

# Zuordnung der Pins im Programm-Generator

Im ProgrammGenerator können in den Pin-Makros auch logische Pin-Namen verwendet werden.

- Vorteil 1: Man muss die Pin-Nummern nicht kennen
- Vorteil 2: Die Pin-Alias-Namen werden auf die jeweiligen Pin-Nummern der verwendeten Hardware umgewandelt,

eine ProgGenerator Konfiguration funktioniert somit ohne Änderung für Arduino und ESP32.

**Makroauswahl:** Filter:

- Licht**
  - Belehtes Haus
  - Straßenbeleuchtung
  - Lichteffekte**
  - Signale**
  - Farbeinstellungen**
- Dynamik**
- Sound**
- Schalten**
- Erweiterungen**
- Konfiguration**
  - Letzte Zustände speichern
  - LEDs der Hauptplatine steuern
  - Pinnummern**
    - Pins Schalter Gruppe A definieren
    - Pins Schalter Gruppe B definieren
    - Pins Schalter Gruppe C definieren
    - Pins Schalter Gruppe D definieren
    - Pins Takt-Leitung definieren
    - Pins Reset-Leitung definieren
    - Pin Fotowiderstand definieren
    - Pins LED Bus definieren
- Tag/Nacht**

**Der Einstieg in die MobaLedLib**  
Mit dieser Funktion wird ein „belebtes“ Haus nachgebildet. Simuliert das Einschaltverhalten und Flackern von gasbetriebenen Straßenlaternen.  
**Vorgefertigte Muster für zahlreiche Lichteffekte.**  
**Lichtsignale mit Single- oder RGB-LEDs**  
**Vom Standard abweichende Farbeinstellungen**  
**Bewegung von Körpern in ihrer Abhängigkeit von den einwirkenden Kräften**  
**Beschallung der Modellbahn mit Soundmodulen**  
**Schalten, Automatisieren, Verknüpfen**  
**ESP32, DMX512**  
**Änderungen am Arduino-Setup**  
Aktiviert die Speicherung der letzten Zustände bei DCC, Selectrix und CAN Kommandos.  
Ansteuerung der LEDs auf der Hauptplatine über DCC, Selectrix, CAN oder Variablen.

**Definition der verwendeten Eingangs-Pins**  
Definiert die verwendeten Eingangs Pins für die analogen Schalter (Gruppe A)  
Definiert die verwendeten Eingangs Pins für die Schalter der Gruppe B (Border).  
Definiert die verwendeten Eingangs Pins für die Schalter der Gruppe C (Console).  
Definiert die verwendeten Eingangs Pins für die Schalter der Gruppe D (Direkt auf Hauptplatine).  
Definiert den verwendeten Eingangs Pin der Takt Leitung für die PushButton\_4017 Platinen.  
Definiert den verwendeten Eingangs Pin der Reset Leitung für die PushButton\_4017 Platinen.  
Definiert den Verwendeten Eingangs Pin für den Helligkeitssensor  
Definiert die Ausgangs Pins zur Ansteuerung der LEDs

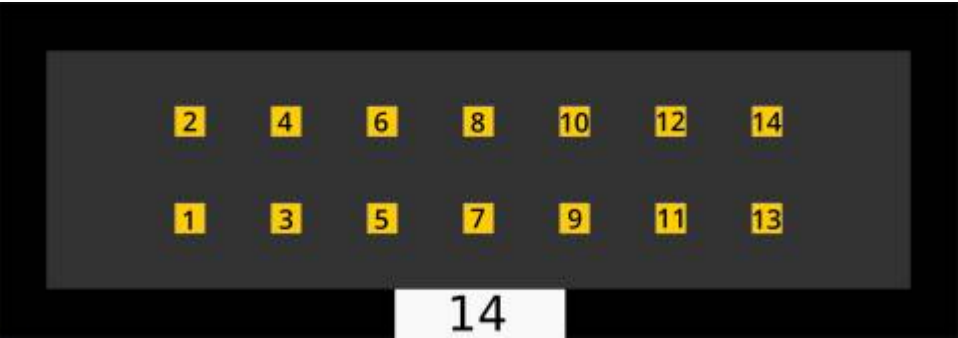
**Beeinflussung des LDR und der Uhrzeit**

## Hier eine Liste der verfügbaren Pin-Alias Namen

### Led-Bus-Pins

LED-Bus	Arduino-Pin LED-Nano	ESP32
LED0 (Haupt-LED-Bus, der über den Wannenstecker unten rechts ausgegeben wird)	D6	27
LED1 ( <a href="#">PushButtons</a> )	A4	32
LED2 ( <a href="#">Nur ESP32</a> )		16
LED3 ( <a href="#">Nur ESP32</a> )		14
LED4 ( <a href="#">Nur ESP32</a> )		18
LED5 ( <a href="#">Nur ESP32</a> )		19
LED6 ( <a href="#">Nur ESP32</a> )		23
LED7 ( <a href="#">Nur ESP32 mit 38 Pins</a> )		0
LED8 DMX oder LED Kanal	2	17

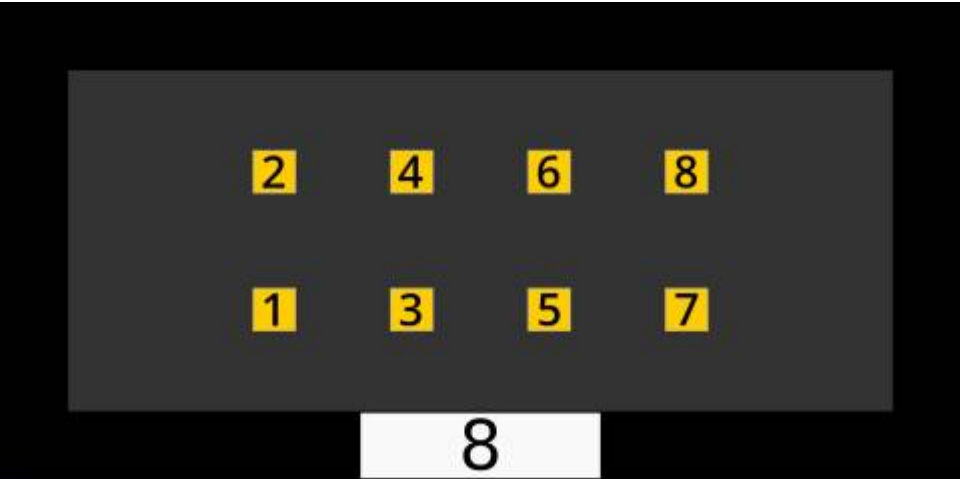
## Stecker Key 80



Pin-Nummer Wannenstecker	Name	alternative Namen	Arduino-Pin LED-Nano	ESP32 GPIO
1	KEY80_P1		D2	17
2	KEY80_P2		D7	12
3	KEY80_P3		D8	26
4	KEY80_P4		D9	25
5	KEY80_P5		D10	5
6	KEY80_P6		D11	23
7	KEY80_P7		D12	19
8	KEY80_P8		A1	4
9	KEY80_P9	KEY80_BUTTONS	A2	34
10	KEY80_P10	KEY80_RESET_K KEY80_RST	A3	33
11	KEY80_P11	KEY80_LEDS_K KEY80_SDA	A4	32
12	KEY80_P12	KEY80_SCL	A5	39
13	-		5V	
14	-		GND	

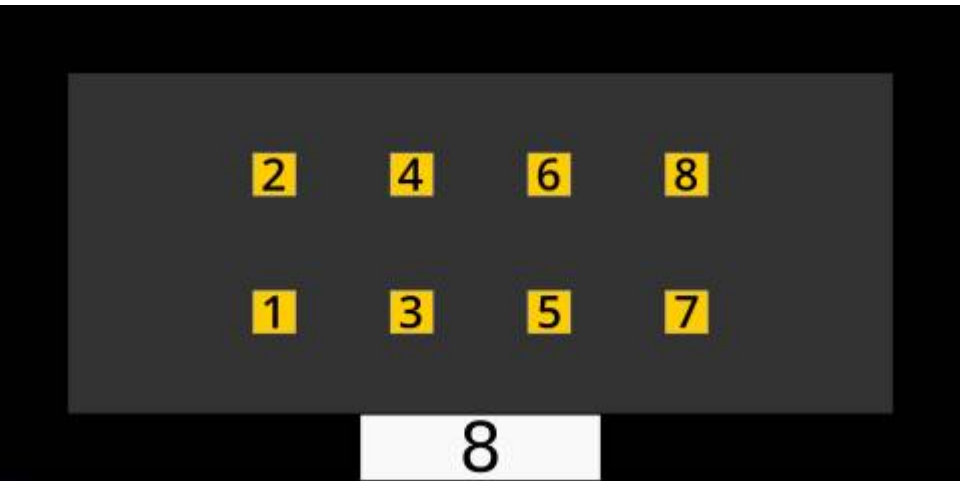
---

## Stecker Keybrd



Pin-Nummer Wannenstecker	Name	alternative Namen	Arduino-Pin LED-Nano	ESP32 GPIO
1	KEYBRD_P1	KEYBRD_LEDS_K	A4	32
2			A5	39
3			5V	
4			GND	
5	KEYBRD_P5	KEYBRD_CLOCK_K KEYBRD_CLK	A0	15
6	KEYBRD_P6	KEYBRD_RESET_K KEYBRD_RST	A3	33
7	KEYBRD_P7	KEYBRD_BUTTONS	A2	34
8			A7 (Jumper „KEY_A7“ schließen) Deaktiviert Tag/Nacht Schaltung	35

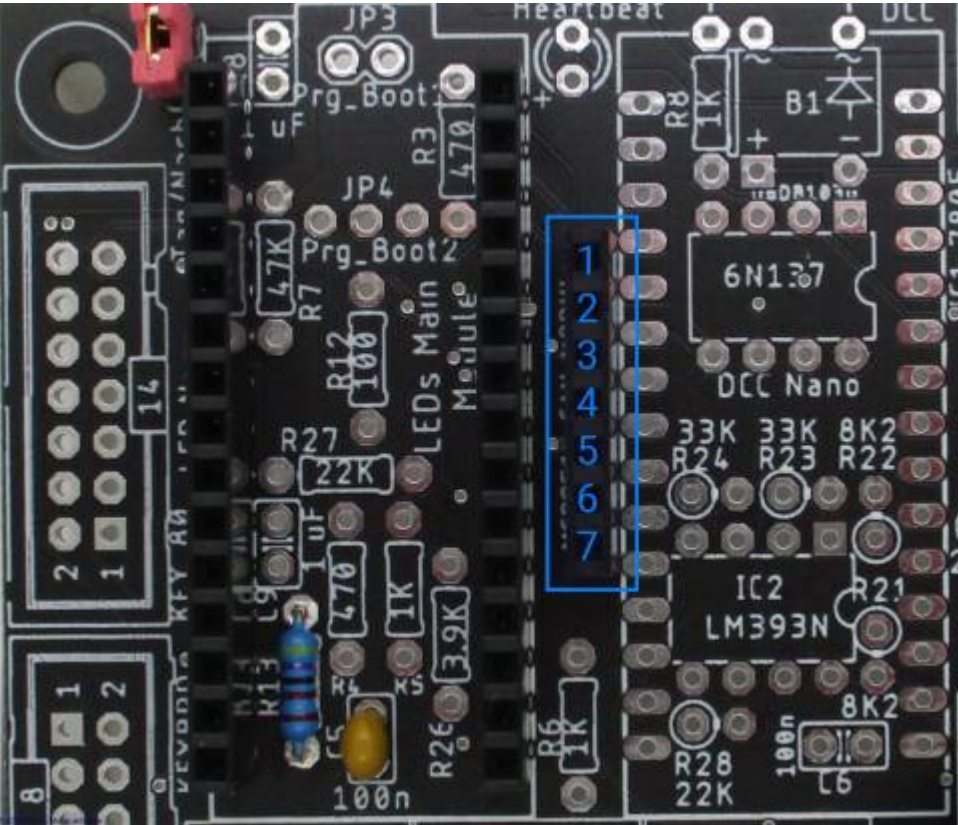
Stecker KeyBrd1



Pin-Nummer Wannenstecker	Name	alternative Namen	Arduino-Pin LED-Nano	ESP32 GPIO
6	KEYBRD1_P6	KEYBRD1_RESET_K KEYBRD1_RST	A3	33
7	KEYBRD1_P7	KEYBRD1_BUTTONS	A2	34

Pin-Nummer Wannenstecker	Name	alternative Namen	Arduino-Pin LED-Nano	ESP32 GPIO
8	KEYBRD1_P8		A6 (Jumper „SJ4“ schließen)	36

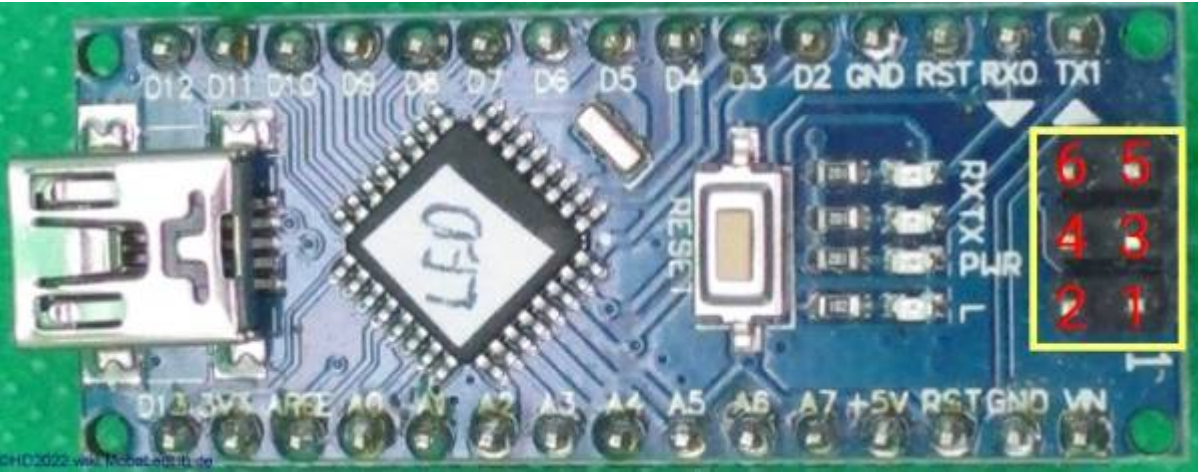
Stecker CAN-Modul



Pin-Nummer Wannenstecker	Name	alternative Namen	Arduino-Pin LED-Nano	ESP32 GPIO
3	CAN_P3	CAN_CS	D10	5
4	CAN_P4	CAN_SO	D12	19
5	CAN_P5	CAN_SI	D11	23
6	CAN_P6	CAN_CLK CAN_SCK	D13	18

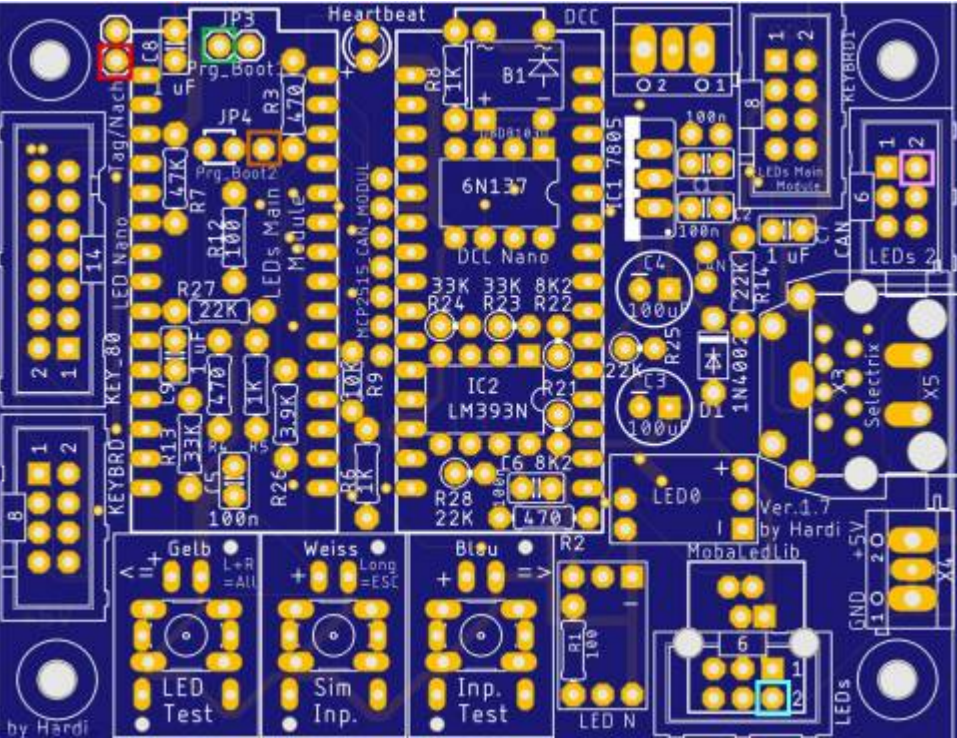
Boot Stecker am LED Arduino





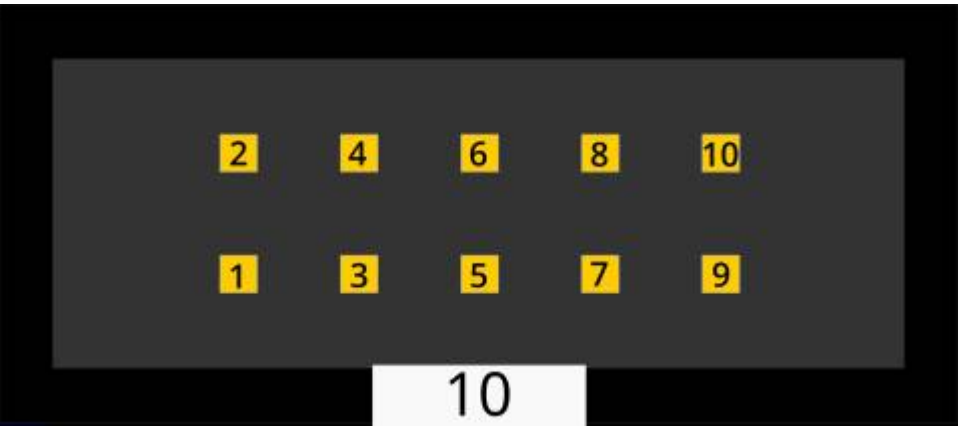
Pin-Nummer Wannenstecker	Name	alternative Namen	Arduino-Pin LED-Nano
1	BOOT_P1	BOOT_SO	D12
3	BOOT_P3	BOOT_SCK	D13
4	BOOT_P4	BOOT_SI	D11

Sonstige Stecker/Jumper auf der Hauptplatine



farbige Markierung am Bild	Name	alternative Namen	Arduino-Pin LED-Nano	ESP32 GPIO
rot	LDR	J1_P2	A7	35
grün	JP3_P2		D13	18
orange	JP4_P3		D10	5
rosa	SV1_P2		D2 (Jumper „LED_BUS2B“ schließen)	17
türkis	SV3_P2		D6 über R1	27

### Extension Stecker am ESP32 Adapter



Pin-Nummer Wannenstecker	Name	alternative Namen	GPIO vom ESP32 30 Pin		GPIO vom ESP32 38 Pin
1	MLL_EXT_OUT_P1	MLL_EXT_OUT_SCL	D22		
2	MLL_EXT_OUT_P2	MLL_EXT_OUT_SDA	D21		
5	MLL_EXT_OUT_P5	LED2	D16 (RX2)		D16
6	MLL_EXT_OUT_P6	LED3	D14		
7	MLL_EXT_OUT_P7	LED4	D18		
8	MLL_EXT_OUT_P8	LED5	D19		
9	MLL_EXT_OUT_P9	LED6	D23		
10	MLL_EXT_OUT_P10	LED7	—		D0

From:

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

Permanent link:

[https://wiki.mobaledlib.de/anleitungen/hilfestellungen/zuordnung\\_der\\_pins](https://wiki.mobaledlib.de/anleitungen/hilfestellungen/zuordnung_der_pins)

Last update:

2024/01/11 20:15

