

# Der Simulator des Programm Generators

Trotz intensivem Studium des Wikis folgt man oft dem „Trial and error“-Prinzip beim Erlernen neuer Programmierungen. Dabei können die Upload-Zeiten zum Arduino nervenzehrend werden, wenn es mal nicht auf Anhieb klappt.

Hinzu kommt, dass der Arduino die Daten nicht unendlich oft speichern kann. Der Flash-Speicher hat eine begrenzte Anzahl von Schreibzyklen, typischerweise 100.000. Das zu erreichen, liegt in weiter Ferne, aber es ist unnötig.

Mit dem im [März 2022](#) von Jürgen vorgestellten Simulator kann man die Programmierung nämlich schon vorab testen, indem man die Programmierung mit gedrückter Shift-Taste zum Arduino überträgt.

Aktiv	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Icon	Name
✓		1	AnAus	0					Belebtes Haus
✓		2	AnAus	0					Baustellenlicht 6x
✓		3	AnAus	0					Andreaskreuz RGB
✓		4	AnAus	0					Blaulich 1
✓		4	AnAus	0					Blaulich 2

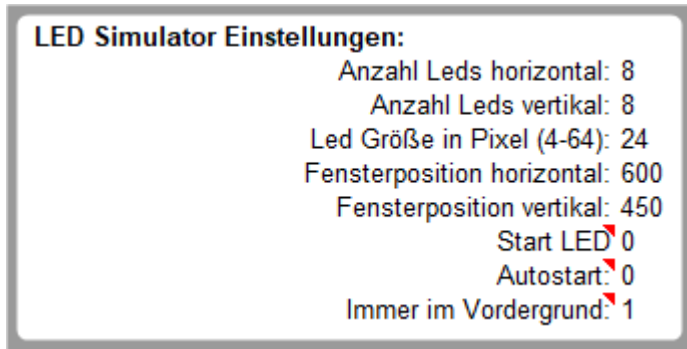
Der Simulator setzt aktuell einen Arduino voraus, den er simulieren soll. Von daher muss in der entsprechenden Excel-Mappe auch ein Arduino als Empfänger eingetragen sein. Ist in den Einstellungen ein ESP32 oder ein Raspi ausgewählt, funktioniert der Simulator nicht.



Es empfiehlt sich daher, neue Programmierungen in einer separaten Excel-Mappe aufzubauen und zu testen. Bei erfolgreichem Ablauf lassen sich die Zeilen ganz leicht in die Hauptprogrammierung kopieren oder per Include-Befehl einbinden. Alternativ kann man das selbstverständlich auch vor und nach dem Testen in den Einstellungen anpassen.

## Die Konfiguration

Über die Konfiguration (Excel-Mappe „Config“) lässt sich das Aussehen und das Verhalten des Simulators beeinflussen.



## Anzahl der LEDs horizontal/vertikal

Hier lässt sich die Größe der virtuellen Matrix einstellen. Ab Werk ist hier die gängige 64er-Matrix mit 8 x 8 LEDs dargestellt.

## LED-Größe in Pixel (4-64)

Je nach Größe des Monitors, der Menge der dargestellten Pixel oder den persönlichen Vorlieben, lassen sich die LEDs der virtuellen Matrix in ihrer Größe variieren. Ab Werk ist hier eine Größe von 24 x 24 Pixeln eingestellt.

## Fensterposition horizontal/vertikal

Mit diesen beiden Werten kann die exakte Position vorgegeben werden, in der der Simulator beim Starten erscheinen soll. Praktisch ist beispielsweise die Ecke unten rechts, damit man Schalter und Programmierung im Hintergrund noch erkennen kann. Will man einen Screencast aufnehmen, eignet sich eher ein Bereich in der Nähe der Programmierung. Die Fensterposition bezieht sich immer auf die Ecke oben links des Fensters. Ändert man die Anzahl der dargestellten LEDs, sieht man diese unter Umständen nicht alle.

## Start LED

Gerade bei umfangreichen Programmierungen reicht die virtuelle 64er Matrix nicht mehr aus. Um beim Test der LED Nr. 200 nicht alle 199 vorangegangenen LEDs simulieren zu müssen, kann man vor dem Aufrufen des Simulators die Start-LED definieren. Diese Funktion eignet sich auch, wenn man beispielsweise die Heartbeat-LED nicht mit simulieren möchte.

## Autostart

Wer den Simulator bei jedem Start automatisch aktivieren möchte, kann das hier einstellen. Folgende Optionen stehen zur Verfügung: 0 = Simulator beim Start ausgeschaltet 1 = Simulator wird beim Programm-Start geöffnet 2 = Simulator und Hauptplatine gemeinsam (die Programmierung des Arduinos wird parallel gestartet) 3 = Nur Simulator

## Immer im Vordergrund

Gerade wenn DCC-Schalter zum Test benötigt werden, empfiehlt es sich, das Fenster des Simulators im Vordergrund zu behalten, weil es sonst beim Klick auf die Excel-Tabelle dahinter verschwinden würde.

Diese Einstellung ist ab Werk aktiviert und lässt sich bei Bedarf deaktivieren.

## Die Ansicht

### Darstellung der Mischfarben

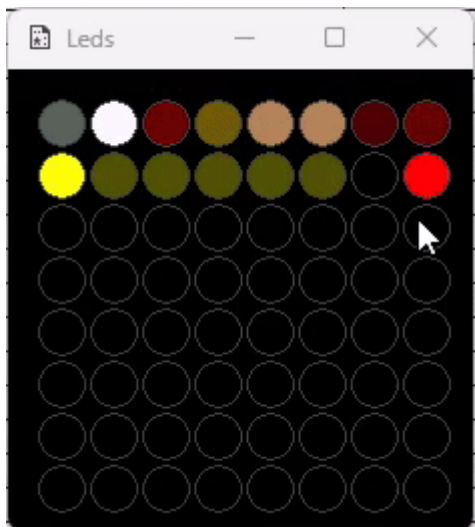
Nicht immer will man RGB LEDs simulieren. Wer beispielsweise Relais schalten will, muss wissen, wann ein bestimmter Kanal (RGB) aktiviert wird.

Per Doppelklick auf eine der dargestellten virtuellen LEDs kann zwischen drei Ansichten gewechselt werden.

**Ansicht 1:** RGB LED stellt die Mischfarbe dar

**Ansicht 2:** Drei Einzel-LEDs stellen die Kanäle Rot, Grün und Blau dar

**Ansicht 3:** Kombiniert die Ansicht 1 & 2

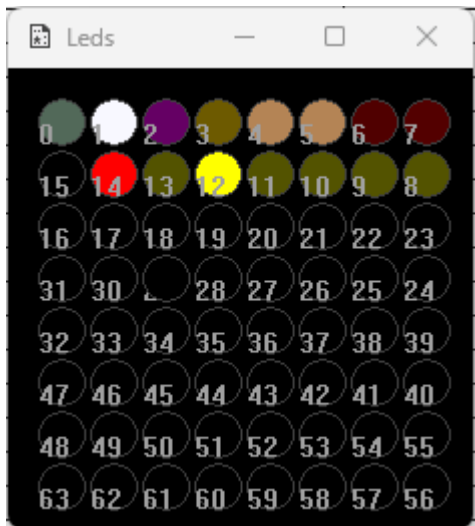


### Adresse der einzelnen LED

Manchmal kann es hilfreich sein, die Nummer der dargestellten LED zu kennen, wenn man beispielsweise das korrekte Schalten einer gezielten LED überprüfen möchte.

Je nach gewählter Start-Nummer, zeigt der Simulator die fortlaufende Nummer der LEDs an. Im gezeigten Beispiel ist die Start LED die Nummer 0.

Die Anzeige der Nummerierung lässt sich mit der Taste „N“ aktivieren und deaktivieren.

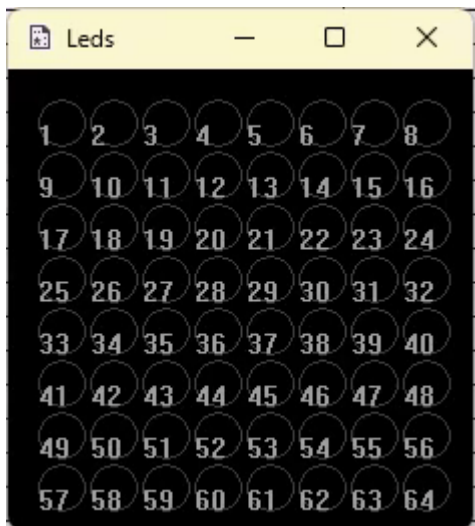


### Reihenfolge der LEDs

Bei der echten 64er Matrix gibt es unterschiedliche Varianten. Bei einer sind die LEDs in Form eines „S“ miteinander verbunden, sodass das Datensignal in der ersten Zeile von links nach rechts und in der zweiten Zeile von rechts nach links läuft (S-Anordnung). Bei der anderen läuft das Signal immer von links nach rechts und wird zwischen den Zeilen mit einer langen Drahtbrücke wieder nach vorn geführt (Z-Anordnung).

Wer ein Projekt auf einer Matrix umsetzen will und dieses vorher simulieren will, muss die richtige Reihenfolge darstellen.

Mit den Tasten „S“ und „Z“ wird zwischen den Verkabelungs-Varianten gewechselt.



From:

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

Permanent link:

[https://wiki.mobaledlib.de/anleitungen/prog\\_gen/simulator?rev=1750352902](https://wiki.mobaledlib.de/anleitungen/prog_gen/simulator?rev=1750352902)

Last update: 2025/06/19 17:08

