min.-Position auf die Rechtspfeil-Tasten und für die zweite Max.-Position auf die Linkspfeil-Tasten. Das funktioniert hervorragend!



*So habe ich die 3 Servos in den Lokschuppen eingebaut.*

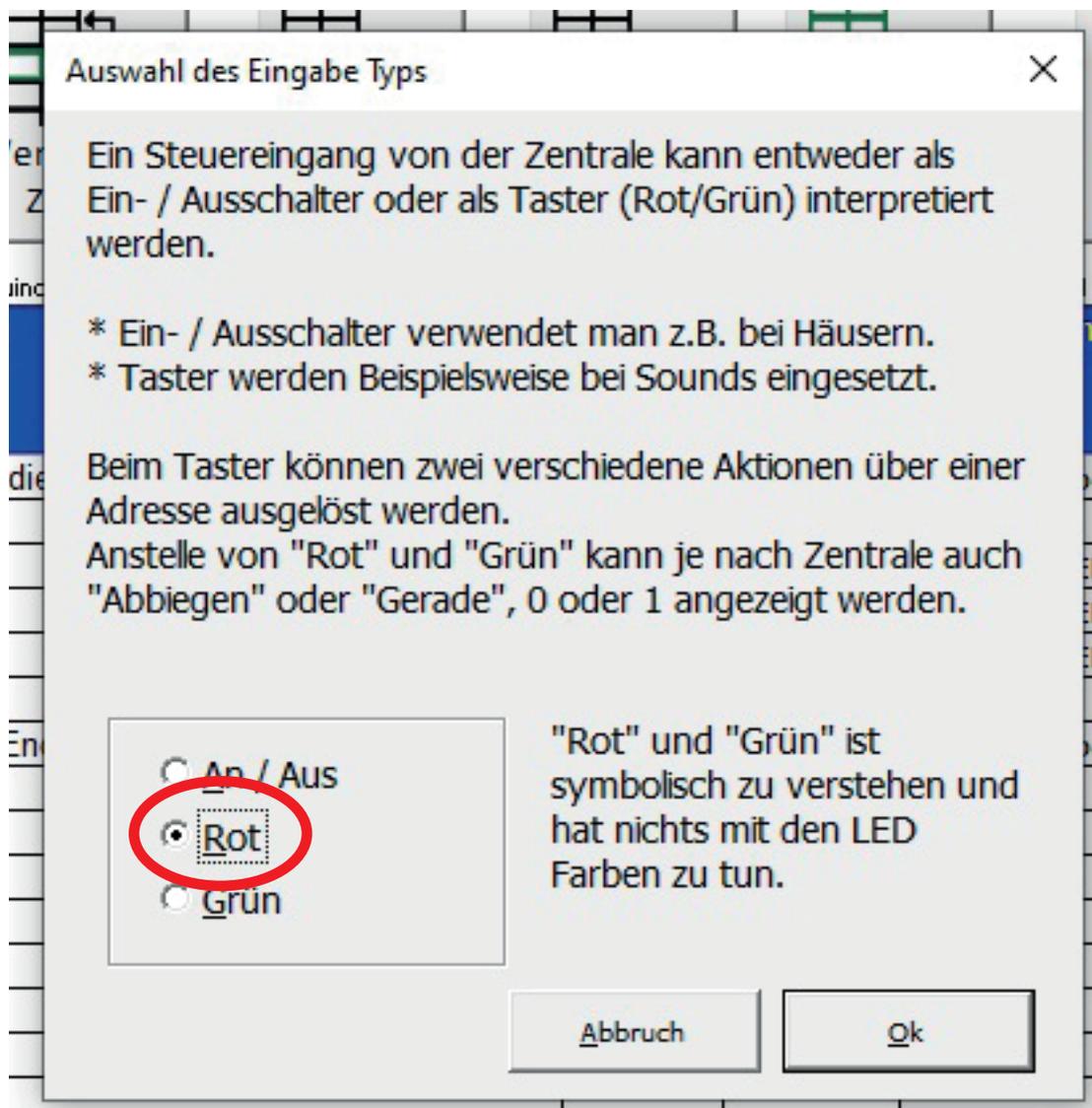
# Servos im Prog\_Generator anlegen

Jetzt müssen die Servos noch i Prog\_Generator angelegt werden. Hierzu einfach auf eine leere Zeile klicken und dann auf die obere Taste „Dialog“.

The screenshot shows the Prog\_Generator\_MobaLedLib software interface. The 'Dialog' button in the top toolbar is circled in red. The main window displays a table with the following columns: Filter, Adresse oder Name, Typ, Startwert, Beschreibung, Verteiler-Nummer, Stecker-Nummer, Beleuchtung, Sound, oder andere Effekte, Start LedNr, LEDs, InCr, and LoInCr. The table contains two rows of data:

	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte	Start LedNr	LEDs	InCr	LoInCr
3	✓				Zeigt an, dass die LEDs angesteuert werden			RGB_Heartbeat(#LED)	0	1	0	0
8	✓				Heartbeat am Ende der LED-Kette			RGB_Heartbeat(#LED)	1	1	0	0

Als erstes wird die DCC-Adresse abgefragt. Hier habe ich für den ersten Servo eine „100“ eingegeben. als nächstes wird die „Auswahl des Eingabetyps“ angezeigt. Da das Servo später in Traincontroller durch einen Umschalter geschaltet werden soll, habe ich hier „Rot“ markiert.



Anschließend kommen noch einige Abfragen zur Platinen- oder Steckernummer und zur Bezeichnung. Hier können Angaben gemacht werden, dies sind aber keine Pflichtangaben. Ich kann aber nur empfehlen, hier einen aussagekräftigen Namen für diese Funktion zu hinterlegen.

Weiter geht es zur „Makroauswahl“. Da das Servo nur zwei Positionen (Tor auf und Tor zu) ansteuern soll wähle ich hier „Servo2“ aus.

Auswahl des Makros

Makroauswahl:

Tipp: Schnellauswahl mit Eingabe des Anfangsbuchstabens

Name	Beschreibung
EntrySignal3_RGB	Einfahrtssignal mit 3 RGB LEDs zu Testzwecken (HP0, HP1, HP2). Es wird über drei Taster gesteuert.
EntrySignal3	Einfahrtssignal mit 3 einzelnen LEDs welche über ein WS2811 Modul angesteuert werden. (HP0, HP1, HP2). Es wird über drei Tas
DepSignal4_RGB	Ausfahrtssignal mit 6 RGB LEDs zu Testzwecken (HP0, HP1, HP2, HP0+SH1). Es wird über vier Taster gesteuert.
DepSignal4	Ausfahrtssignal mit 6 einzelnen LEDs welche über zwei WS2811 Module angesteuert werden. (HP0, HP1, HP2, HP0+SH1). Es wird
EntrySignal3Bin_RGB	Einfahrtssignal mit 3 RGB LEDs zu Testzwecken (HP0, HP1, HP2). Es wird über zwei binäre Eingänge gesteuert. Damit eignet es sic
EntrySignal3Bin	Einfahrtssignal mit 3 einzelnen LEDs welche über ein WS2811 Modul angesteuert werden. (HP0, HP1, HP2). Es wird über zwei bin
DepSignal4Bin_RGB	Ausfahrtssignal mit 6 RGB LEDs zu Testzwecken (HP0, HP1, HP2, HP0+SH1). Es wird über zwei binäre Eingänge gesteuert. Damit
DepSignal4Bin	Ausfahrtssignal mit 6 einzelnen LEDs welche über zwei WS2811 Module angesteuert werden. (HP0, HP1, HP2, HP0+SH1). Es wird
KS_Vorsignal_Zs3V_RGB	RGB Ks-Vorsignalwiederholer: Ks 2 / Ks 1 / Ks 1 + Zs 3V / Kennlicht
KS_Vorsignal_Zs3V	Einzel LED Ks-Vorsignalwiederholer: Ks 2 / Ks 1 / Ks 1 + Zs 3V / Kennlicht
KS_Hauptsignal_Zs3_Zs1_RGB	RGB Ks-Hauptsignal: Hp 0 / Ks 1 / K2 1 + Zs 3 / Hp 0 + Zs 1
KS_Hauptsignal_Zs3_Zs1	Einzel LED Ks-Hauptsignal: Hp 0 / Ks 1 / K2 1 + Zs 3 / Hp 0 + Zs 1
KS_Hauptsignal_Zs3_Zs6_Zs1_RGI	RGB Ks-Hauptsignal: HP 0 / Ks 1 / Ks 1 + Zs 3 + Zs 6 / HP 0 + Zs 1
KS_Hauptsignal_Zs3_Zs6_Zs1	Einzel LED Ks-Hauptsignal: HP 0 / Ks 1 / Ks 1 + Zs 3 + Zs 6 / HP 0 + Zs 1
RGB_Heartbeat	RGB LED welche als Funktionsindikator in wechselnden Regenbogenfarben blinkt.
RGB_Heartbeat2	RGB LED welche als Funktionsindikator in wechselnden Regenbogenfarben blinkt. Minimale und maximale Helligkeit kann angegabe
Servo2	Servo Steuerung mit zwei Positionen welche über Taster angesteuert werden.
Servo3	Servo Steuerung mit drei Positionen welche über Taster angesteuert werden.
Servo4	Servo Steuerung mit vier Positionen welche über Taster angesteuert werden.
Servo5	Servo Steuerung mit fünf Positionen welche über Taster angesteuert werden.
Herz_BIRelais_V1_AB	Herzstückpolarisierung HW 1 mit einem bistabilen Relais welches verzögert angesteuert wird.
Herz_BIRelais_I_V1_AB	Herzstückpolarisierung HW 1 mit einem bistabilen Relais welches invers verzögert angesteuert wird.

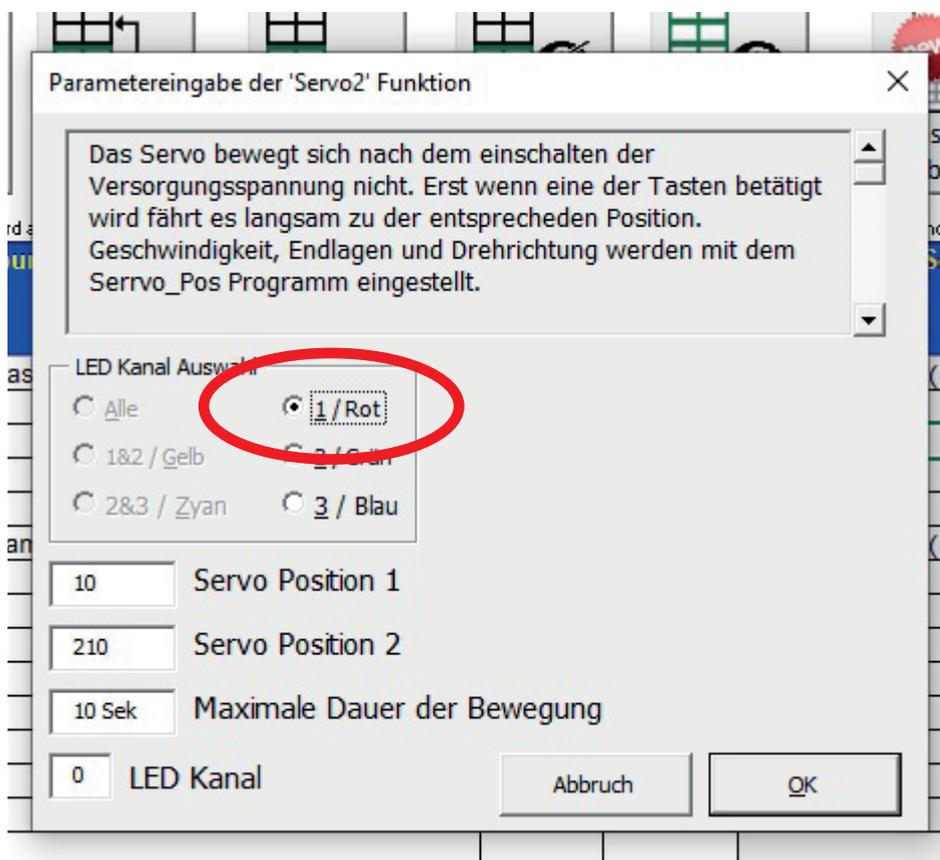
Das Servo bewegt sich nach dem einschalten der Versorgungsspannung nicht. Erst wenn eine der Tasten betätigt wird fährt es langsam zu der entsprechenden Position. Geschwindigkeit, Endlagen und Drehrichtung werden mit dem Servo\_Pos Programm eingestellt.

Servo2(LED, InCh, B\_LED\_Cx, Servo21, Servo22, ServoT)

Expertenmodus

Abort Auswahl

Als letztes muss ich noch die richtige Nummer des auf der Platine anzusprechenden Servos auswählen. Das erste Servo auf der Platine wird mit dem Rot-Kanal angesteuert, das zweite Servo mit dem Grün-Kanal und das dritte Servo mit dem Blau-Kanal. Für das 2. und das 3. Servo muss ich im Prog.-Generator eine neue Zeile anlegen, eine neue DCC-Nummer vergeben und dann im unteren Fenster die 2/Grün bzw. 3/Blau auswählen. Komischerweise wird jetzt auf einmal beim Zählen mit „1“ begonnen.



So sehen die drei Zeilen für die 3 Servos im Prog\_Generator aus. Als letzter Schritt wird diese Konfiguration zur Hauptplatine / Arduino übertragen und schon ist „der Drops gelutscht“.

The screenshot shows the MobaLedLib software interface. The main window displays a table with columns for configuration parameters. A red oval highlights the rows for three servos (rows 5, 6, and 7).

Ver.	by	Start	LEDs	Loc
2.0.0	Hal	LEDnr	InCh	InCh
		0	1	0
		1	C1-1	0
		1	C2-2	0
		1	C3-3	0

The table also includes columns for 'Aktiv', 'Filter', 'Adresse oder Name', 'Typ', 'Startwert', 'Verteiler Nummer', 'Stecker Nummer', and 'Effekte'. The highlighted rows correspond to the servo configurations mentioned in the text.

## Ansteuerung der Tore in Traincontroller

Im Traincontroller schalte ich auf „Editor-Modus“ um. Auf die entsprechenden Abstellgleise des Lokschuppens füge ich je ein Symbol „Tor“ ein. Druck Doppelklick auf ein positioniertes „Tor“-Symbol kann ich die Eigenschaften eingeben. Name, Digitalsystem und die vorher im Prog\_Generator hinterlegte DCC-Adresse sind schnell eingegeben. Zum direkten Test kann ich im Eigenschaften-Fenster auf den grün/roten Testbutton klicken. Sollte jetzt bei einem Klick auf den roten Button das Tor öffnen und bei einem Klick auf den grünen Button wieder schließen so muss ich nicht die Verkabelung ändern. Unter dem Testbutton kann ich die „Anschlüsse“ durch Klicken tauschen und schon gehen bei rot die Tore zu und bei grün die Tore auf.

Wenn ich jetzt den Editor-Modus verlasse und eine Zugfahrt von der Schiebebühne in den Lokschuppen auslöse, so öffnet sich das Tor automatisch und nach der Einfahrt der Lok in den Schuppen schließt sich das Tor wieder.

So macht Basteln Spaß!!!

## Links

Hier noch einige Links:

MobaLedLib im Stummiforum:

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

MobaLedLib-Wiki:

<https://wiki.mobaledlib.de>

## Danksagung

Ganz zum Schluss möchte ich - zum wiederholten Male - den Initiatoren der MobaLedLib danken. Was Ihr hiermit uneigennützig auf die Beine gestellt habt und dann kostenlos zur Verfügung stellt (natürlich bis auf Platinen und elektronische Teile) ist „erste Sahne“!

Außerdem möchte ich mich bei allen Stummis bedanken, die im Stummi-Forum einen super schnellen und sehr kompetenten „Support“ zum Thema „MobaLedLib“ bieten.

Jeder Interessent ist herzlich eingeladen, im Stummi-Forum (unter „MobaLedLib“) Fragen zu stellen, Informationen zu erhalten, an der MobaLedLib mit weiterzuentwickeln oder einfach nur über eigene Projekte zu berichten.

Stand 3/2021 (Ver. C)

Rückfragen zu dieser Schrift  
bitte nur per eMail an: [jochem@familieheinen.name](mailto:jochem@familieheinen.name)