

LichtMaschine PICO Zero

(Art. Nr.: MLL010)



hier als

Konstruktionsbild

und so sieht die Zentrale dann in original aus (Auslieferungszustand)

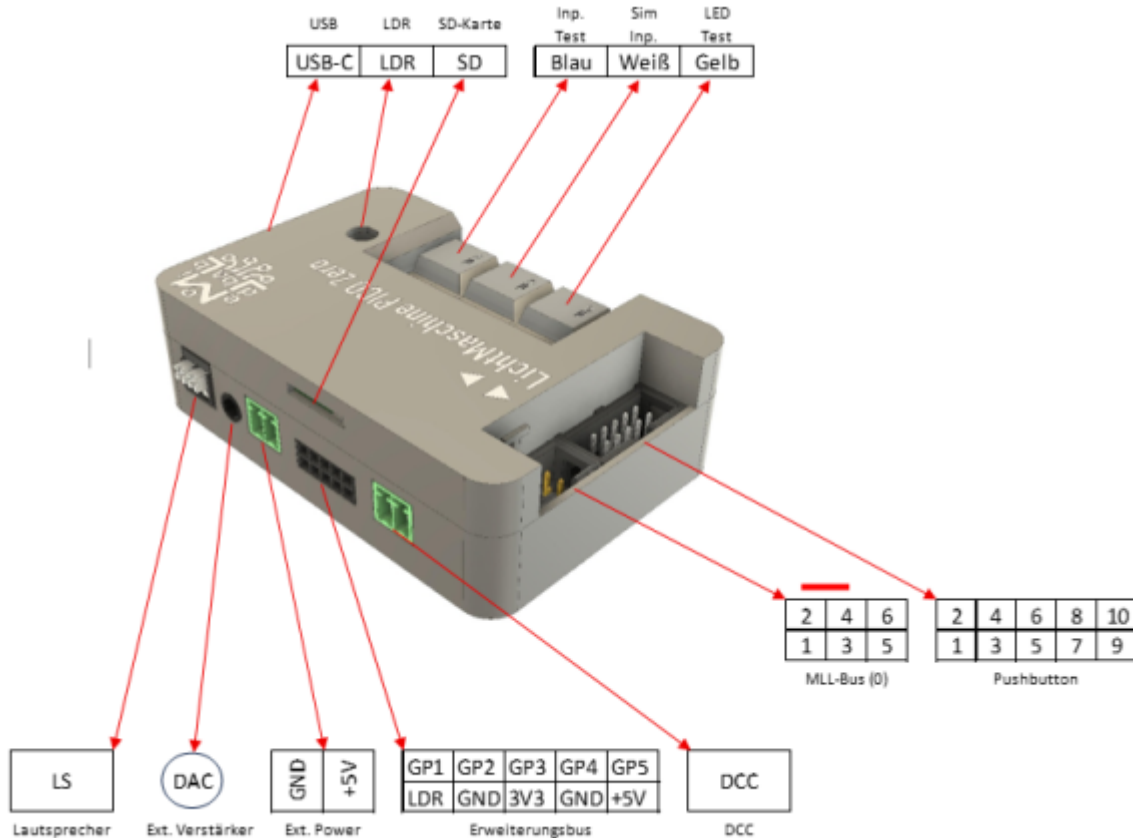




Das erste einsatzbereite MobaLedLib-Modul mit vielen Möglichkeiten

- - **Kompakt**, nur 8 x 5,4 x 2,4 cm (LxBxH)
- - **Montiert geliefert**, inklusive Gehäuse

Anschlüsse



- - **MLL-Bus (0)**
- - **Pushbutton (SwitchB)**

1. MLL-Pushbutton (TMaa Version)
2. 2 x 5-polig - über Adapter auch für MLL Platine 300 und 301 geeignet

- - **USB-C**, Programmierverbindung, +5V
- Die drei **Standard-MLL-Tasten** mit den entsprechenden **LEDs**

1. o Gelb, LED Test $\Leftarrow$ / L+R=All
2. o Weiß, Sim Inp. Long=ESC
3. o Blau, Inp. Test $\Rightarrow$

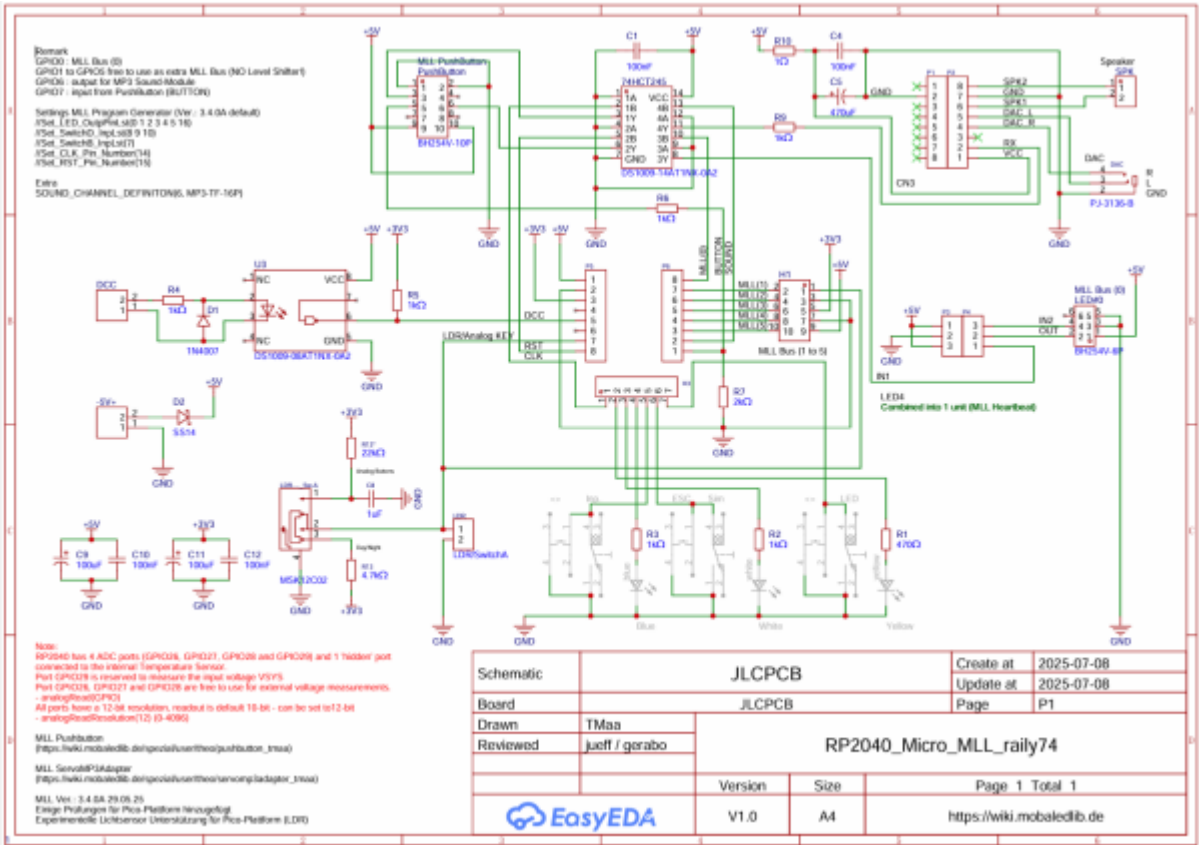
- - **LDR oder Analog Taster (SwitchA)**, wählbar über einen Schalter auf der Platine
- - **DCC-Anschluss**
- - **Externer Stromanschluss +5V**, Automatische Auswahl USB oder Extern
- - **MP3-Player**, mit SD-Kartensteckplatz
- - **Lautsprecheranschluss**, 8Ω/1W
- - **Anschluss für externen Verstärker**, Klinkenstecker
- - **Erweiterungsbus**, für fortgeschrittene Anwender!

Direkter Zugriff auf RP2040-Zero ([Waveshare RP2040-Zero](https://www.waveshare.com/rp2040-zero.htm))

1. o GP1 bis GP5 Diese Ports können beispielsweise über einen Level-Shifter für den MLL-Bus (1 bis 5) genutzt werden. Die Definition hierfür ist bereits im Program-Generator festgelegt. (ab Version 3.4.0) (Pin 2 bis 6)
2. o +5V (Pin 23)
3. o +3V3 (Pin 21)
4. o LDR/Analog-Taster (SwitchA) (Pin 17 – GP26)

5. o GND (Pin 22)

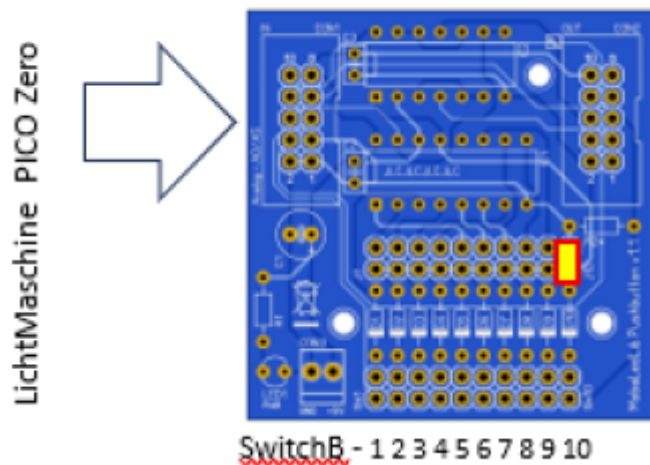
Schaltplan



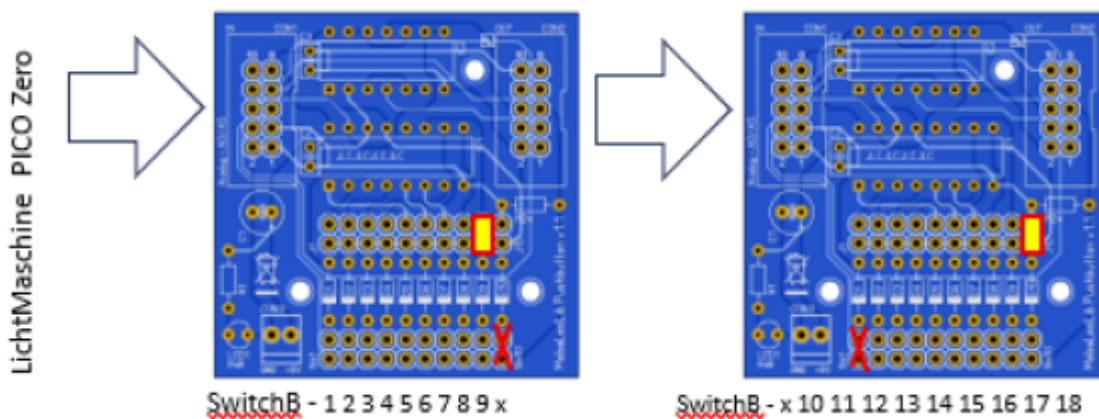
Pushbutton

MLL-Pushbutton-platinen können bis zu 10 Stück kaskadiert werden und mit dem 2×5-Pushbutton-anschluss verbunden.

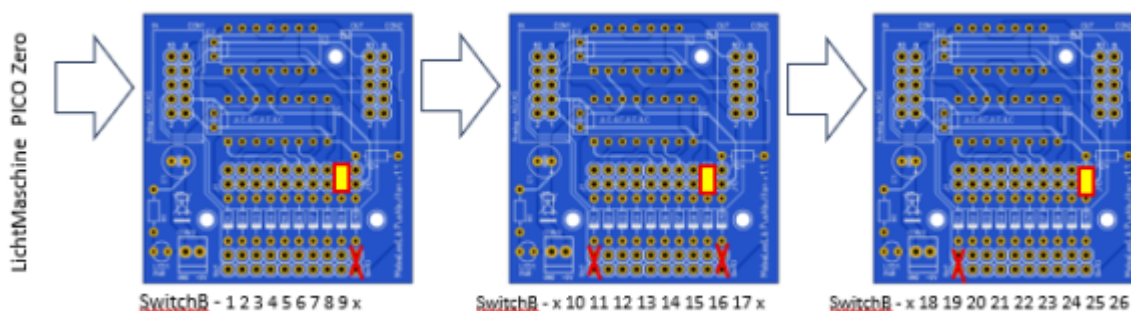
Bei Verwendung nur einer Platine können 10 Schalter oder Taster angeschlossen werden. (Die korrekte Jumper-Einstellung finden Sie im Beispiel



Bei Verwendung von zwei Platinen opfert die erste Platine ihren letzten Anschlusspunkt und die zweite ihren ersten. Insgesamt ergeben sich so 9 + 9 Anschlusspunkte. (Dies muss durch Jumper gekennzeichnet werden – siehe Beispiel –



Bei mehr als zwei Platinen wiederholt sich dieser Vorgang, wobei die zweite Platine nicht nur ihren ersten, sondern auch ihren letzten Anschlusspunkt opfert. Insgesamt ergeben sich so 9 + 8 + 9 Anschlusspunkte. (Dies muss ebenfalls durch Jumper gekennzeichnet werden – siehe Beispiel

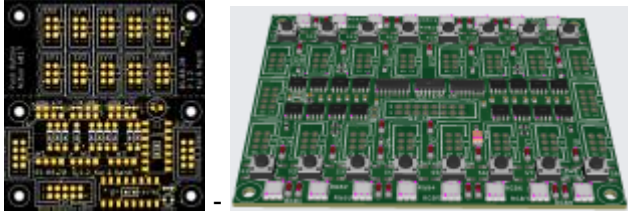


Siehe auch die Wiki-Seiten zu den Platinen [300](#) und [301](#).

Die hier verwendete Methodik ist identisch mit der der hier anzuschließenden MLL-Pushbutton-Platinen.

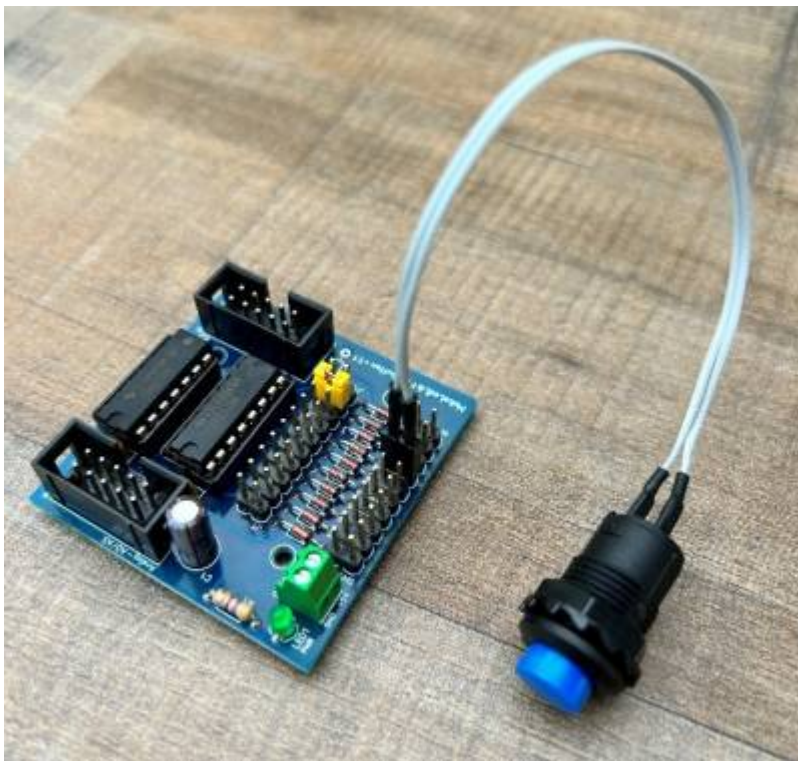
Daher können diese Platinen ebenso eingesetzt und auch kaskadiert werden.

Abbildungen zeigen die Push-Button Platinen 300 und 301:



Siehe hierzu auch das Dokument: [Schalter und Taster einlesen mit der MobaLedLib.pdf](#)

Die Pushbutton-platinen verwenden alle ein Schieberegister, das aus einem CD4017 und einem CD4011 IC besteht. Wer mehr darüber lesen möchte, findet es in der Elektor-Halbleiterzeitschrift von 1984, insbesondere in der Schaltung Nr. 99 (Schaltplan 84506) – „Schieberegister mit Schieberegistern 4017“ auf Seite 110 □



Das Foto zeigt einen Taster auf einer Zwischenplatine (der Jumper befindet sich in der vorletzten Position) und platziert auf der siebte Anschluss. (Im oben beschriebenen Beispiel ist dies SwitchB15.)

MP3-Player

Die integrierten Musikdateien liegen im MP3 oder WAV-Format vor und können auf eine Micro-SD-

Karte geladen werden.

Die gespeicherten Dateien werden in der Reihenfolge abgespielt, in der sie gespeichert wurden, nicht alphabetisch!

Im Datei-Explorer werden sie alphabetisch aufgelistet. Das kann sehr verwirrend sein. Um sie in einem wirklich übersichtlichen Format anzuzeigen,

müssen Sie sie sortieren. Der Haken: Nicht jedes Programm kann das. [SD-Sorter](#) kann sie sicher sortieren und verhindert so, dass Sie versehentlich Ihre eigene Festplatte sortieren.

Es sortiert nur die Micro-SD-Karte. Das Programm ist zwar auf Englisch, bietet aber so wenige Funktionen, dass es durchaus zu bewältigen ist.

Auf meinem Windows 11 Rechner (vorher Windows 10) läuft es einwandfrei.

Datei 1 ist die allererste gespeicherte Datei, unabhängig vom Namen. Datei 2 ist die nächste gespeicherte Datei usw.

Sie können ihr logische Namen mit einer vorangestellten Nummer geben, sodass dieselbe Nummer wie im Dateinamen aufgerufen wird. Beispiel: 001_Test.mp3 – dann ist 001 die erste Datei, der Rest ist lediglich ein Name zur Identifizierung.

Was bleibt ist, dass im Program-Generator (PG) eine der ersten Zeilen anzeigt, dass das Musikmodul an Port 6 angeschlossen ist. Verbunden mit dem „Mainboard“.

SOUND_CHANNEL_DEFINITON(6, MP3-TF-16P)

Geluid MP3-TF-16P Nieuw MP3-TF-16P JQ6500 JQ6500 blauw <-> groen ATTiny85 Moederbord Selecteer geluidsmodule Speel het nummer # Speel willekeurig nummer af Toneelstuk Pauze Eindeloos afspelen Volume verhogen Volume verminderen Definieer volume	Akoestiek voor de modelbaan met geluidsmodules Geluidsfuncties voor nieuwe en oude MP3-TF-16P-geluidsmodules Geluidsfuncties voor de oudere MP3-TF-16P-geluidsmodule Geluidsfuncties voor de JQ6500 geluidsmodule *** Geluidsfuncties voor de JQ6500-geluidsmodule met een andere WS2811 *** Geluidsmodules via ATTiny85 Geluidsmodules die rechtstreeks op het moederbord zijn aangesloten Selecteer geluidsmodule Definieer de pin van de seriële geluidsmodule Track: # vom angegebenen Modul abspielen (Rootverzeichnis) Zufälligen Titel vom angegebenen Modul abspielen (Rootverzeichnis) Gaat door met het afspelen van geluid Pauzeert het afspelen van geluid Stelt de lusmodus in Volume verhogen Volume verminderen Volume instellen tot 0..100%
--	---

Parameterinvoer van de 'SOUND_CHANNEL_DEFINITON' functie

Definieer een pin op de hoofdprintplaat voor de seriële aansturing van een geluidsmodule en het type van de aangesloten geluidsmodule.
Het pinnummer kan worden opgegeven als een getal of als een symbolische constante.
Voorbeeld: KEY80_P1

Het type bepaalt het protocol waarmee de geluidsmodule wordt aangestuurd.
Sommige MP3-TF-modules hebben een fout en moeten worden geadresseerd met de variant NO_CRC (geen controlesom). Modules met de volgende geluidschipnummers zijn tot nu toe als defect aangemerkt: MH2024K-16SS

6| Arduino-pincode

MP3-TF-16P MP3-TF-16P MP3SerialType

0 Geluidskanaal

Annuleren OK

Sehen Sie sich auch das DEMO-Programm an, um zu sehen, wie es verwendet wird.

LDR oder Analoge Tasten

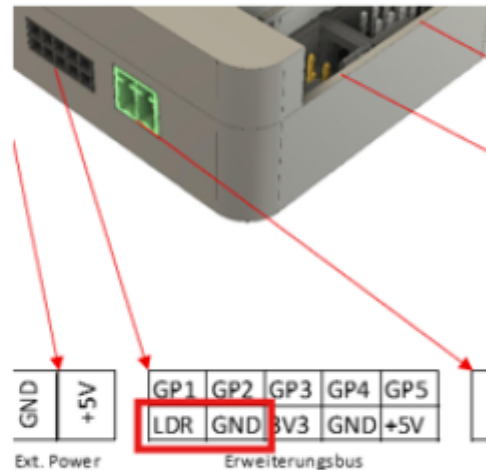
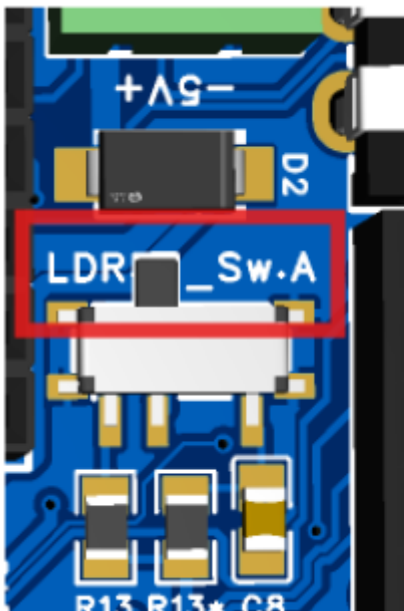
Light **D**ependent **R**esistor - Lichtempfindlicher Widerstand.

Die Elektronik ist bereits in die Hardware integriert.

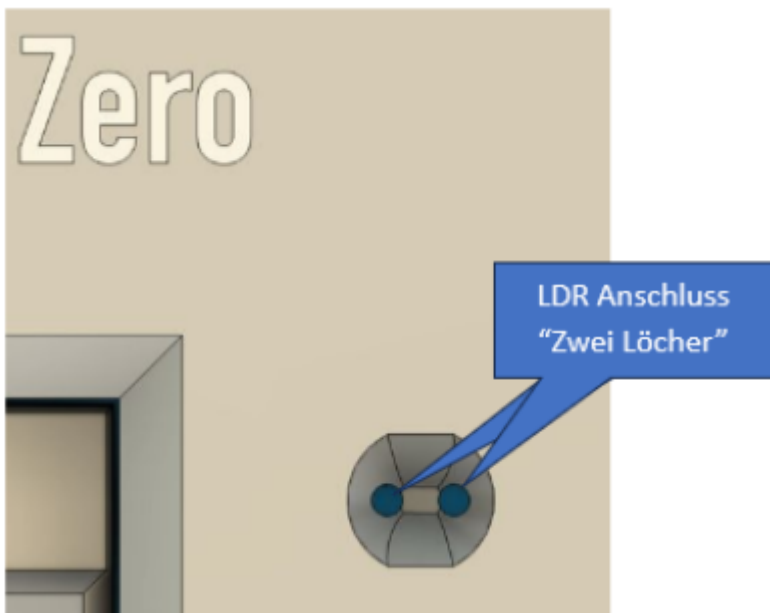
Auf der Platine im Gehäuse befindet sich ein Schalter, mit dem Sie wählen können, ob Sie einen LDR oder analoge Tasten

an den externen Ein-/Ausgang anschließen möchten. Standardmäßig ist ein LDR installiert, der Schalter steht auf LDR,

und Sie können mit dem PG eine funktionierende Anwendung erstellen.



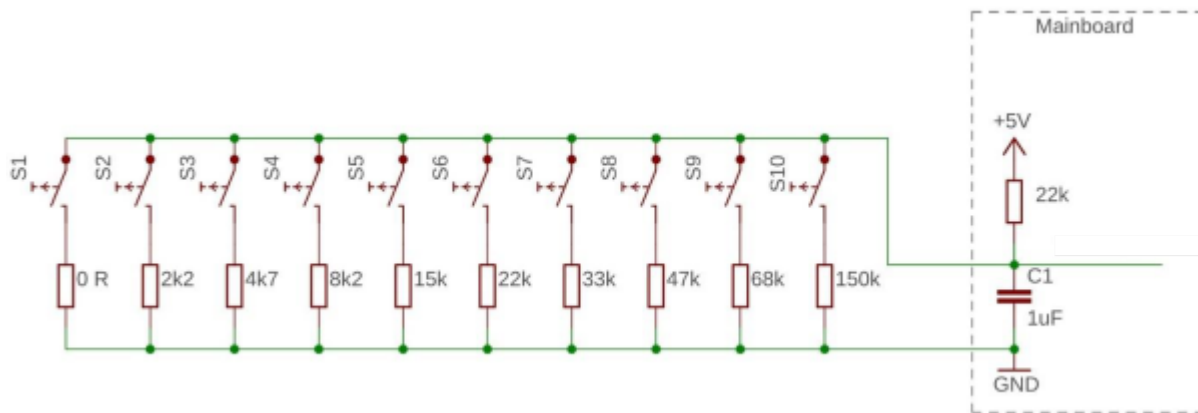
Um analoge Tasten an den externen Eingang anzuschließen, stellen Sie den Schalter auf **Sw.A** und entfernen Sie den installierten LDR. Sie können ihn einfach mit einer Pinzette herausziehen. Ein Saugnapf oder ein Stück Klebeband funktionieren auch.
Der Wiedereinbau erfolgt in umgekehrter Reihenfolge: Setzen Sie den LDR in die Aussparung oben am Gehäuse ein.
(Es gibt zwei Löcher, die die Beine des LDR zum Anschluss führen.)



Ich habe zufällig entdeckt, dass man Anwendungen, die normalerweise mit einem Taster (SwitchAx) bedient werden, steuern kann, wenn man den LDR/Analog-Tastenschalter auf Sw.A stellt und den LDR angeschlossen lässt, indem man sie nun vom eingehenden Licht am LDR abhängig macht.
Dann müssen keine externen Tasten mehr angeschlossen werden! Die Lichtintensität wird in einen entsprechenden Tastendruck übersetzt, wodurch Sie eine Anwendung starten können. So können Sie beispielsweise das Licht bei Dunkelheit automatisch einschalten, indem Sie eine Aktion mit der entsprechenden Taste verknüpfen – SwitchAx. (Die Taste „x“ hängt dann von der Lichtmenge ab.)

Hinweis: Verwenden Sie für die analogen Tasten denselben Standardsatz von 1 %-Widerständen wie bei anderen MLL-LichtMaschinen.

(Der 22K-Widerstand und der 1uF-Kondensator sind bereits auf dem Mainboard installiert und müssen daher nicht mehr installiert werden)

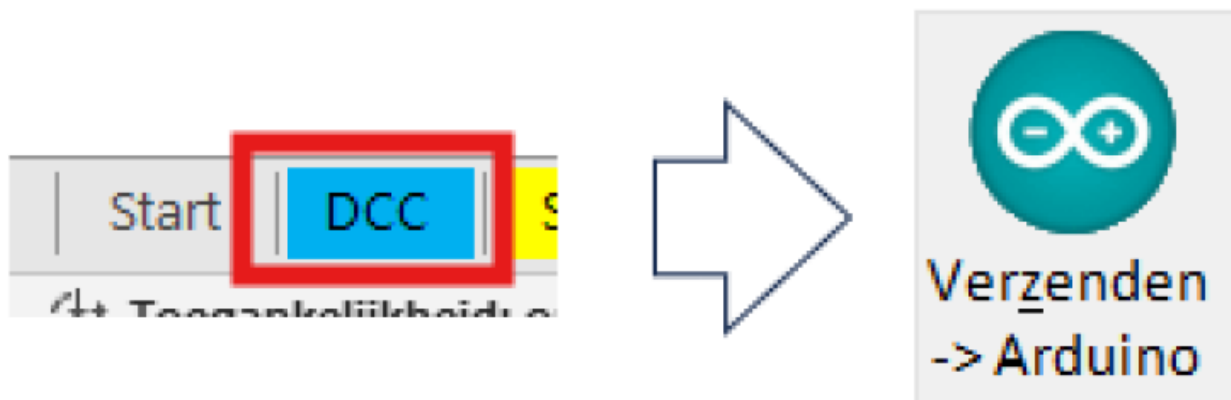


Programmierung

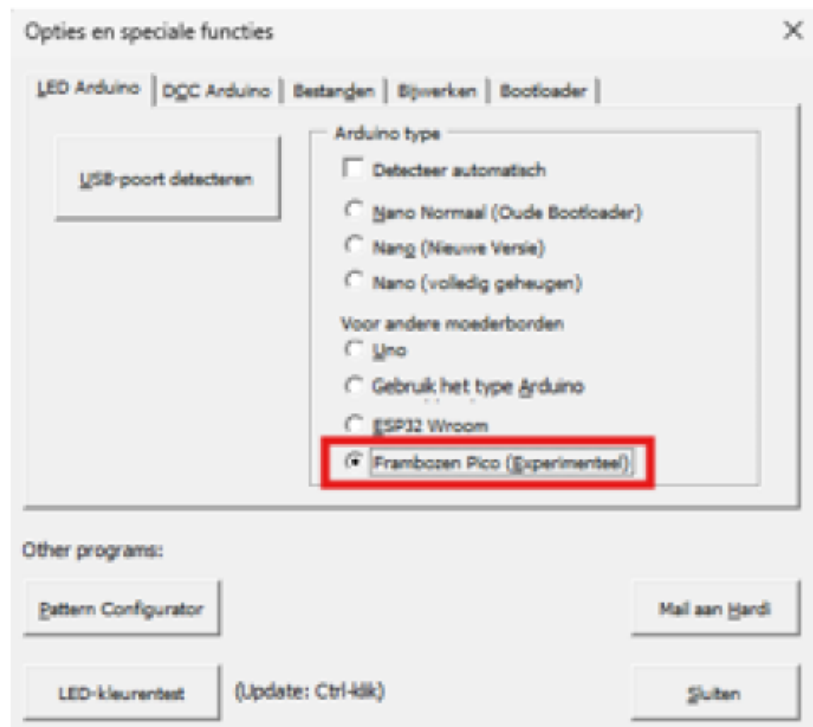
Programmierung der MLL-Anwendung vom Program-Generator (PG) auf die LichtMaschine PICO Zero. Die LichtMaschine PICO Zero wird mit einer vorinstallierten „Demo“-Anwendung geliefert.

Wenn Sie die LichtMaschine anschließen und vom PG aus starten, kann eine Anwendung direkt geladen werden.

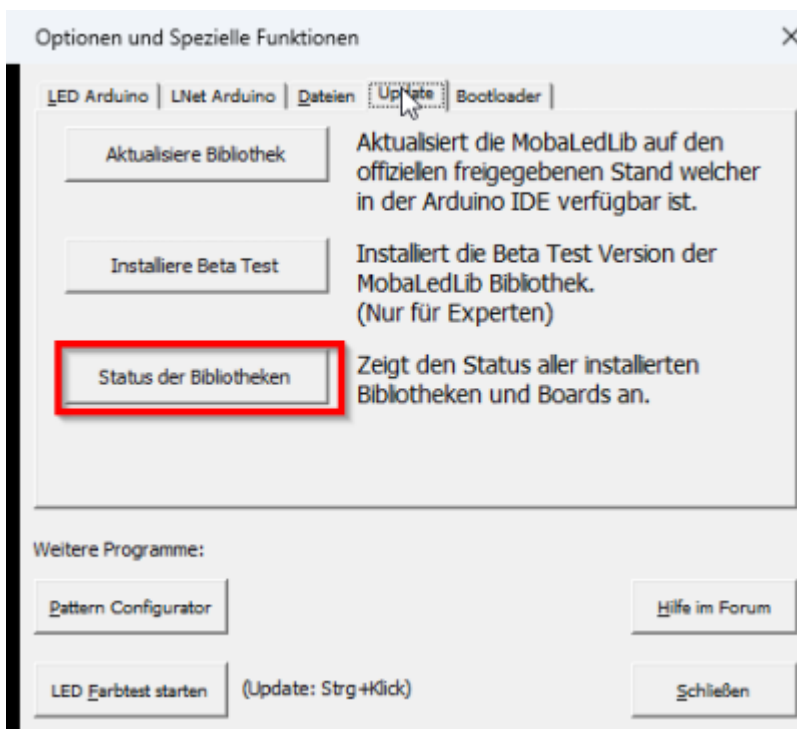
Voraussetzung ist, dass Sie im PG über den Reiter „DCC“ eine Anwendung schreiben und von dort an den Arduino senden.



Außerdem müssen Sie bei der ersten Verwendung dieses Reiters angeben, dass es sich bei der verwendeten LichtMaschine um einen Raspberry handelt.



Hinweis: Bei der aktuellen Version des Prog_Generators wird der Raspberry in der Auswahl noch nicht angezeigt. Hier muss man einen kleinen Eingriff vornehmen: Nach Klick auf „Optionen“ das Register „Update“ auswählen und den Button „Status der Bibliotheken“ anklicken.



Damit öffnet sich die Seite mit den allen Bibliotheken. Hier in der Zeile mit dem rp2040 in der vordersten Spalte einen Haken setzen (einfach einmal rein klicken) und den Button „Install selected“ betätigen.

Check actual versions		Last update	Arduino IDE	1.8.10	Sketchbook_Path	C:\Users\peter\Documents\Arduino	Open Path
Install selected		20.8.26 18:32	1.8.10				
Delete selected							
Index	Installs	Libs	Version	Deleted	Other Sources	Libs	Comments
1	✓	✓	1.8.10				Standard Arduino Libraries
2	✓	✓	1.8.10				Standard Arduino Libraries
3	✓	✓	1.8.10				Standard Arduino Libraries
4	✓	✓	1.8.10				Standard Arduino Libraries
5	✓	✓	1.8.10				Standard Arduino Libraries
6	✓	✓	1.8.10				Standard Arduino Libraries
7	✓	✓	1.8.10				Standard Arduino Libraries
8	✓	✓	1.8.10				Standard Arduino Libraries
9	✓	✓	1.8.10				Standard Arduino Libraries
10	✓	✓	1.8.10				Standard Arduino Libraries
11	✓	✓	1.8.10				Standard Arduino Libraries
12	✓	✓	1.8.10				Standard Arduino Libraries
13	✓	✓	1.8.10				Standard Arduino Libraries
14	✓	✓	1.8.10				Standard Arduino Libraries
15	✓	✓	1.8.10				Standard Arduino Libraries
16	✓	✓	1.8.10				Standard Arduino Libraries
17	✓	✓	1.8.10				Standard Arduino Libraries
18	✓	✓	1.8.10				Standard Arduino Libraries
19	✓	✓	1.8.10				Standard Arduino Libraries
20	✓	✓	1.8.10				Standard Arduino Libraries
21	✓	✓	1.8.10				Standard Arduino Libraries
22	✓	✓	1.8.10				Standard Arduino Libraries
23	✓	✓	1.8.10				Standard Arduino Libraries
24	✓	✓	1.8.10				Standard Arduino Libraries
25	✓	✓	1.8.10				Standard Arduino Libraries
26	✓	✓	1.8.10				Standard Arduino Libraries
27	✓	✓	1.8.10				Standard Arduino Libraries
28	✓	✓	1.8.10				Standard Arduino Libraries
29	✓	✓	1.8.10				Standard Arduino Libraries
30	✓	✓	1.8.10				Standard Arduino Libraries
31	✓	✓	1.8.10				Standard Arduino Libraries
32	✓	✓	1.8.10				Standard Arduino Libraries
33	✓	✓	1.8.10				Standard Arduino Libraries
34	✓	✓	1.8.10				Standard Arduino Libraries
35	✓	✓	1.8.10				Standard Arduino Libraries
36	✓	✓	1.8.10				Standard Arduino Libraries
37	✓	✓	1.8.10				Standard Arduino Libraries
38	✓	✓	1.8.10				Standard Arduino Libraries
39	✓	✓	1.8.10				Standard Arduino Libraries
40	✓	✓	1.8.10				Standard Arduino Libraries
41	✓	✓	1.8.10				Standard Arduino Libraries
42	✓	✓	1.8.10				Standard Arduino Libraries
43	✓	✓	1.8.10				Standard Arduino Libraries
44	✓	✓	1.8.10				Standard Arduino Libraries
45	✓	✓	1.8.10				Standard Arduino Libraries
46	✓	✓	1.8.10				Standard Arduino Libraries
47	✓	✓	1.8.10				Standard Arduino Libraries
48	✓	✓	1.8.10				Standard Arduino Libraries
49	✓	✓	1.8.10				Standard Arduino Libraries
50	✓	✓	1.8.10				Standard Arduino Libraries
51	✓	✓	1.8.10				Standard Arduino Libraries
52	✓	✓	1.8.10				Standard Arduino Libraries
53	✓	✓	1.8.10				Standard Arduino Libraries
54	✓	✓	1.8.10				Standard Arduino Libraries
55	✓	✓	1.8.10				Standard Arduino Libraries
56	✓	✓	1.8.10				Standard Arduino Libraries
57	✓	✓	1.8.10				Standard Arduino Libraries
58	✓	✓	1.8.10				Standard Arduino Libraries
59	✓	✓	1.8.10				Standard Arduino Libraries
60	✓	✓	1.8.10				Standard Arduino Libraries
61	✓	✓	1.8.10				Standard Arduino Libraries
62	✓	✓	1.8.10				Standard Arduino Libraries
63	✓	✓	1.8.10				Standard Arduino Libraries
64	✓	✓	1.8.10				Standard Arduino Libraries
65	✓	✓	1.8.10				Standard Arduino Libraries
66	✓	✓	1.8.10				Standard Arduino Libraries
67	✓	✓	1.8.10				Standard Arduino Libraries
68	✓	✓	1.8.10				Standard Arduino Libraries
69	✓	✓	1.8.10				Standard Arduino Libraries
70	✓	✓	1.8.10				Standard Arduino Libraries
71	✓	✓	1.8.10				Standard Arduino Libraries
72	✓	✓	1.8.10				Standard Arduino Libraries
73	✓	✓	1.8.10				Standard Arduino Libraries
74	✓	✓	1.8.10				Standard Arduino Libraries
75	✓	✓	1.8.10				Standard Arduino Libraries
76	✓	✓	1.8.10				Standard Arduino Libraries
77	✓	✓	1.8.10				Standard Arduino Libraries
78	✓	✓	1.8.10				Standard Arduino Libraries
79	✓	✓	1.8.10				Standard Arduino Libraries
80	✓	✓	1.8.10				Standard Arduino Libraries
81	✓	✓	1.8.10				Standard Arduino Libraries
82	✓	✓	1.8.10				Standard Arduino Libraries
83	✓	✓	1.8.10				Standard Arduino Libraries
84	✓	✓	1.8.10				Standard Arduino Libraries
85	✓	✓	1.8.10				Standard Arduino Libraries
86	✓	✓	1.8.10				Standard Arduino Libraries
87	✓	✓	1.8.10				Standard Arduino Libraries
88	✓	✓	1.8.10				Standard Arduino Libraries
89	✓	✓	1.8.10				Standard Arduino Libraries
90	✓	✓	1.8.10				Standard Arduino Libraries
91	✓	✓	1.8.10				Standard Arduino Libraries
92	✓	✓	1.8.10				Standard Arduino Libraries
93	✓	✓	1.8.10				Standard Arduino Libraries
94	✓	✓	1.8.10				Standard Arduino Libraries
95	✓	✓	1.8.10				Standard Arduino Libraries
96	✓	✓	1.8.10				Standard Arduino Libraries
97	✓	✓	1.8.10				Standard Arduino Libraries
98	✓	✓	1.8.10				Standard Arduino Libraries
99	✓	✓	1.8.10				Standard Arduino Libraries
100	✓	✓	1.8.10				Standard Arduino Libraries

Dann werden einige Bibliotheken installiert. Anschließend dann der Raspberry ausgewählt werden.

Optionen und Spezielle Funktionen

LED Arduino

DCC Arduino

Dateien

Update

Bootloader

USB Port erkennen

Arduino Typ

☐ Automatisch erkennen
 ☐ Nano Normal (old Bootloader)
 ☐ Nano (neue Version)
 ☐ Nano (Full memory)

Für andere Hauptplatine

☐ Uno
 ☐ Typ von Arduino IDE benutzen
 ☐ ESP32 Wroom
 ☒ Raspberry Pico (Experimental)

Weitere Programme:

Pattern Configurator

Hilfe im Forum

LED Farbtest starten

(Update: Strg+Klick)

Schließen

Das Skript für diese Anleitung wurde uns von Theo zur Verfügung gestellt.
Gestaltung und Umsetzung von fromue

24.08.2025 by fromue

From:
<https://wiki.mobaledlib.de/> - MobaLedLib Wiki

Permanent link:
https://wiki.mobaledlib.de/anleitungen/bauanleitungen/ml010/lichtmaschine_pico-zero?rev=1756046397

Last update: 2025/08/24 14:39

