

Anleitung für den ersten Schnelleinstieg



Die MobaLedLib unterstützt im aktuellen Release als LED-Controller nur den **Arduino Nano mit einem AtMega328P** und den **ESP32**.

Die Verwendung eines **Raspberry Pico 2040** befindet sich noch in der Entwicklung. Den Arduino Nano mit einem **AtMega168** kann man ausschließlich für den DCC-Arduino verwenden, auf Grund des beschränkten Speicherplatzes mit nur 16KB im Gegensatz zum AtMega328P mit 32KB. [weitere Infos](#)

Installation ARDUINO IDE

Los geht's

Als Einstieg und zum Ausprobieren reichen

- ein Arduino (Original oder preiswerter Clone)
- ein paar WS2812 RGB-LEDs und
- drei Kabel

völlig aus. Der Arduino kann ein Uno, Nano oder auch ein Mini Pro o.ä. sein.

Diese Anleitung gilt für eine WINDOWS 10 Umgebung mit einem installierten Microsoft Office/Excel. MS Office Alternativen wie OpenOffice oder LibreOffice werden zur Zeit nicht unterstützt.

Tipp: Um diese Anleitung und das aktuelle Installationsfenster gleichzeitig auf dem Bildschirm zu sehen, kann man den Bildschirm teilen. Damit lässt sich ein Ausdruck vermeiden. Eine Anleitung dazu befindet sich am Ende der Anleitung unter [Tipp](#).

Zunächst muss man die aktuelle Arduino Entwicklungsumgebung IDE (Integrated Development Environment) herunterladen und installieren.

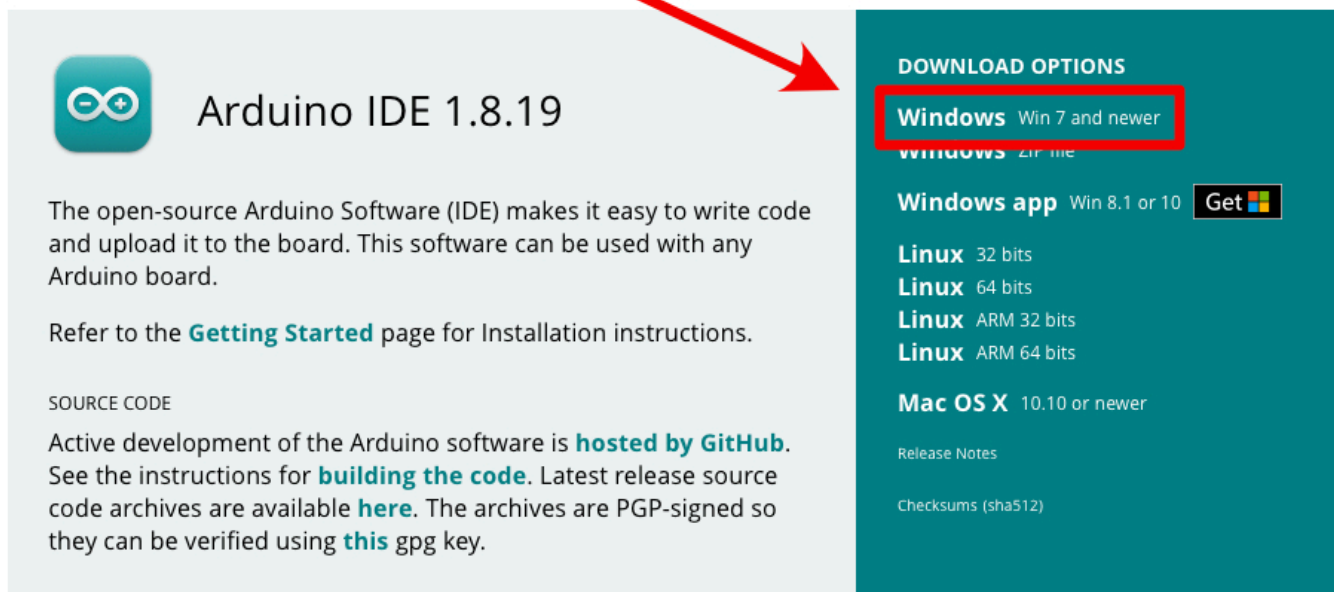
Die ARDUINO Download Seite erreicht man über den Link: <https://www.arduino.cc/en/Main/Software>


Die MobaLedLib läuft mit allen aktuellen Versionen der Arduino IDE und wurde mit den Versionen 1.8.8 - 1.8.19 erfolgreich getestet. Bei der Verwendung der Arduino IDE in den Versionen 1.8.8 und 1.8.9 kann es zu Problemen mit der Bibliothek „FastLED“ und einer veralteten Version der Arduino Boards kommen. In diesem Fall bitte die folgenden [Problemlösung](#) lesen.

Für die Unterstützung aller Funktionen der MobaLedLib empfehlen wir die aktuelle Version 1.8.19 zu installieren.

Der Arduino WEB-Editor oder die Arduino App können für die MobaLedLib nicht genutzt werden.

Legacy IDE (1.8.X)



 **Arduino IDE 1.8.19**


The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

- Windows** Win 7 and newer
- Windows ZIP file
- Windows app** Win 8.1 or 10 
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM 32 bits
- Linux** ARM 64 bits
- Mac OS X** 10.10 or newer


[Release Notes](#)

[Checksums \(sha512\)](#)

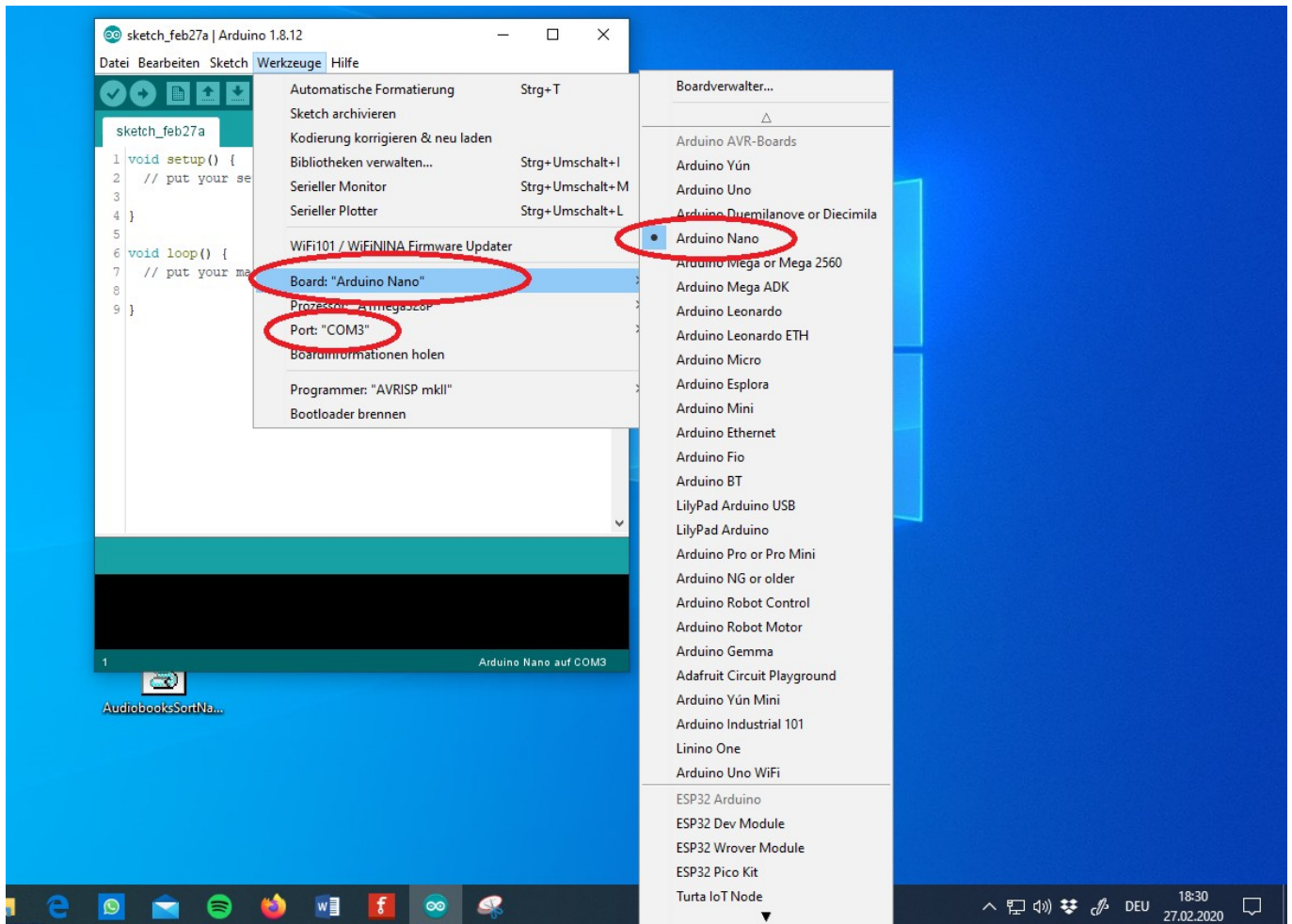
Anschließend wird man aufgefordert, für das Arduino Projekt zu spenden. Es ist gut, wenn man das großartige Projekt auf diese Weise unterstützt. Das Programm kann aber problemlos auch ohne Spenden heruntergeladen werden. Das heruntergeladene Programm, hier „arduino-1.8.19-windows.exe“ findet man in dem „Downloads“-Ordner seines PC. Die Position des Ordners hängt vom verwendeten Internet Browser und dessen Einstellungen ab.

Zur Installation wird das Programm mit einem Doppelklick gestartet. Anschließend befolgt man die Anweisungen auf dem Bildschirm. Zur Installation der IDE findet man bei Bedarf im Internet ausführliche Anleitungen, zum Beispiel:

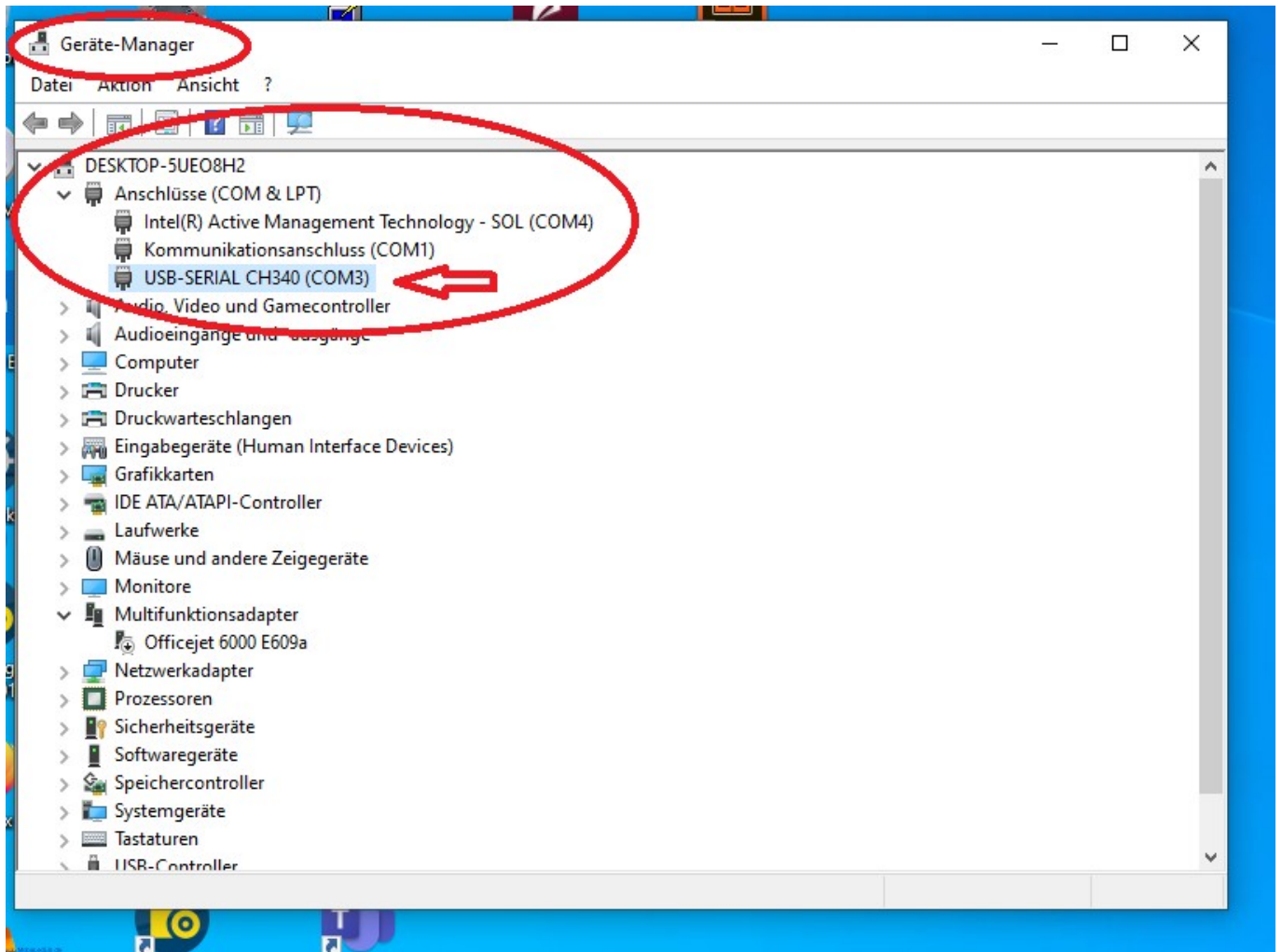
<https://draeger-it.blog/arduino-ide-installieren/?cn-reloaded=1&cn-reloaded=1>

Bei der Installation wird auf dem Desktop eine ICON angelegt, über das die Entwicklungsumgebung gestartet werden kann. 

Nun den Arduino an einen freien USB-Port anschließen. WINDOWS installiert nun bei erstmaliger Verbindung USB/Geräte-Treiber für den Arduino. Bei fehlerfreier Installation sollten COM-Port und Arduino, wie im folgenden Bild dargestellt, aufrufbar sein. Die Nummer des COM-Ports ist abhängig von der Belegung der Schnittstellen des PC mit anderen Geräten.



Falls das nicht der Fall ist, sollte zunächst überprüft werden, ob der Arduino erkannt wird und an welchem COM-Port er angeschlossen wurde. Dazu im WINDOWS Geräte-Manager, zu erreichen über die WINDOWS Starttaste mit rechter Maustaste anklicken, unter „Anschlüsse(COM&LPT)“ prüfen ob der Arduino dort aufgelistet ist.



Im Falle von original Arduinos erscheint der Namen z.B. NANO. Clones werden hingegen mit dem verbauten Chip aufgeführt, hier der CH340. Das liegt daran, dass dort statt der FTDI-Chips die deutlich preiswerteren CH340G USB 2 Serial Chips verbaut werden.

Werden im Gerätemanager weder der NANO noch der CH340 angezeigt sondern nur ein gelber Warnhinweis, ist der notwendige Treiber möglicherweise nicht installiert.

In dem Fall wie auch bei sonstigen Problemen: Hilfe, mein Arduino wird nicht erkannt. → siehe Abschnitt [Fehlerbehebung](#).

Tritt beim Kompilieren ein Fehler auf, helfen vielleicht [FAQ](#) weiter.

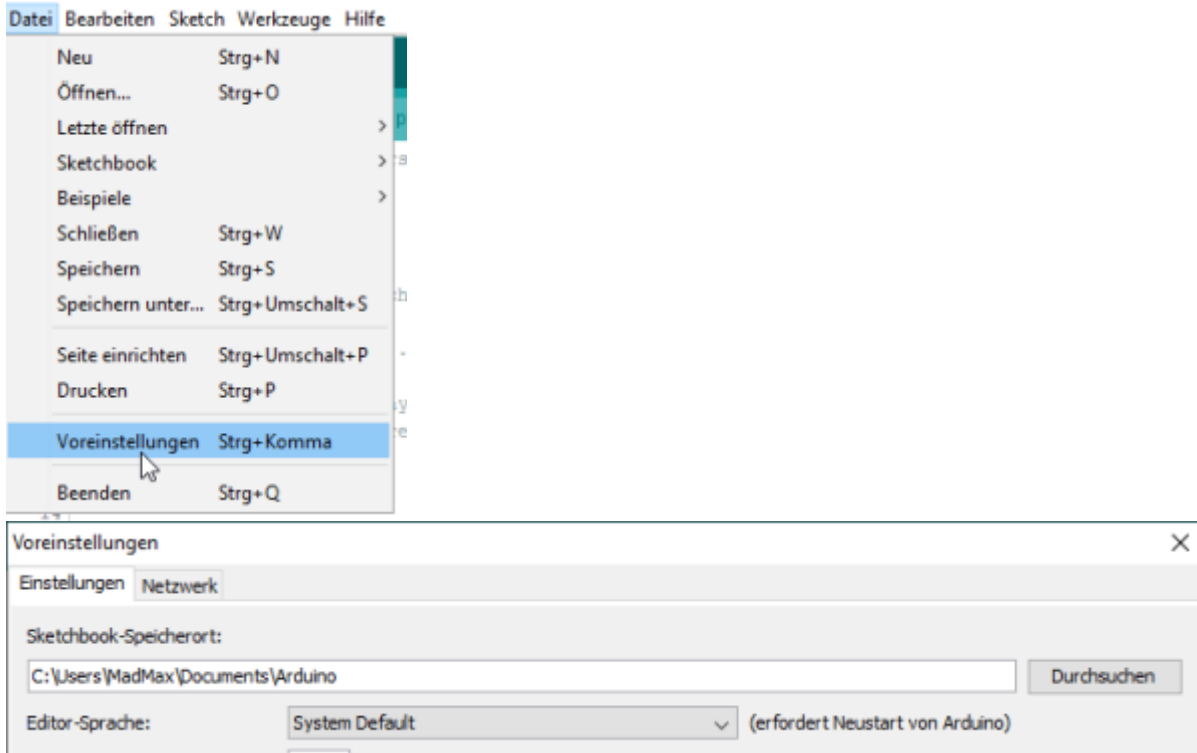
Für Einsteiger ist möglicherweise ein kleiner zusätzlicher Test interessant. Mehr dazu am Ende der Anleitung unter [Test](#).

Überprüfung Sketch-Ordner

Damit alle Funktionen der MobaLedLib aufgerufen werden können, muss der Pfad zum Sketch-Ordner auf dem Standard-Pfad stehen.

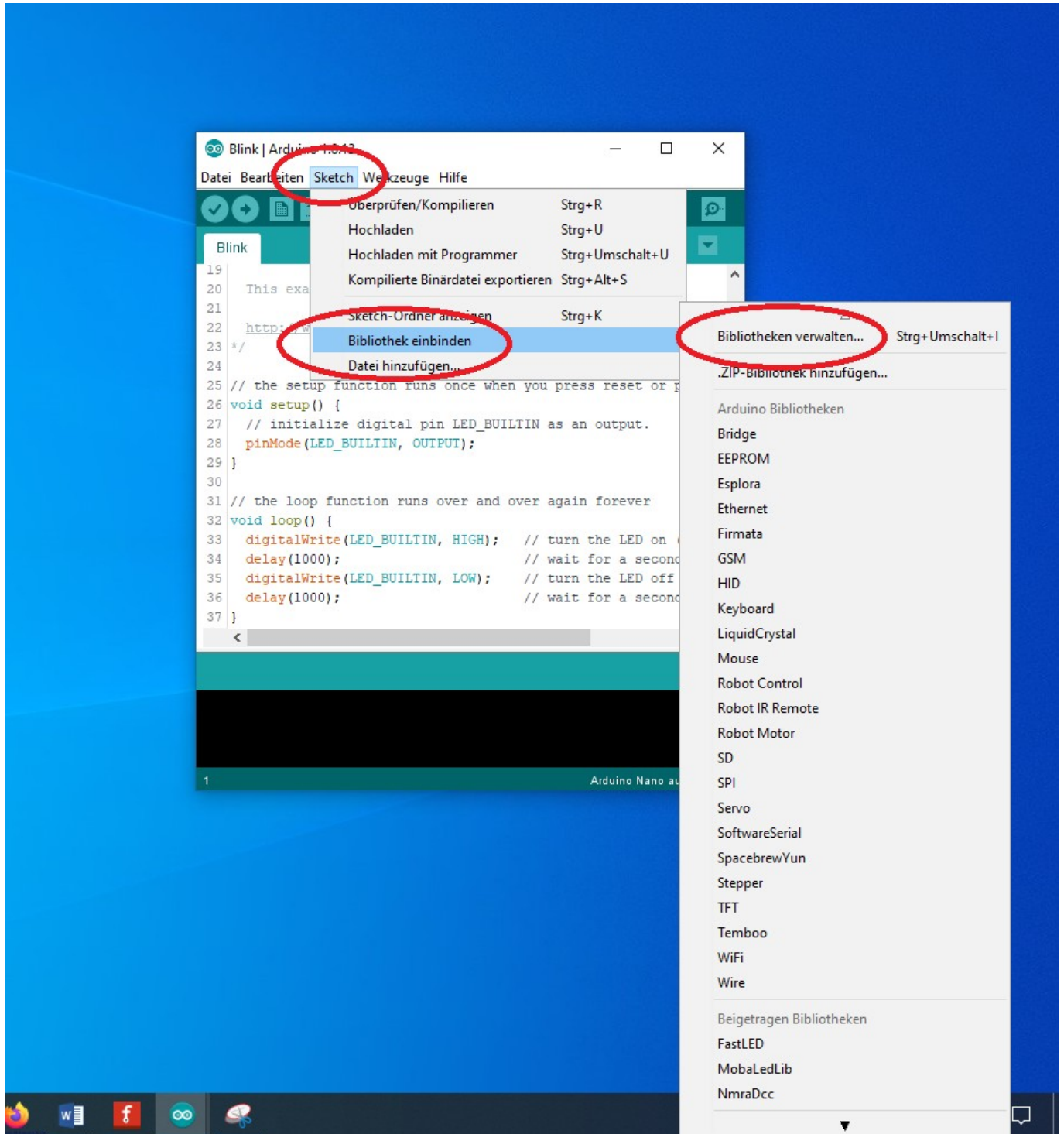
```
C:\Users\**BENUTZERNAME**\Documents\Arduino
```

Dies kann über das Menü innerhalb der Arduino IDE gemacht werden.

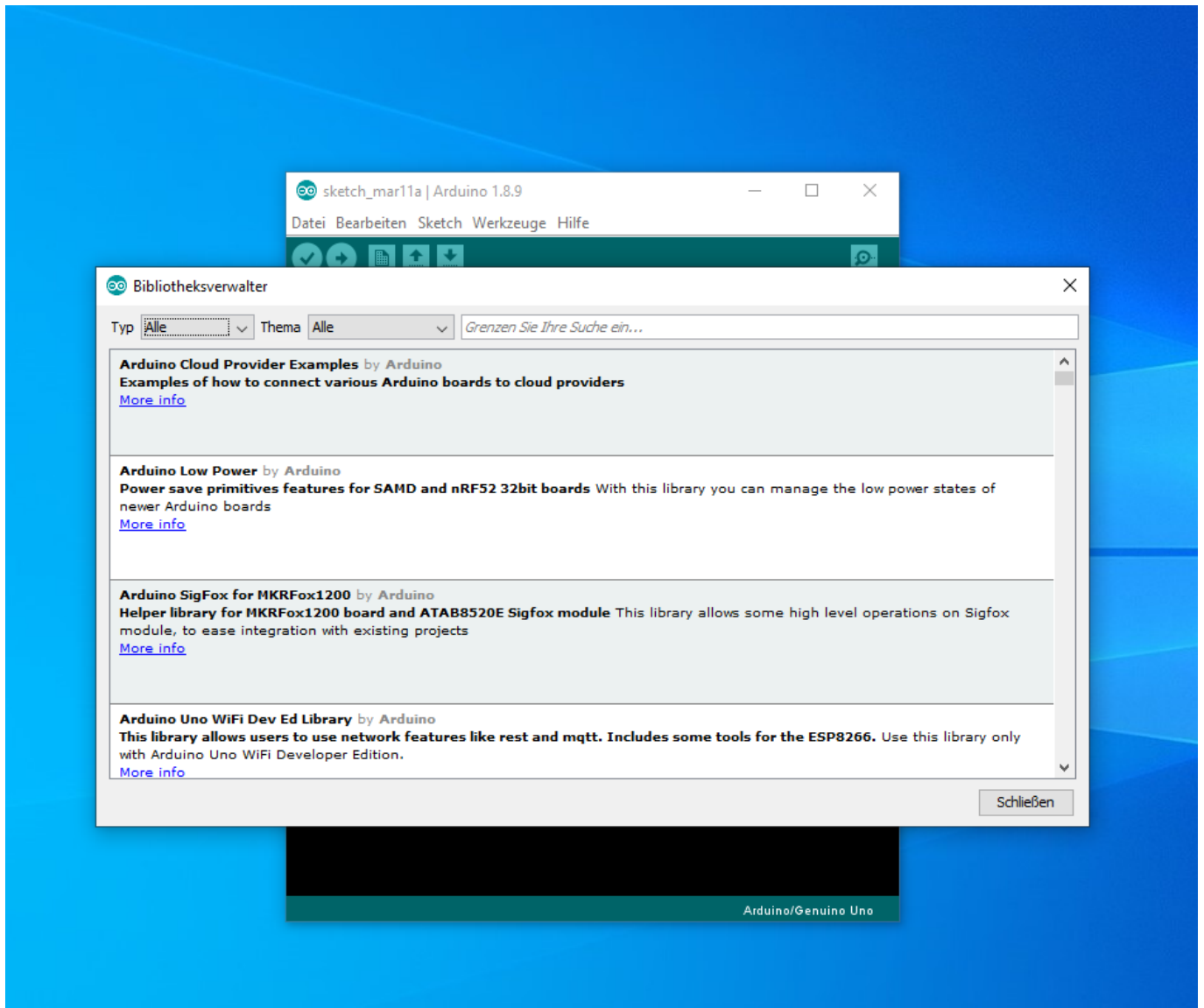


MobaLedLib (MLL) einrichten

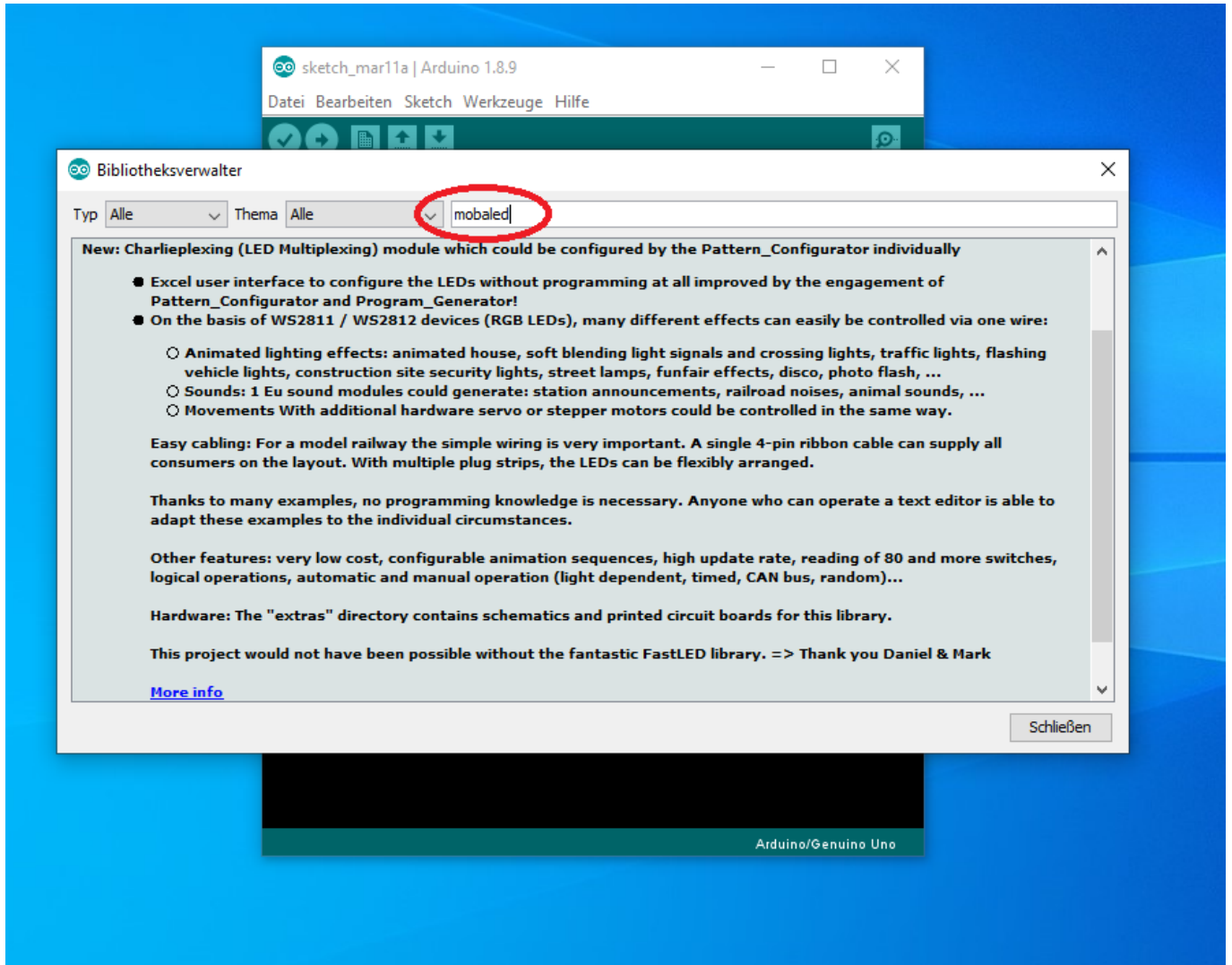
Zunächst muss die Bibliotheksverwaltung der IDE über Sketch → Bibliothek einbinden → Bibliothek verwalten geöffnet werden.



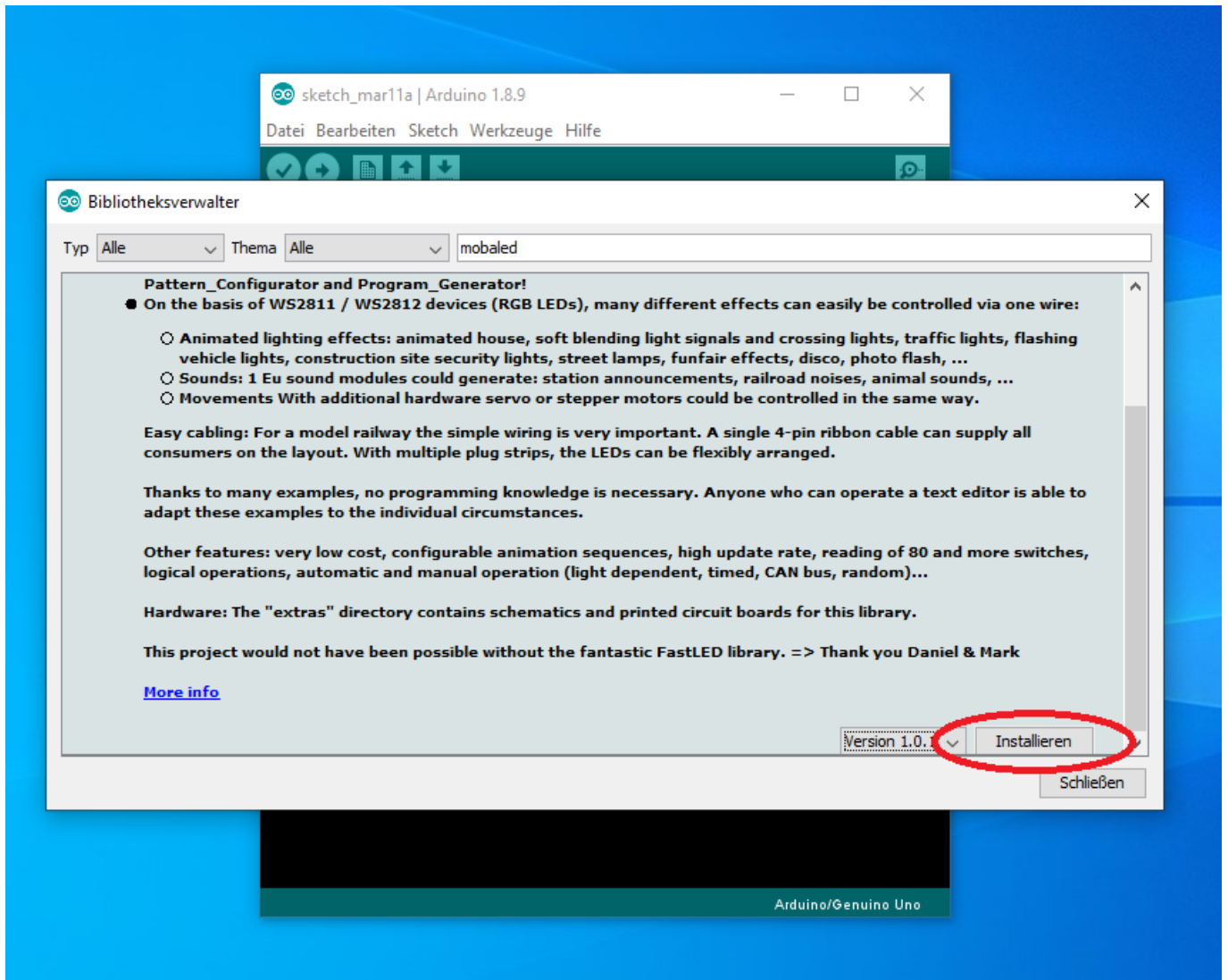
Es öffnet sich der Bibliotheksverwalter.



Nun in das Suchfeld rechts oben „mobaLedlib“ eingeben.

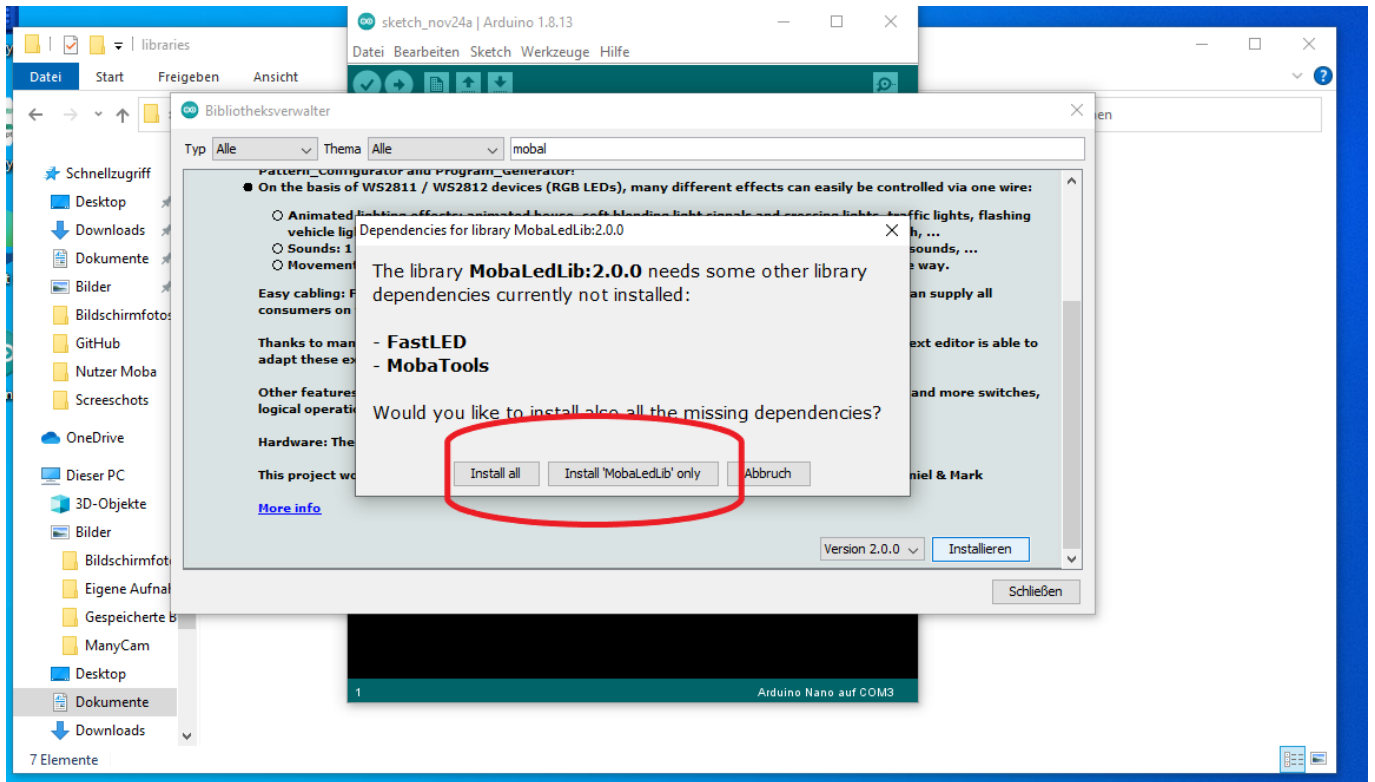


In den gefundenen Eintrag klicken, dann erscheint der Installieren-Knopf



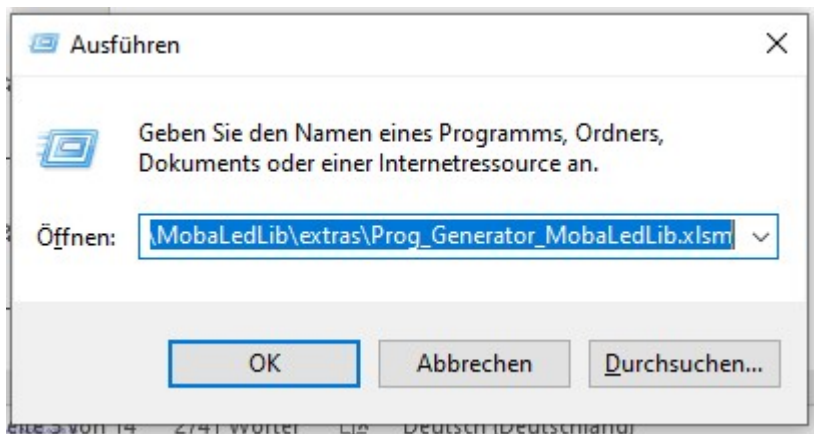
Im Bedarfsfall kann über diesen Menüpunkt auch eine andere Version oder ein Update installiert werden.

Mit einem Klick auf den Knopf wird die Installation angestoßen. Im folgenden Fenster kann man noch entscheiden ob alle möglicherweise fehlenden Libraries zum Betrieb der MobaLEDLib ebenfalls automatisch installiert werden sollen.



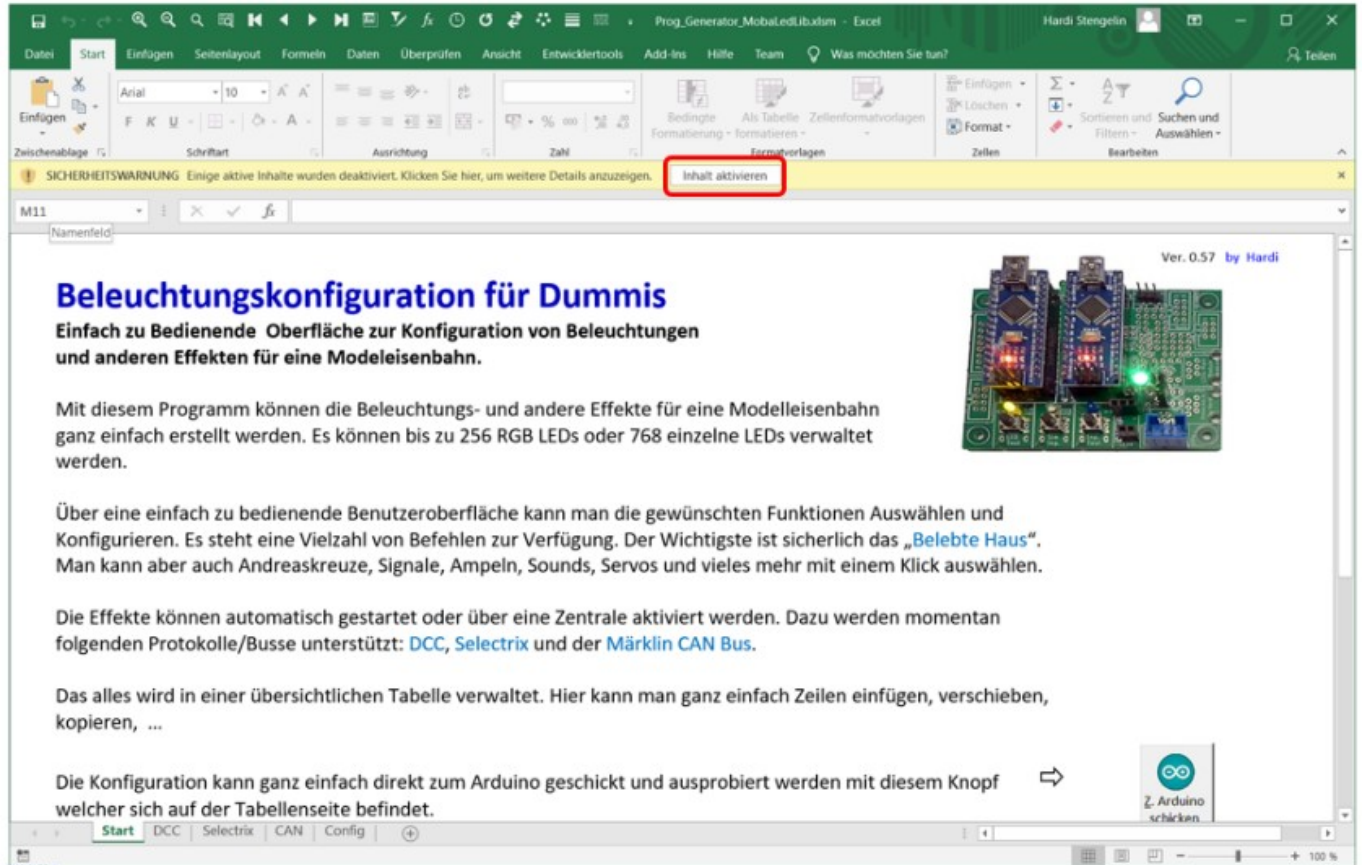
Jetzt kann endlich das MobaLedLib-Excel Programm geöffnet werden. Dazu auf der Tastatur die WINDOWS und die „R“ Taste gleichzeitig drücken. In den erscheinenden „Ausführen“ Dialog die folgende Zeile kopieren:

```
%USERPROFILE%\Documents\Arduino\libraries\MobaLedLib\extras\Prog_Generator_MobaLedLib.xlsm
```



Nun sollte sich endlich die aktuelle Version des Programm Generators der MobaLedLib öffnen. Sollte dies nicht der Fall sein, hat der Virens Scanner die MobaLedLib blockiert.

→ siehe Abschnitt [Fehlerbehebung](#).

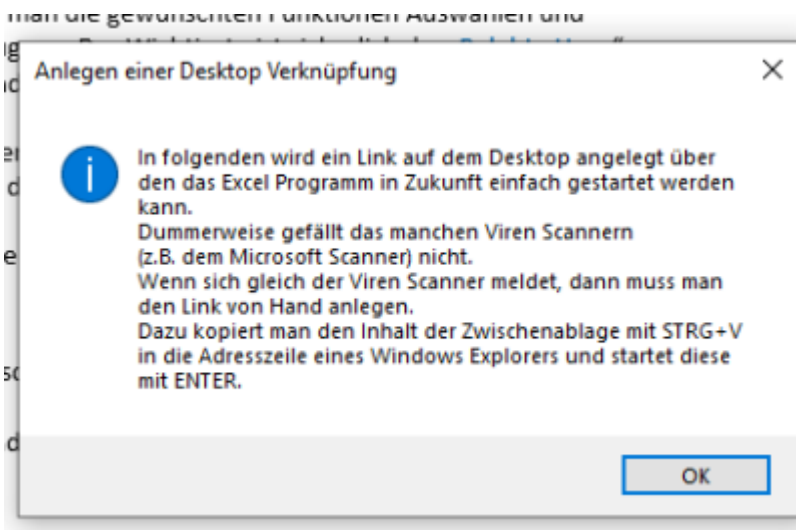


Da es sich um ein Excel Programm mit eingebetteten Makros handelt, wird beim ersten Start eine Sicherheitswarnung angezeigt. Mit einem Klick auf „Inhalte aktivieren“ werden die Makros ausführbar gemacht.

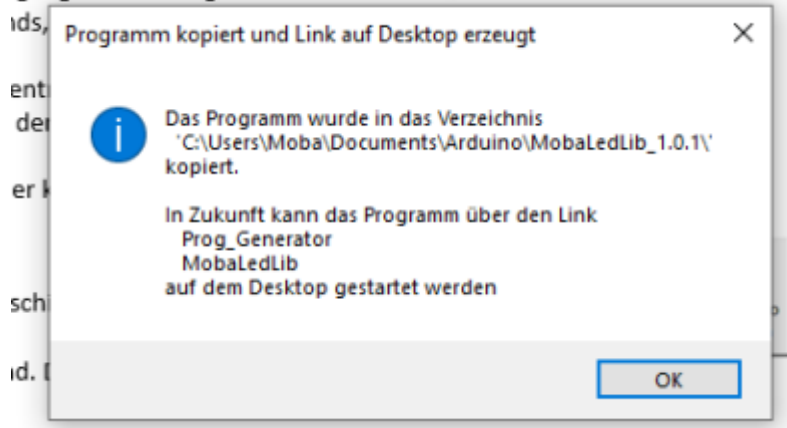
Danach wird das Programm automatisch in das Verzeichnis

```
%USERPROFILE%\Documents\Arduino\MobaLedLib_x.y.z\LEDs_AutoProg
```

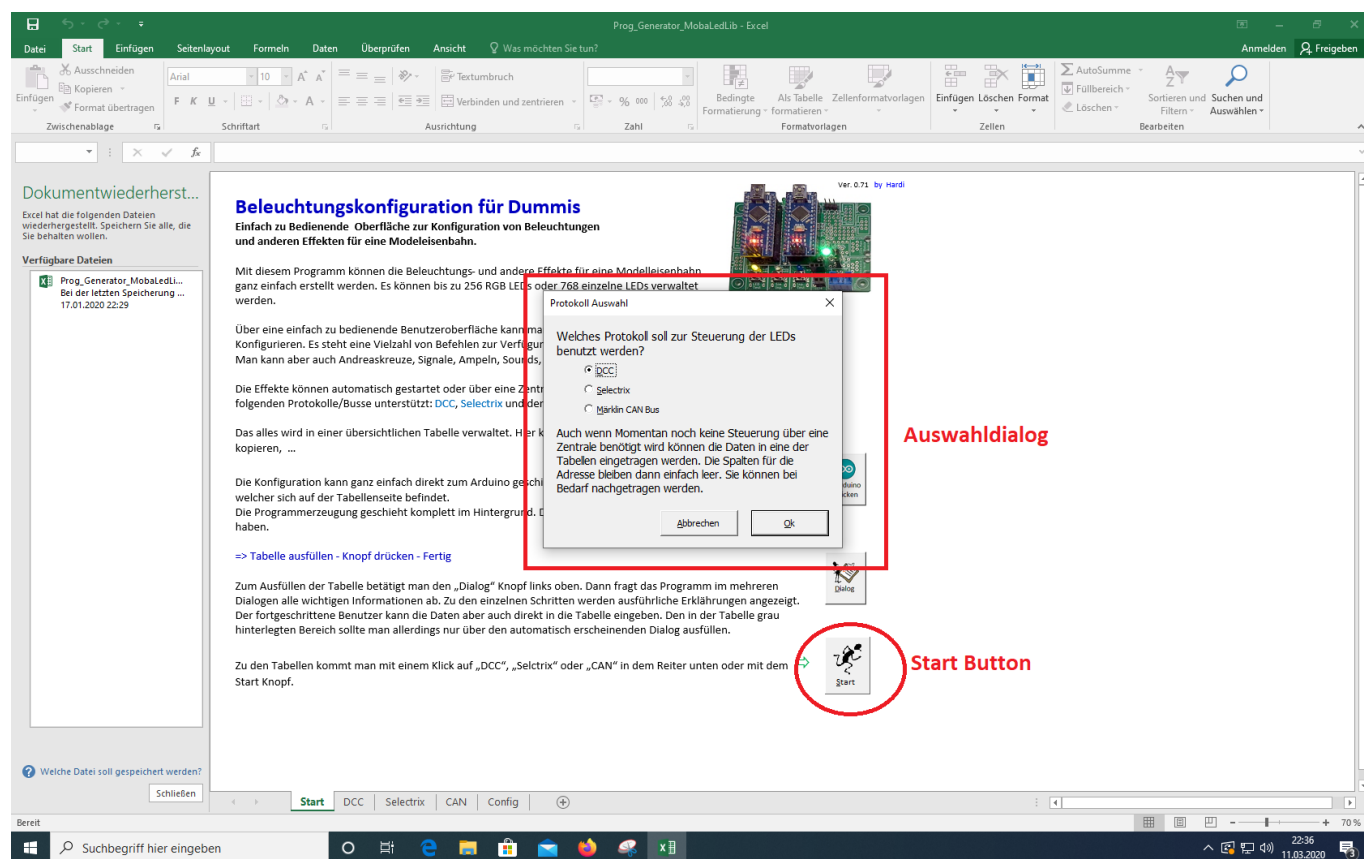
kopiert und ein Icon auf dem Desktop angelegt (x.y.z entspricht der Versionsnummer). Darüber kann der Prog-Generator künftig gestartet werden.



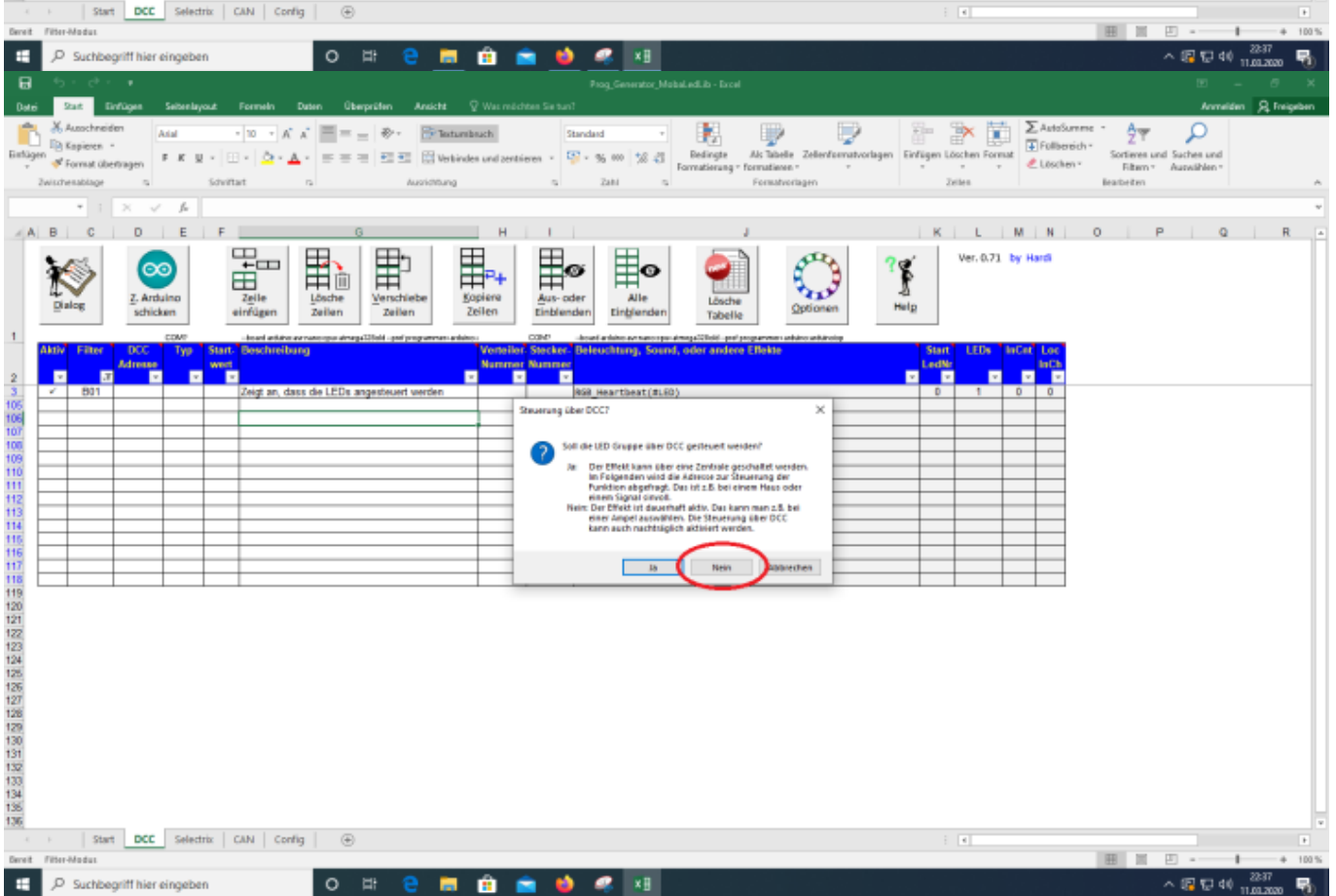
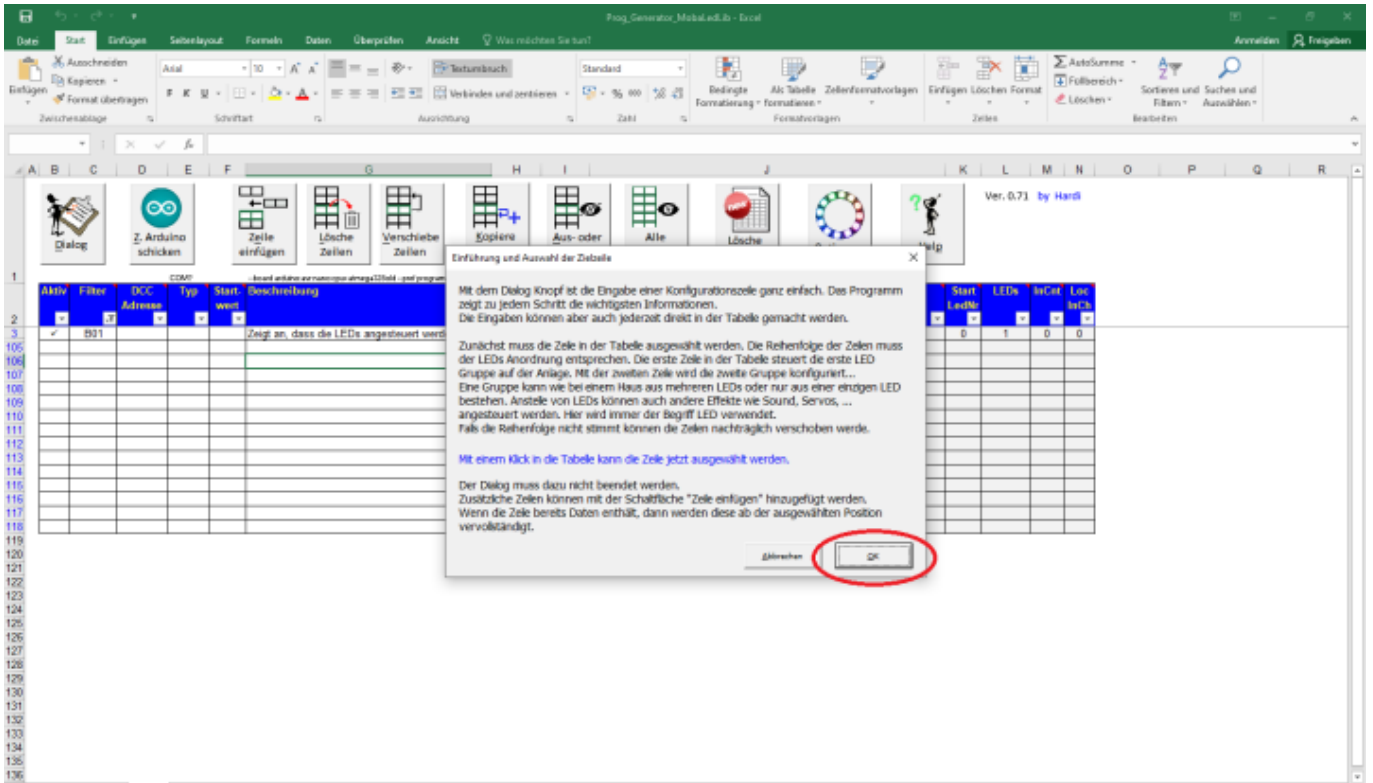
igung. Der Wichtigste ist sicherlich das „Belebte Haus“.

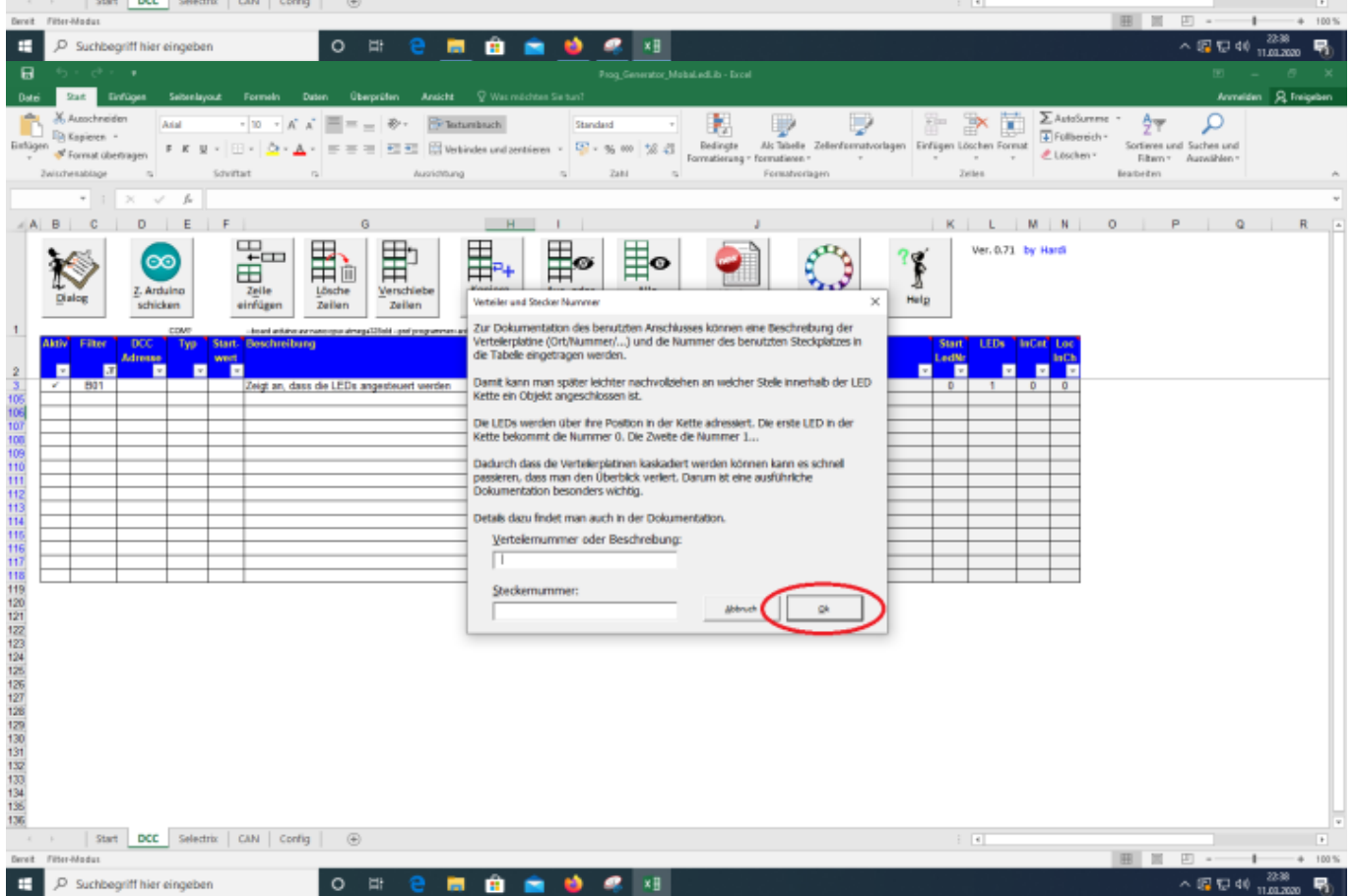
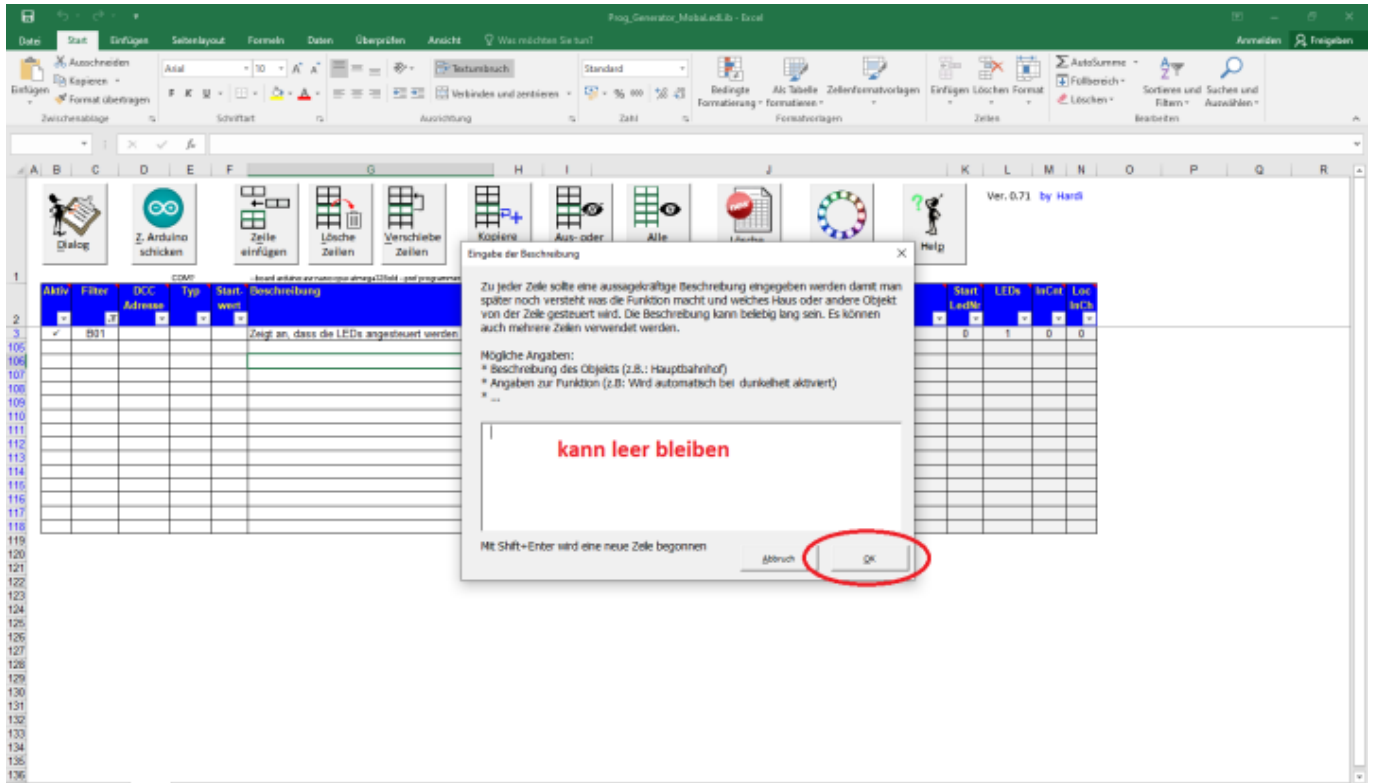


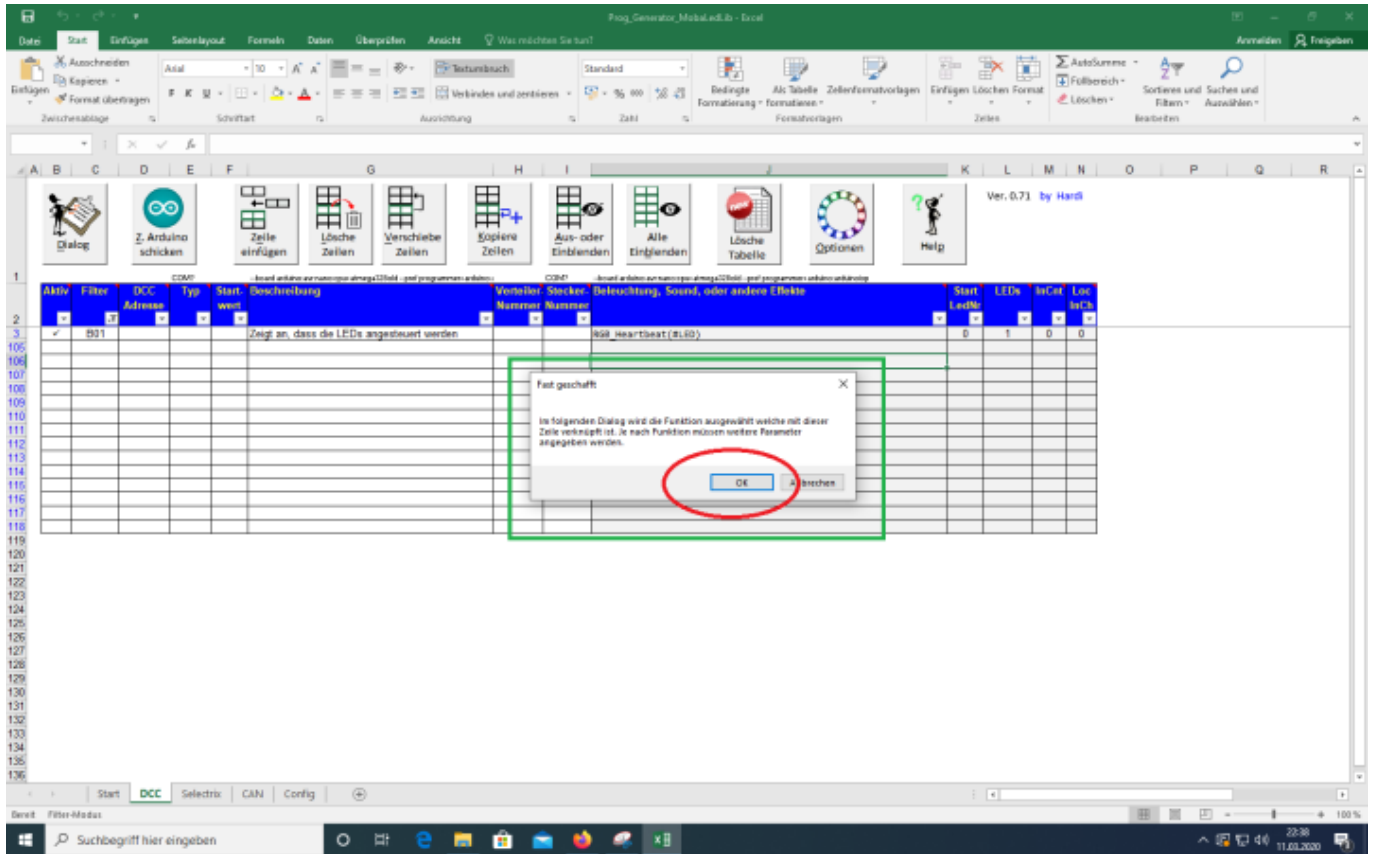
Ein Klick auf OK führt zu einem ersten Auswahldialog. Für erste Versuche ist es zunächst unerheblich welche der drei angebotenen Möglichkeiten (DCC-, Selektrix- oder CAN-Steuerung) ausgewählt wird.



Nach Anklicken des Start-Knopfes werden weitere Dialoge geöffnet. Eingaben sind für den ersten Einstieg nicht notwendig. Es genügt, die rot markierten Optionen auszuwählen. Erläuterungen zu den Dialogen sind in der ausführlichen Anleitung zum Prog_Gen zu finden. Sollte der Interviewer-Dialog nicht aufgehen, einfach auf den Button „Dialog“ klicken.







Die letzte Schaltfläche führt uns schließlich zur Makroauswahl.

✕

Auswahl des Makros

Makroauswahl:

Name	Beschreibung
Const	LED welche, gesteuert von "InCh", dauerhaft An oder Aus ist.
House	Mit dieser Funktion wird ein „belebtes“ Haus nachgebildet.
GasLights	Simuliert das einschaltverhalten und flackern von Gasbetriebenen Straßenlaternen.
Set_ColTab	Mit dem Makro „Set_ColTab()“ kann man die Farben und Helligkeiten der Lampen individuell anpassen.
Set_TV_COL1	Definiert die Simulation des Farb TV Kanals 1
Set_TV_COL2	Definiert die Simulation des Farb TV Kanals 2
Set_TV_BW1	Definiert die Simulation des Schwarz-Weiß TV Kanals 1
Set_TV_BW2	Definiert die Simulation des Schwarz-Weiß TV Kanals 2
Set_Def_Neon	Definiert die Simulation der defekten Leuchtstoffröhre
Button	Dieses Makro speichert ein Ereignis (z.B. Tastendruck) für eine bestimmte Zeit mit Abbruchmöglichkeit.
ButtonNOff	Dieses Makro speichert ein Ereignis (z.B. Tastendruck) für eine bestimmte Zeit ohne Abbruchmöglichkeit.
Blinker	Blinker mit einstellbarer Periode.
BlinkerInvInp	Blinker mit inversem Ausgang und einstellbarer Periode.
BlinkerHD	Blinker mit einstellbarer Periode bei dem der Ausgang abwechselnd Hell und Dunkel aber nicht ganz aus geht.
Blink2	Blinker mit einstellbaren Zeiten und Helligkeitswerten.
Blink3	Blinker mit einstellbaren Zeiten und Helligkeitswerten. Zusätzlich kann die Helligkeit im deaktivierten Zustand bestimmt werden.
BlueLight1	Diese Funktion generiert das typische doppelte Blitzen eines Blaulichts bei Einsatzfahrzeugen.
BlueLight2	Diese Funktion generiert das typische doppelte Blitzen eines Blaulichts mit geringfügig anderer Frequenz als BlueLight1.
Leuchtf Feuer	Dieses Makro generiert das Blinkmuster eines Windrads.
Andreaskreuz	Generiert das abwechselnd Blinkenden Lampen in Andreaskreuzen.
AndreaskrRGB	Generiert das abwechselnd Blinkenden Lampen in Andreaskreuzen zu Demonstrationszwecken mit einer RGB LED.
RGB_AmpelX	Damit wird das Muster zweier Ampeln für eine Kreuzung mit 6 RGB LEDs erzeugt.
RGB_AmpelXFade	Damit wird das Muster zweier Ampeln für eine Kreuzung mit 6 RGB LEDs erzeugt. Die Lichter werden langsam übergeblendet.
AmpelX	Damit wird das Muster zweier Ampeln für eine Kreuzung erzeugt.
Flash	Die „Flash()“ Funktion erzeugt ein zufälliges Blitzen eines Fotografen.

Mit dieser Funktion wird ein „belebtes“ Haus nachgebildet. In diesem Haus sind zufällig nur einige der Räume beleuchtet. Die Farbe und die Helligkeit der Beleuchtungen können individuell vorgegeben werden. Es lassen sich auch bestimmte Effekte wie Fernseher flackern oder ein offener Kamin für einzelne Räume konfigurieren. Außerdem kann das Einschaltverhalten angepasst werden (Neonröhrenflackern oder langsam heller werdende Gaslampen).

House(LED,InCh, On_Min,On_Limit, ...)

Expertenmodus

Auswahl eines Beispiels (belebtes Haus)

Für unseren ersten Test wählen wir mit dem Select-Knopf das „House“ aus und erhalten die folgende Seite zur weiteren Auswahl der Funktionen:

House: Simulation eines "belebten" Hauses in dem zufällig und abwechselnd nur einige der Räume beleuchtet sind

Das ist vermutlich die am häufigsten genutzte Funktion auf einer Modelleisenbahn. Mit Ihr wird ein „belebtes“ Haus nachgebildet. In diesem Haus sind zufällig nur einige der Räume beleuchtet. Die Farbe und die Helligkeit der Beleuchtungen können individuell vorgegeben werden. Es lassen sich auch bestimmte Effekte wie Fernseher flackern oder ein offener Kamin für einzelne Räume konfigurieren. Außerdem kann das Einschaltverhalten angepasst werden (Neonröhrenflackern oder langsam heller werdende Gaslampen).

Mögliche Beleuchtungstypen:

ROOM_DARK	ROOM_BRIGHT	ROOM_WARM_W	ROOM_RED	ROOM_D_RED	*		
ROOM_COL0	ROOM_COL1	ROOM_COL2	ROOM_COL3	ROOM_COL4	ROOM_COL5	ROOM_COL345	
FIRE	FIRE0	FIREB	ROOM_CHIMNEY	ROOM_CHIMNEYD	ROOM_CHIMNEYB	*	
ROOM_TV0	ROOM_TV0_CHIMNEY	ROOM_TV0_CHIMNEYD	ROOM_TV0_CHIMNEYB	ROOM_TV1	ROOM_TV1_CHIMNEY	ROOM_TV1_CHIMNEYD	ROOM_TV1_CHIMNEYB
NEON_LIGHT	NEON_LIGHT1	NEON_LIGHT2	NEON_LIGHT3	NEON_LIGHTD	NEON_LIGHT1D	NEON_LIGHT2D	NEON_LIGHT3D
NEON_LIGHTM	NEON_LIGHT1M	NEON_LIGHT2M	NEON_LIGHT3M	NEON_LIGHTL	NEON_LIGHT1L	NEON_LIGHT2L	NEON_LIGHT3L
NEON_DEF_D	NEON_DEF1D	NEON_DEF2D	NEON_DEF3D				
SINGLE_LED1	SINGLE_LED2	SINGLE_LED3	SINGLE_LED1D	SINGLE_LED2D	SINGLE_LED3D		
GAS_LIGHT	GAS_LIGHT1	GAS_LIGHT2	GAS_LIGHT3	GAS_LIGHTD	GAS_LIGHT1D	GAS_LIGHT2D	GAS_LIGHT3D
SKIP_ROOM							

* = Unveränderbare Farben
Alle anderen Farben können mit dem Set_ColTab Befehl und dem Farbtest Programm angepasst werden

Ausgewählte Beleuchtungen: Anzahl: 5

ROOM_COL0, ROOM_TV0, ROOM_WARM_W, NEON_DEF_D, NEON_LIGHTM

Lösche Raum

2 Minimale Anzahl der zufällig aktiven Beleuchtungen

3 Maximale Anzahl der zufällig aktiven Beleuchtungen

Eingang invertieren

Individuelle Zeiten

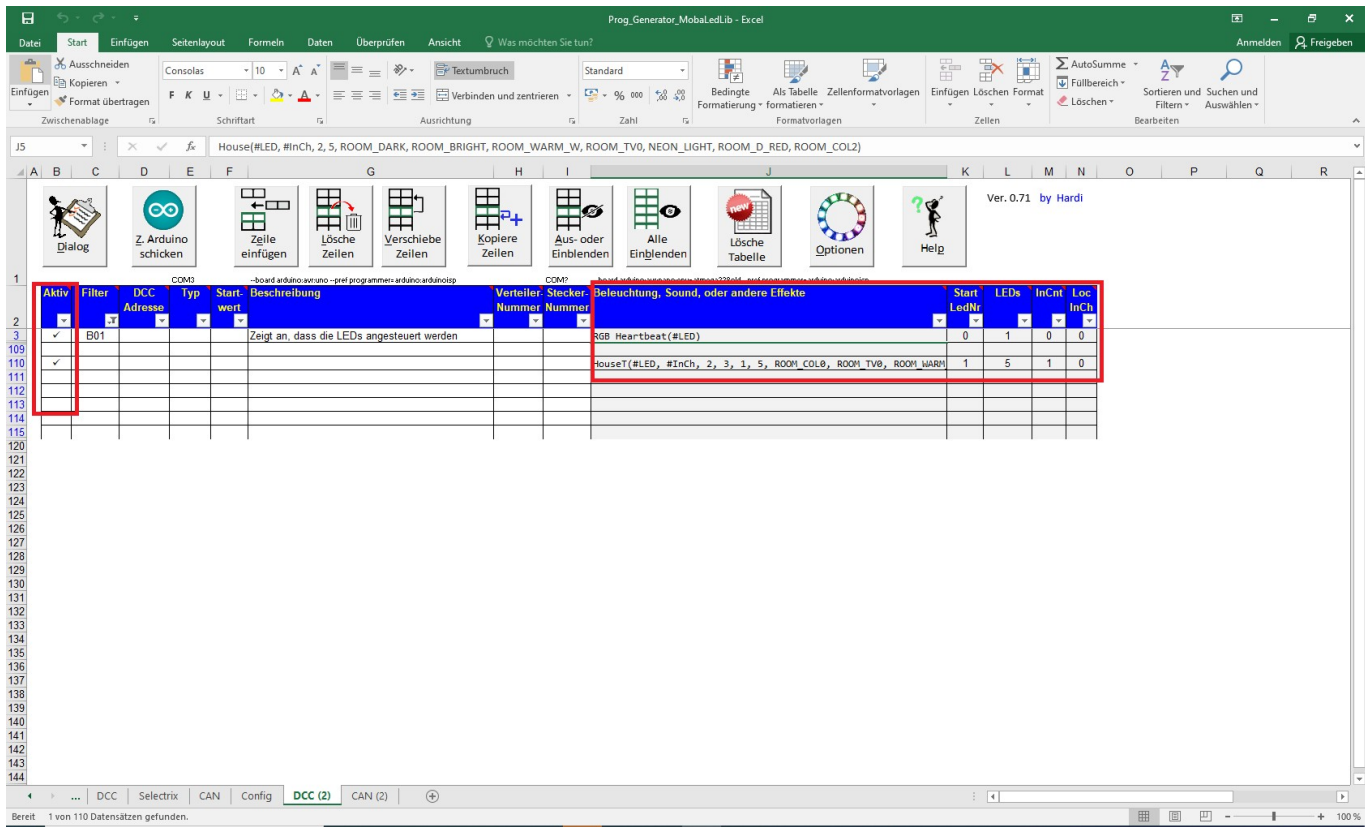
1 Minimale Zeit bis zur nächsten Änderung [sek]

5 Maximale Zeit bis zur nächsten Änderung [sek]

Abbrechen OK

RGB LED Kanäle: 5

Über das Tastenfeld „mögliche Beleuchtungstypen“ können unterschiedliche Beleuchtungen für die Räume eines Hauses ausgewählt werden. Wenn in einem Gebäude fünf Räume beleuchtet sind, müssen fünf Beleuchtungen ausgewählt werden. Auswahl und Anzahl werden angezeigt. Über die Taste „lösche Raum“ lassen sich Räume löschen um eine andere Beleuchtung auszuwählen. Durch Anklicken der Option „individuelle Zeiten“ lassen sich die Zeiten für den Beleuchtungswechsel den eigenen Bedürfnissen entsprechend anpassen. Für Testzwecke bietet es sich an, die „Maximale Zeit bis zur nächsten Änderung“ auf 5 [Sec] zu setzen. Nun über „OK“ die Auswahl bestätigen. Damit werden die Werte in die Excel-Tabelle eingefügt und die Seite sollte wie folgt aussehen:

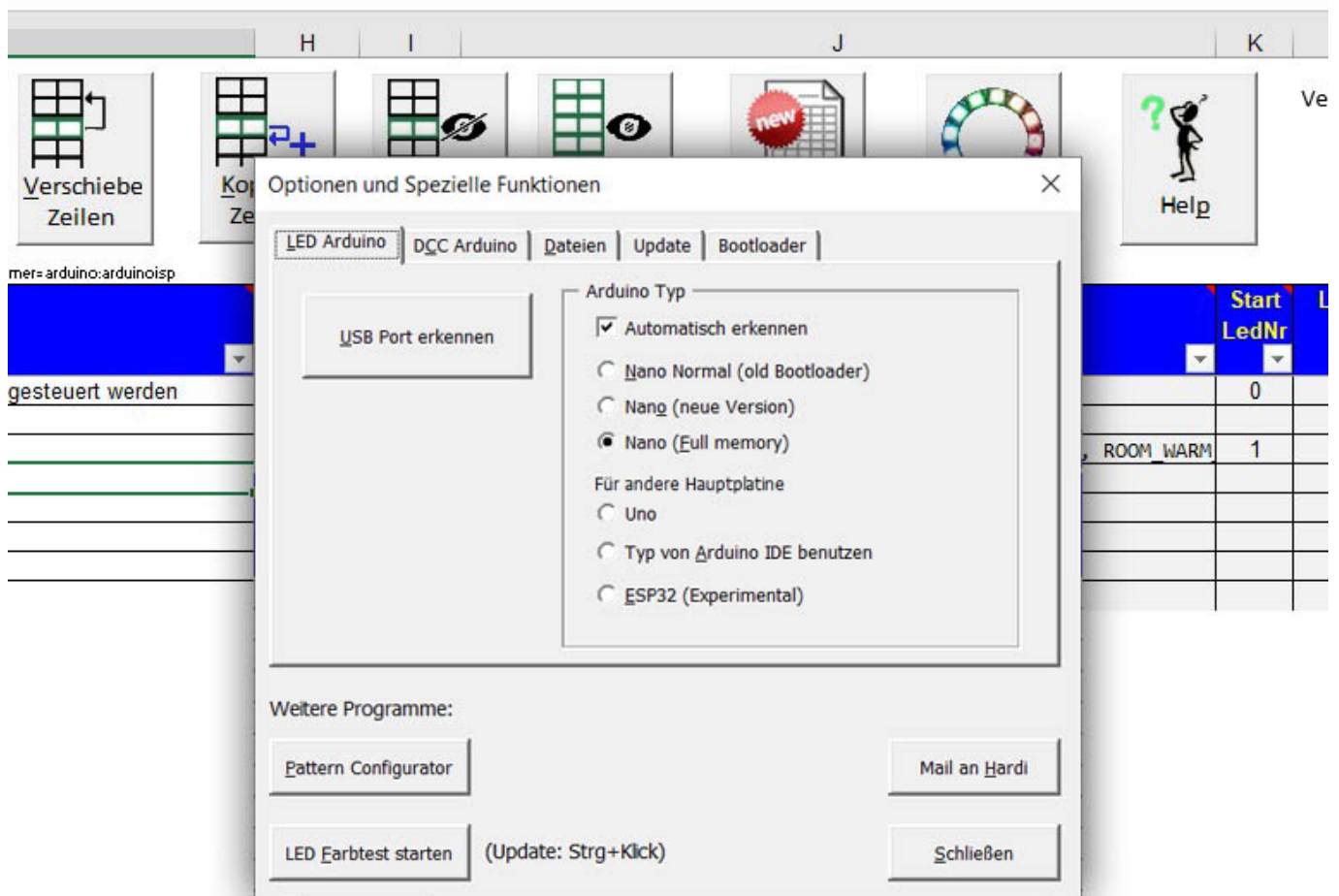


Zwei Zeilen sind aktiv geschaltet. Das ist an dem per Mausclick gesetzten Häkchen zu erkennen. Nicht-aktive Zeilen werden nicht zum Arduino übertragen. Die Felder in den Spalten Filter, DCC Adresse, Typ und Startwert bleiben frei. Das Feld „Beschreibungen“ kann zur Erläuterung für eigene Zwecke genutzt werden. Der grau hinterlegte Bereich ist automatisch befüllt worden und kann bzw. sollte nicht geändert werden. Über die roten Dreiecke in den Feldern könne zur weiteren Erklärung Tooltips aufgerufen werden.

Die erste Zeile mit dem Beispiel B01 „RGB_Heartbeat(#LED)“ wird standardmäßig gesetzt. Damit wird die erste LED in der Kette genutzt, um zu signalisieren, dass die Übertragung des Programms an den Arduino erfolgreich war und das System „lebt“. Falls die folgenden LEDs dann trotzdem nicht so arbeiten wie erwartet, hat man irgendwo in der Auswahl für die LEDs einen Fehler gemacht oder in der Verdrahtung der LEDs liegt ein Fehler vor.

Im nächsten Schritt wird unsere Auswahl zum ARDUINO geschickt. Beim ersten Senderversuch erfolgt die Aufforderung, den benutzten COM-Port festzulegen. Einfach den Anweisungen folgen und anschließend den „Z.Arduino schicken“ Knopf drücken.

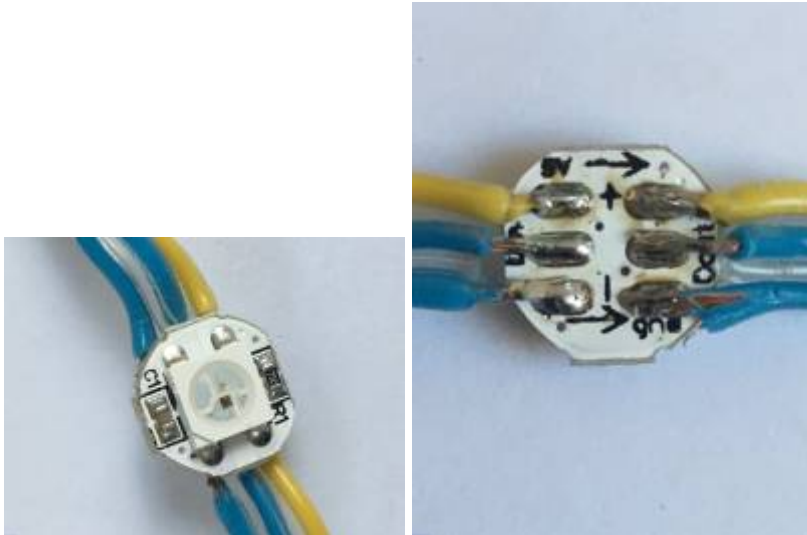
Der Vorgang zum Erkennen des COM-Ports kann auch über „Optionen“ → „USB Port erkennen“ angestoßen werden. Inzwischen erkennt das Programm den Arduino automatisch wenn das entsprechende Häkchen gesetzt ist. Die Auswahl benötigt man nur für den Uno oder sonstige Arduinos. Nach der automatischen Erkennung sollte für eine fehlerfreie Übertragung zum Arduino im Auswahlmeneu der genutzte Arduino-Typ markiert sein, für Arduino Nano Clones i.d.R. „Nano Normal (old Bootloader)“. Ist der Arduino Typ nicht aufgeführt, sollte über das Arduino IDE der Typ ausgewählt werden und hier übernommen werden: „Typ von Arduino IDE benutzen“.



Hat bis hierher alles wie gewünscht funktioniert ist jetzt zwar das Programm auf dem Arduino aber es fehlen noch die LEDs.

Anschluss der RGB LEDs an einen Arduino

Im Abschnitt [Bezugsquellen](#) sind Händler aufgeführt, die schnell und etwas teurer bzw. „billiger“ und mit längeren Lieferzeiten die Bauteile liefern.



Für den ersten Versuche reicht es, einfach ein paar RGB LEDs zusammen zu löten und direkt mit den Anschlüssen des Arduinos zu verbinden. Vorsichtshalber vorher den Arduino vom Computer trennen. Aber nicht übertreiben. Wenn der Arduino mit einem normalen USB-Netzteil und USB2.0 betrieben wird stehen 500 mA am 5V-Anschluss zur Verfügung. Das entspricht 5 RGB-LEDs mit maximaler Helligkeit auf allen Kanälen. Verbindet man mehr LEDs, reicht die Stromversorgung über den Arduino ggf. nicht mehr aus.

Bitte beachten, dass hierfür unbedingt RGB-LED verwendet werden müssen und keine RGBW-LED. Das W steht hierbei für einen zusätzlichen Weiß-Kanal, der von der Software nicht unterstützt wird. Wird zwischen mehreren RGB-LED eine RGBW-LED in Reihe geschaltet, kann es passieren, dass die Kanalprogrammierung aus dem Prog-Generator nicht mehr die richtigen LED anspricht, weil er nacheinander drei Farb-Kanäle erwartet, wo tatsächlich aber vier Kanäle (Rot, Grün, Blau plus Weiß) sind. RGBW- und RGB-LED sehen äußerlich gleich aus und auch in den Produktbeschreibungen ist nicht immer eindeutig erkennbar, welche Art von LED angeboten wird.

Es werden nur drei Kabel benötigt:

- 5 Volt von der ersten LED zum 5V Pin des Arduino
- GND von der ersten LED zu einem GND Pin des Arduino
- In von der ersten LED zum Pin D6 des Arduino.

Nun den Arduino wieder über das USB Kabel mit dem Computer verbinden. Sind keine Änderungen vorgesehen, reicht auch eine 5 Volt Spannungsquelle aus. Die LEDs sollten nun leuchten, wie oben über das „House“ festgelegt.



Nun können weitere Macros ausprobiert werden, um erste Erfahrungen zu sammeln. Einfach im House Macro andere Einstellungen wählen und erneut zum Arduino schicken. Ebenso können zusätzliche Zeilen eingefügt werden und vorhandene aktiviert bzw. deaktiviert werden. Bitte bedenken, dass die Anzahl der im MACRO ausgewählten LEDs die der tatsächlich angeschlossenen LEDs für ein sinnvolles Ergebnis nicht übersteigen sollte.

Nun viel Spaß mit der MobaLedLib!

Tipp

Fenster teilen:

Anordnung einstellen mit der Maus Um ein Fenster zu teilen, müssen Sie mit der linken Maustaste auf die obere Leiste des Fensters klicken. Halten Sie die Maustaste gedrückt und ziehen Sie das Fenster entweder bis an die linke oder rechte Seite Ihres Bildschirms. Es ist auch möglich, das Fenster in eine der vier Ecken zu ziehen, wenn Sie den Bildschirm in 4 Fenster unterteilen möchten. Das Fenster wird nun automatisch an eine Seite andockt und ermöglicht die Auswahl anderer Fenster anhand einer Miniaturvorschau. Klicken Sie auf einen der Vorschläge, wird dieses Programm an die andere Seite andockt.

Anordnung einstellen durch Taskleiste Es ist auch möglich, die Anordnung über die Taskleiste

einzustellen. Klicken Sie dafür mit der rechten Maustaste unten auf ein freies Feld in der Taskleiste. Es erscheint ein Menü, bei welchem Sie zwischen „Fenster überlappen“, „Fenster gestapelt anzeigen“ oder „Fenster nebeneinander anzeigen“ auswählen können.

Shortcut für das Anordnen der Fenster Öffnen Sie eines der Fenster, das Sie verwenden wollen. Halten Sie die spezielle Windows-Taste auf der Tastatur gedrückt und bestimmen Sie mit den Pfeiltasten, an welche Seite bzw. Ecke des Bildschirms das Fenster angedockt werden soll: z.B.: [Windows] + [Pfeil rechts]

Bezugsquellen

Die angegebenen Bezugsquellen sind nur Beispiele. Die Teile gibt es sicherlich kostengünstig auch bei vielen anderen Herstellern.

- WS2812 LEDs schnelle Lieferzeit:

https://www.amazon.de/Kuman-100pcs-WS2812B-Addressable-Arduino/dp/B07B7GVFRG/ref=pd_sbs_60_2?_encoding=UTF8&pd_rd_i=B07B7GVFRG&pd_rd_r=c86c60ff-1663-11e9-bcfc-89e94cb2608a&pd_rd_w=loTGG&pd_rd_wg=RoOus&pf_rd_p=51bcaa00-4765-4e8f-a690-5db3c9ed1b31&pf_rd_r=H975JS0KT92BHS6HGA50&psc=1&refRID=H975JS0KT92BHS6HGA50

- WS2812 LEDs billiger aber lange Lieferzeit:

<https://www.aliexpress.com/item/32694592019.html?spm=a2g0s.9042311.0.0.27424c4d1oivPG>

- WS2812 LEDs mit angelöteten Kabeln:

<https://de.aliexpress.com/item/1941066393.html?spm=a2g0s.9042311.0.0.3da24c4dXk8Cxy>

- WS2811 Module zum ansteuern normaler LEDs:

<https://www.aliexpress.com/item/32755999690.html?spm=a2g0s.9042311.0.0.27424c4d1R2J3z>

- 64 WS2812 LEDs für Tests am Schreibtisch:

https://www.amazon.de/AZDelivery-Matrix-CJMCU-8-Arduino-Raspberry/dp/B078HYP681/ref=sr_1_19?keywords=Ws2812+Led&qid=1570436873&sr=8-19

- Platinen: Mail an: LedLib@yahoo.com

From:

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

Permanent link:

<https://wiki.mobaledlib.de/anleitungen/quickstart?rev=1674636351>

Last update: **2023/01/25 09:45**

