

Installation auf einem Raspberry

ACHTUNG: Die Seite ist noch in ARBEIT

Die Installation der MobaLedLib auf einem Raspberry Pi wird folgendermassen durchgeführt:

Vorraussetzung

Raspberry Pi 3B, B+ oder 4 mit installiertem Rasbian. Ich empfehle die neueste Version zu installieren.

Schritt 1: Die ARDUINO IDE auf dem Raspberry Pi installieren

Im Prinzip erfolgt die Installation der ARDUINO IDE auf dem Raspberry ähnlich der Installation auf einem Windows PC. Da es einige kleine aber wichtige Unterschiede gibt, findest Du im folgenden die angepasste Anleitung:

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
- ein Raspberry PI 3 B, B+ oder 4

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

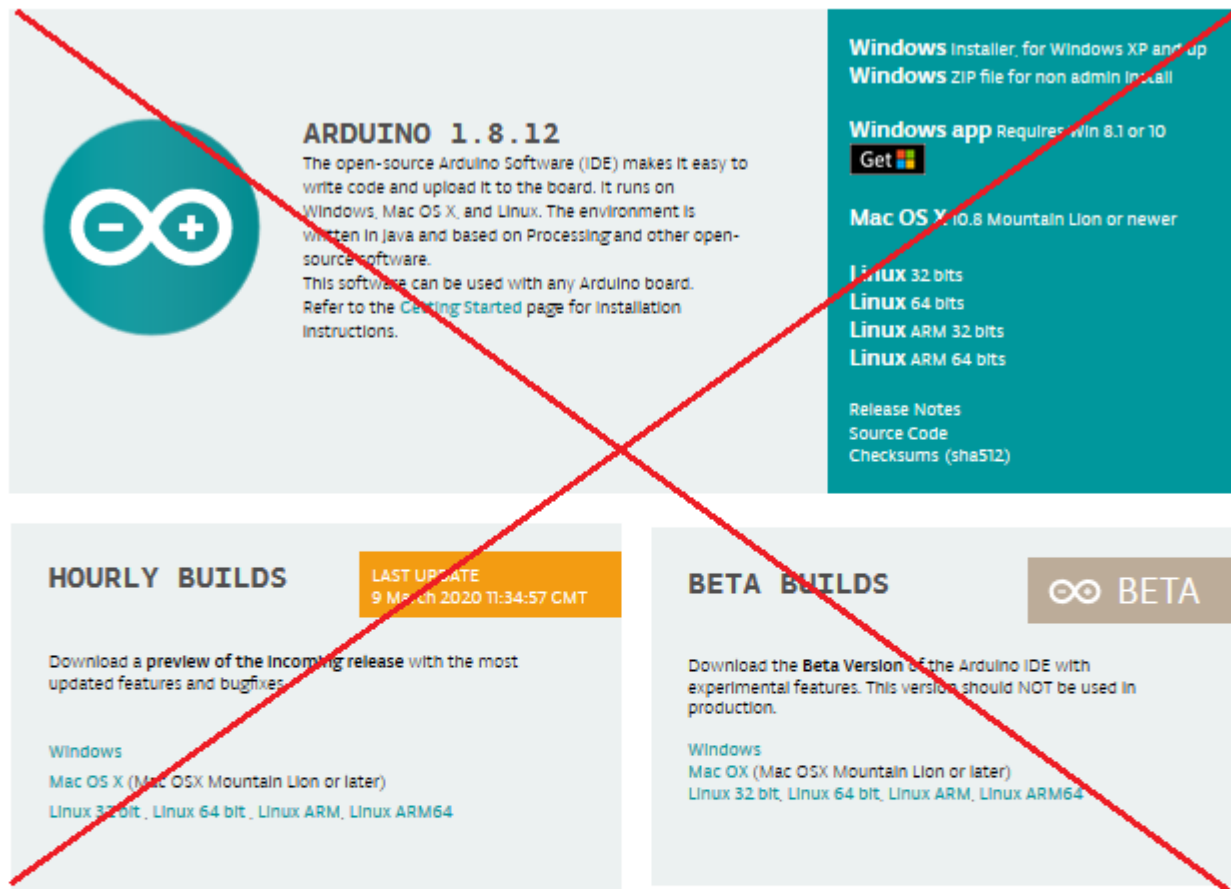
Diese Anleitung gilt für eine Raspberry Umgebung mit einem installierten Raspbian.

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

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

Download the Arduino IDE



The screenshot shows the Arduino IDE download page for version 1.8.12. It includes the Arduino logo, a description of the IDE, and links to download the installer or ZIP file for Windows, Mac OS X, and Linux. There are also links for hourly builds and beta builds. The entire page is crossed out with a large red X.

ARDUINO 1.8.12
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
[Get](#)

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

HOURLY BUILDS **LAST UPDATE** 9 March 2020 11:34:57 CMT

Download a **preview of the incoming release** with the most updated features and bugfixes.

Windows
Mac OS X (Mac OS X Mountain Lion or later)
Linux 32 bit, Linux 64 bit, Linux ARM, Linux ARM64

BETA BUILDS **BETA**

Download the **Beta Version** of the Arduino IDE with experimental features. This version should NOT be used in production.

Windows
Mac OS X (Mac OS X Mountain Lion or later)
Linux 32 bit, Linux 64 bit, Linux ARM, Linux ARM64

Previous Releases

Download the [previous version of the current release](#) the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta version](#).

All the [Arduino 00xx versions](#) are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.

Source Code

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#).

The source code archives of the latest release are available [here](#). The archives are PGP-signed so they can be verified using this [gpg key](#).

Wir empfehlen, die **Version 1.8.9** aus den „Previous Releases“ zu nutzen, da in den aktuelleren Versionen (Stand März 2020) viele Debug Ausgaben beim Kompilieren angezeigt werden.

Nach dem Klick auf den Download-Link öffnet sich das folgende Fenster, in dem die Version **1.8.9 LINUX ARM 32 Installer** ausgewählt werden kann.



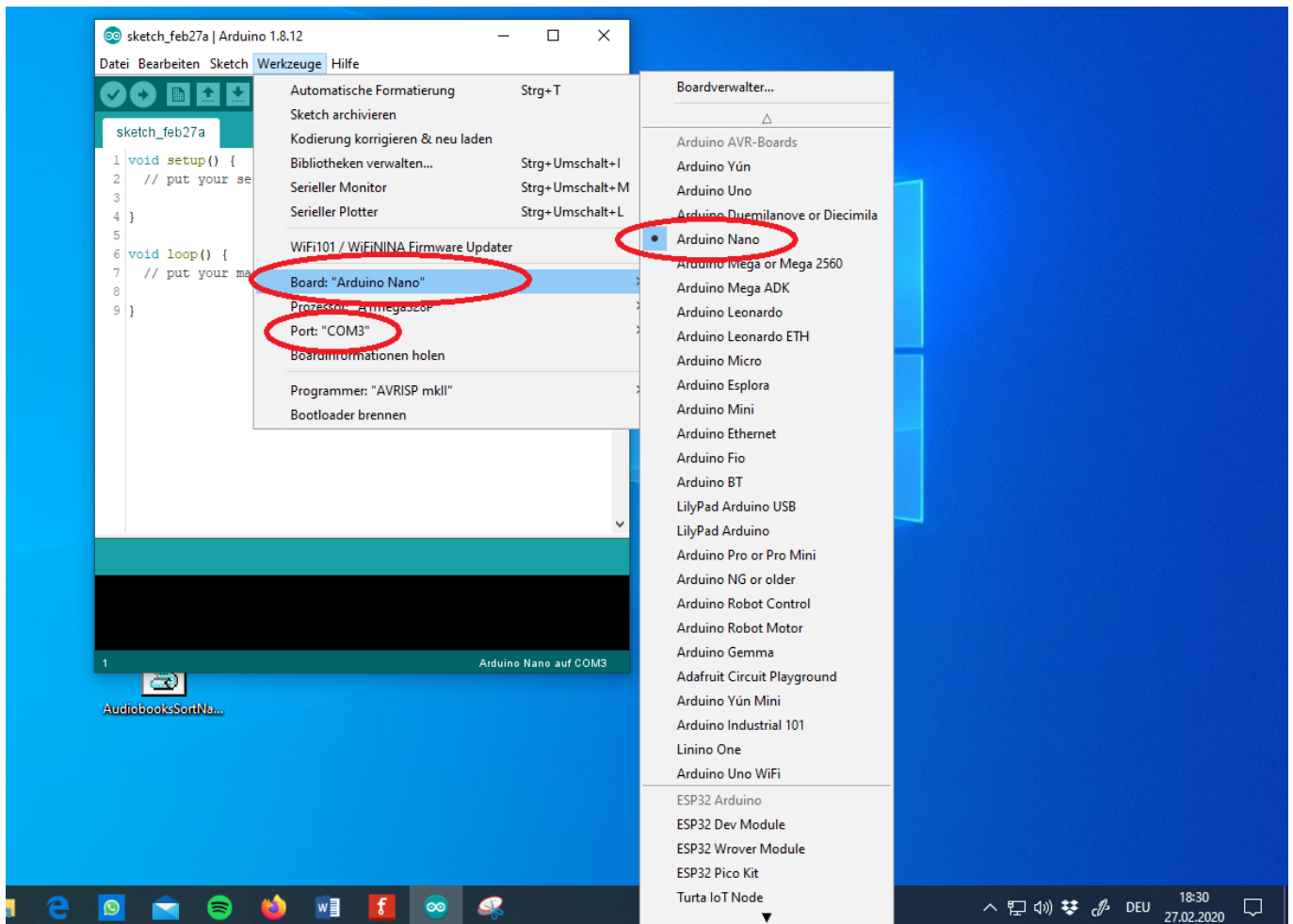
Anschließend wird man aufgefordert, für das Arduino Projekt zu spenden. Es ist gut, wenn man das

großartige Projekt auf diese Weise unterstützt. Das Programm kann aber problemlos auch ohne Spenden heruntergeladen werden. Das heruntergeladene Programm, hier „arduino-1.8.9-linux.arm.tar.gz“ findet man in dem „Downloads“-Ordner des Raspberry.

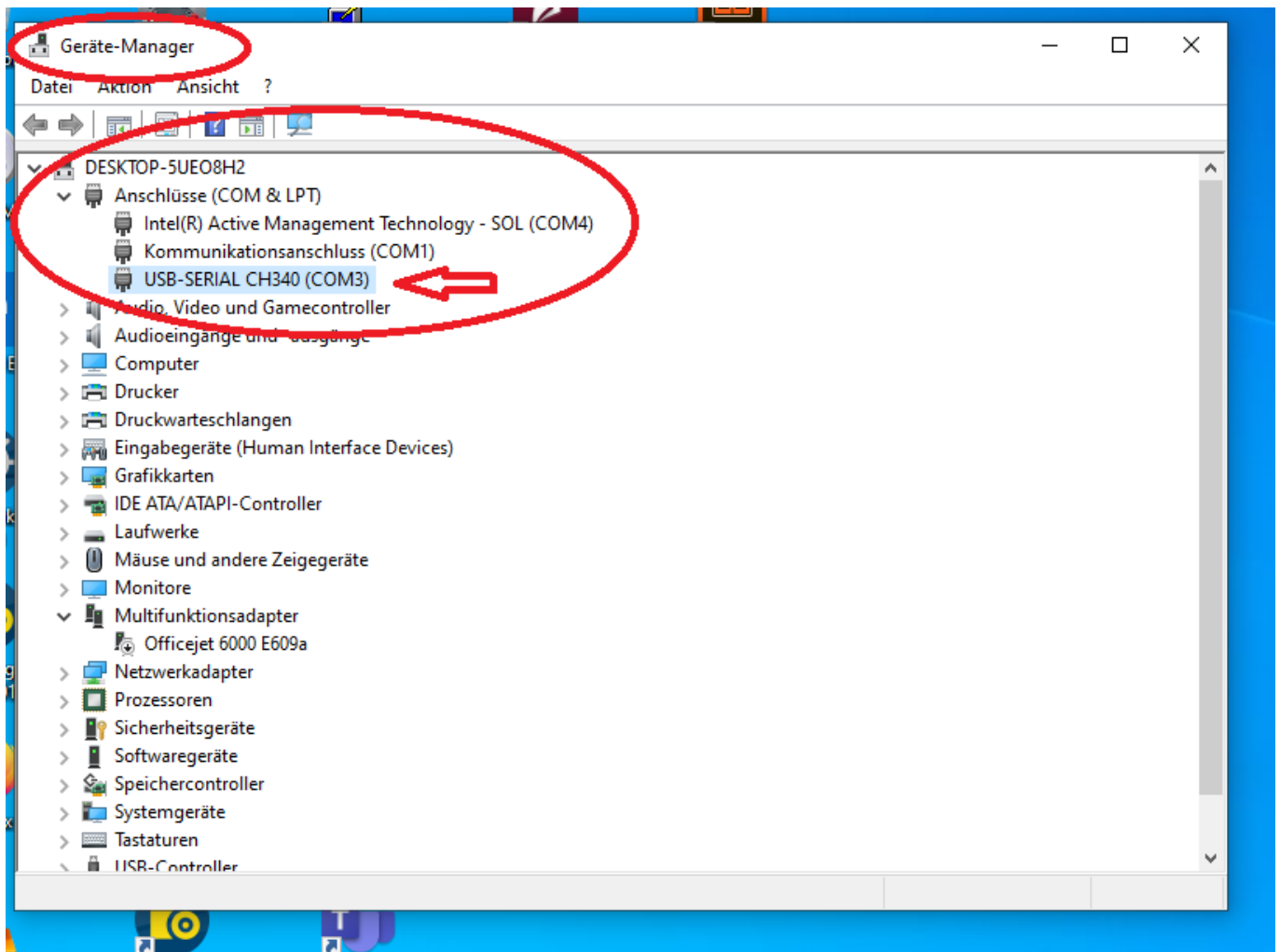


Zur Installation wird das Programm entpackt. Ich habe es gleich im Downloads Ordner entpackt. Ist nicht optimal, aber funktioniert.

Ab hier muss noch überarbeitet werden 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.



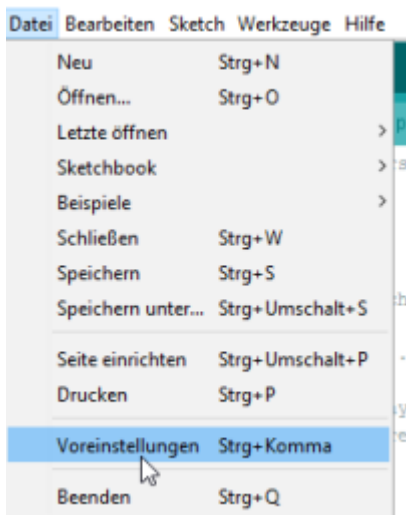
Falls das nicht der Fall ist, sollte zunächst überprüft werden, ob der Arduino erkannt wird und an welchem COM-Port er angeschlossen wurde. Dazu im WINDOWS Geräte-Manager, zu erreichen über die WINDOWS Starttaste mit rechter Maustaste anklicken, unter „Anschlüsse(COM&LPT)“ prüfen ob der Arduino dort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

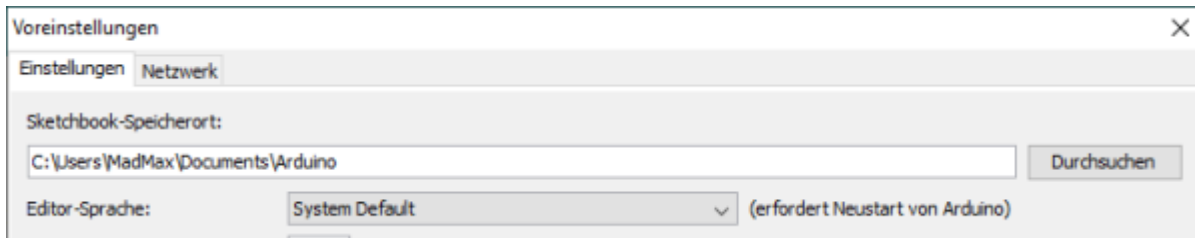


Im Falle von original Arduinos erscheint der Namen z.B. NANO. Clones werden hingegen mit dem verbauten Chip aufgeführt, hier der CH340. Das liegt daran, dass dort statt der FTDI-Chips die deutlich preiswerteren CH340G USB 2 Serial Chips verbaut werden. Hilfe, mein Arduino wird nicht erkannt. → siehe Abschnitt [Fehlerbehebung](#). 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

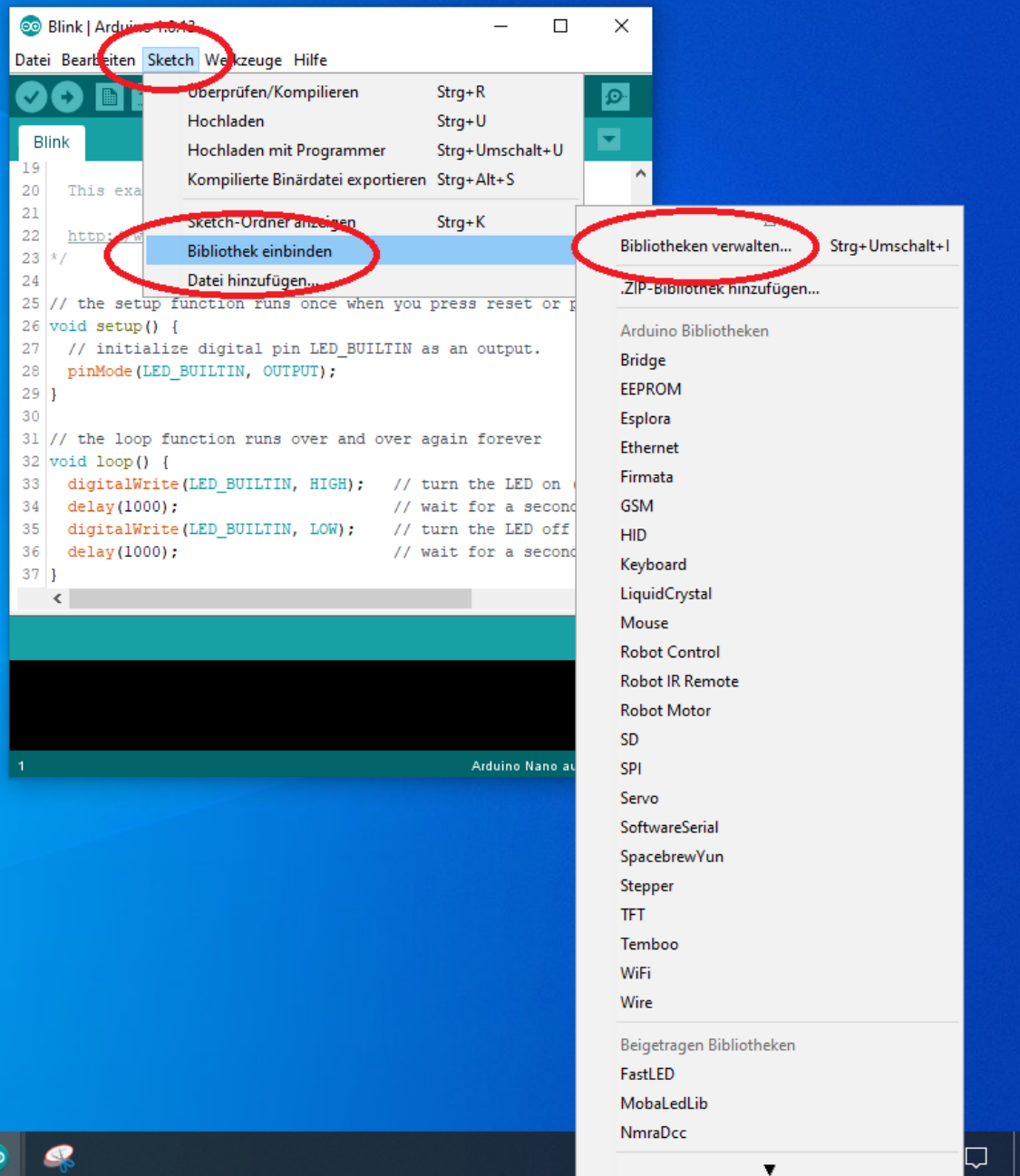
Dies kann über das Menü innerhalb der Arduino IDE gemacht werden.



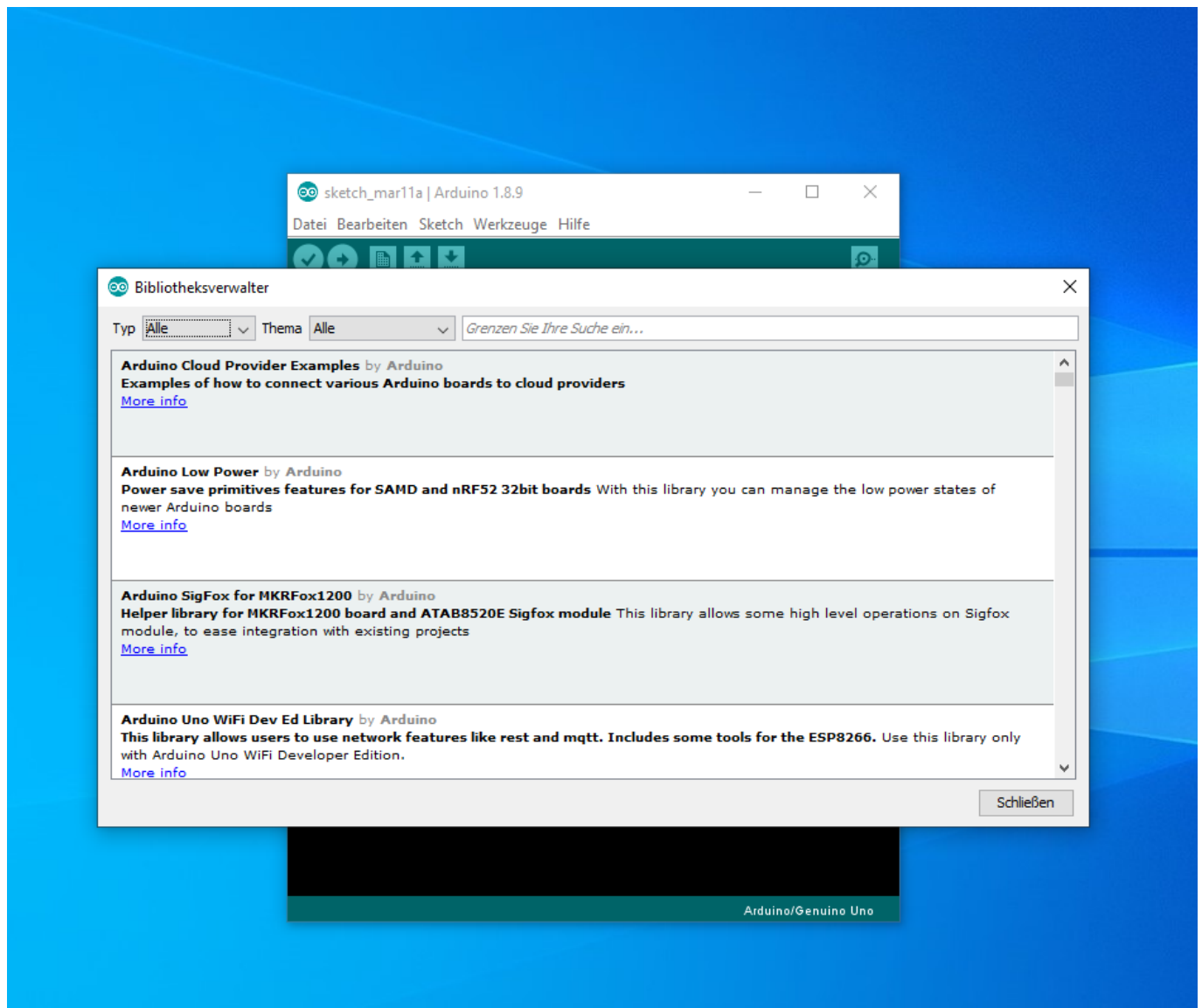


=====

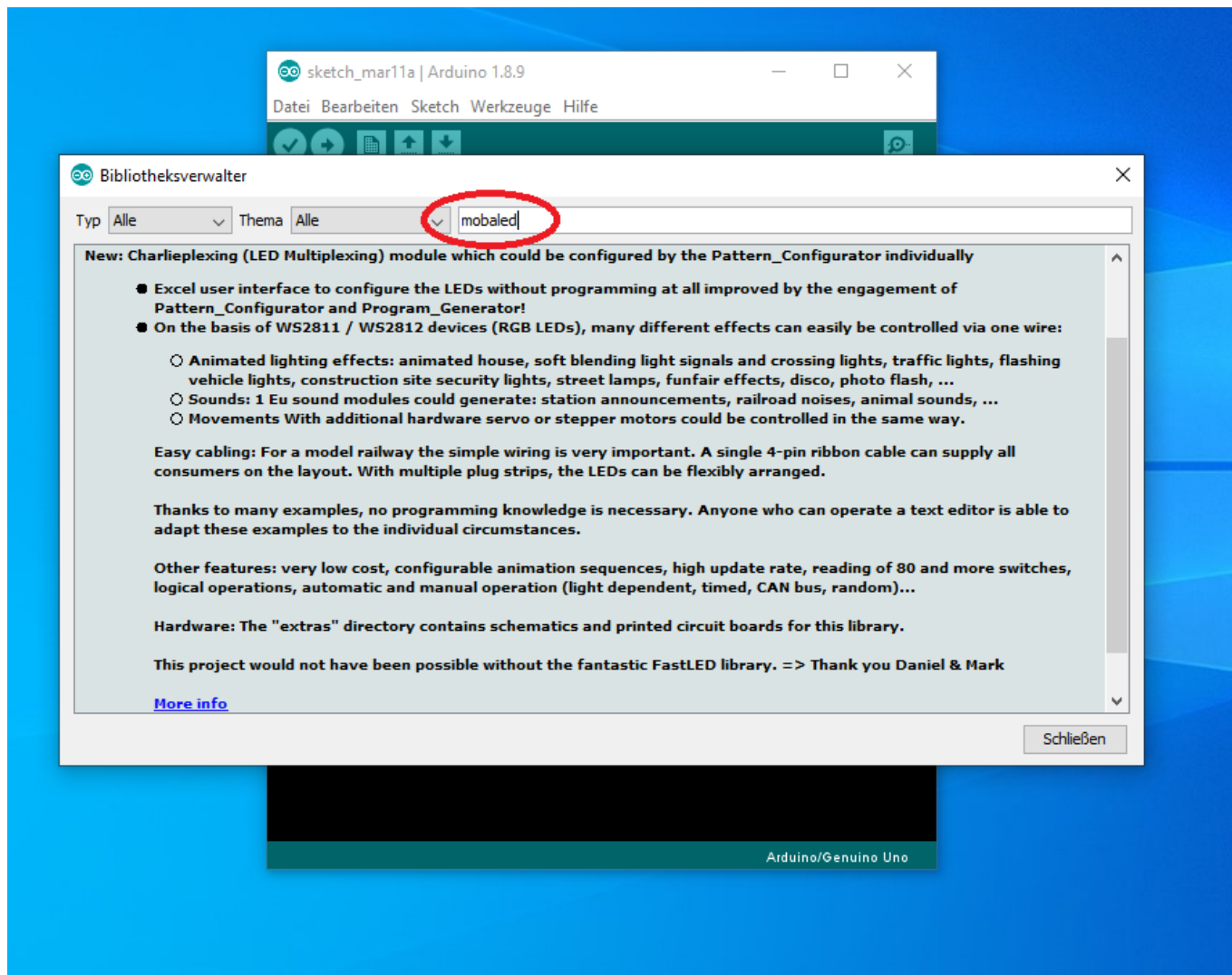
MobaLedLib (MLL) einrichten ===== Zunächst muss die Bibliotheksverwaltung der IDE über Sketch → Bibliothek einbinden → Bibliothek verwalten geöffnet werden.



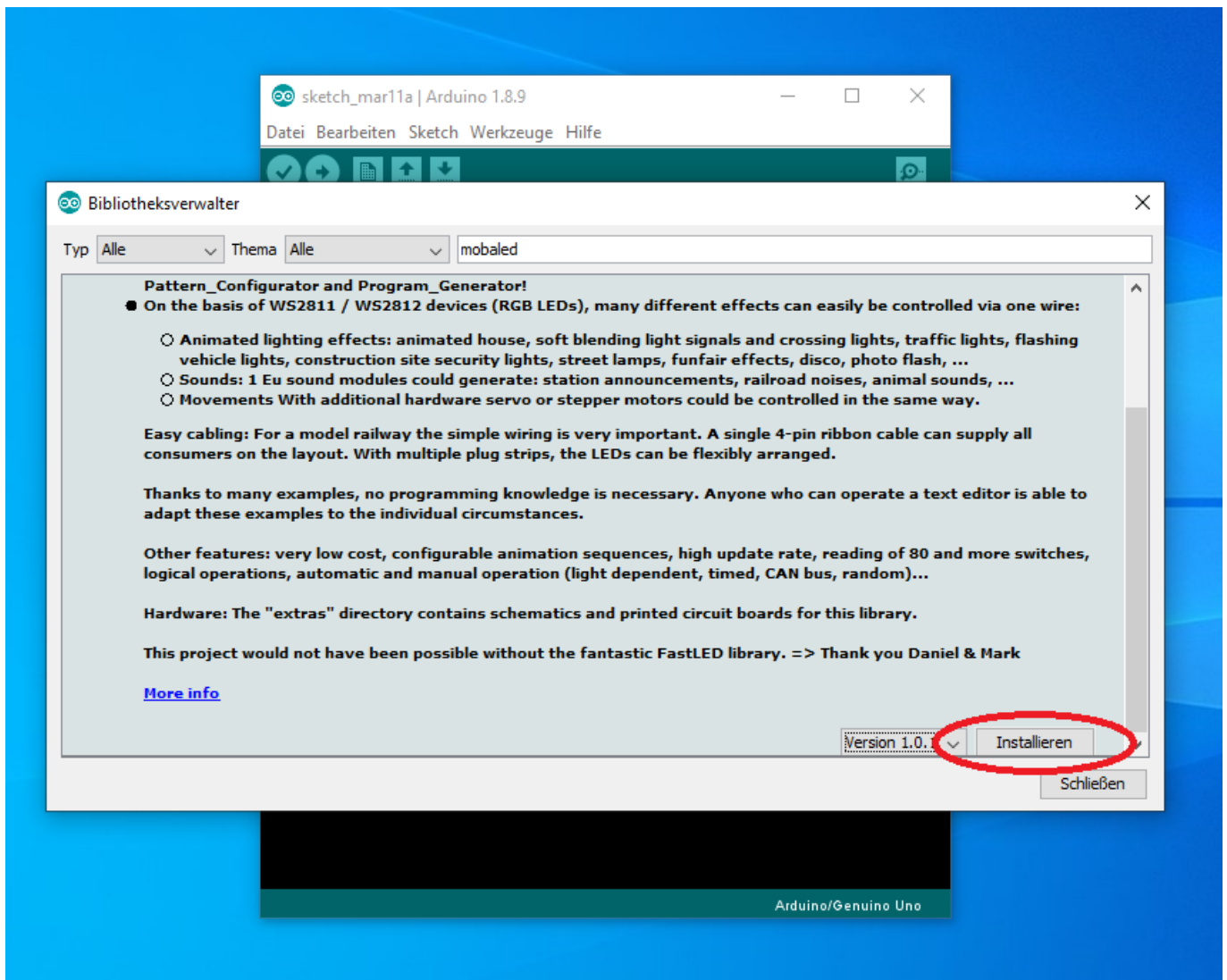
Es öffnet sich der Bibliotheksverwalter.



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.

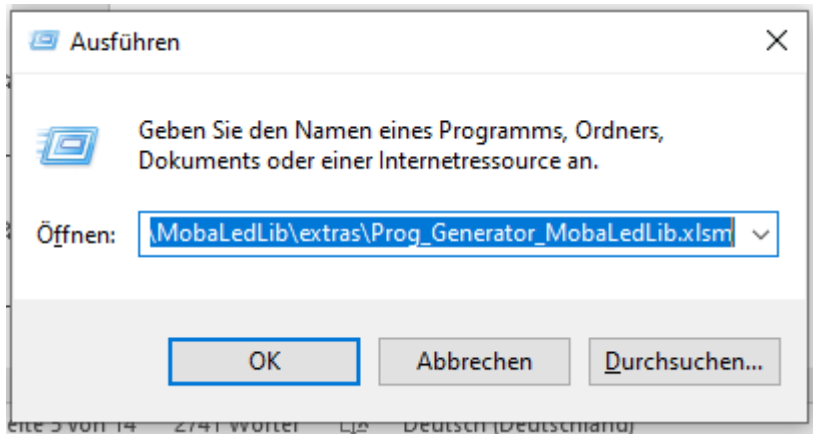
Mit einem Klick auf den Knopf wird die Installation abgeschlossen. Bitte nun noch mit der gleichen Vorgehensweise in das Suchfeld „fastled“ eingeben und installieren.

(Sollte es zu Fehlermeldungen in Zusammenhang mit der FastLED kommen, gibt es unter https://wiki.mobaledlib.de/anleitungen/faq#bekannte_fehler_und_fehlermeldungen unter Punkt „Bekannte Fehler und Fehlermeldungen“ bzw.

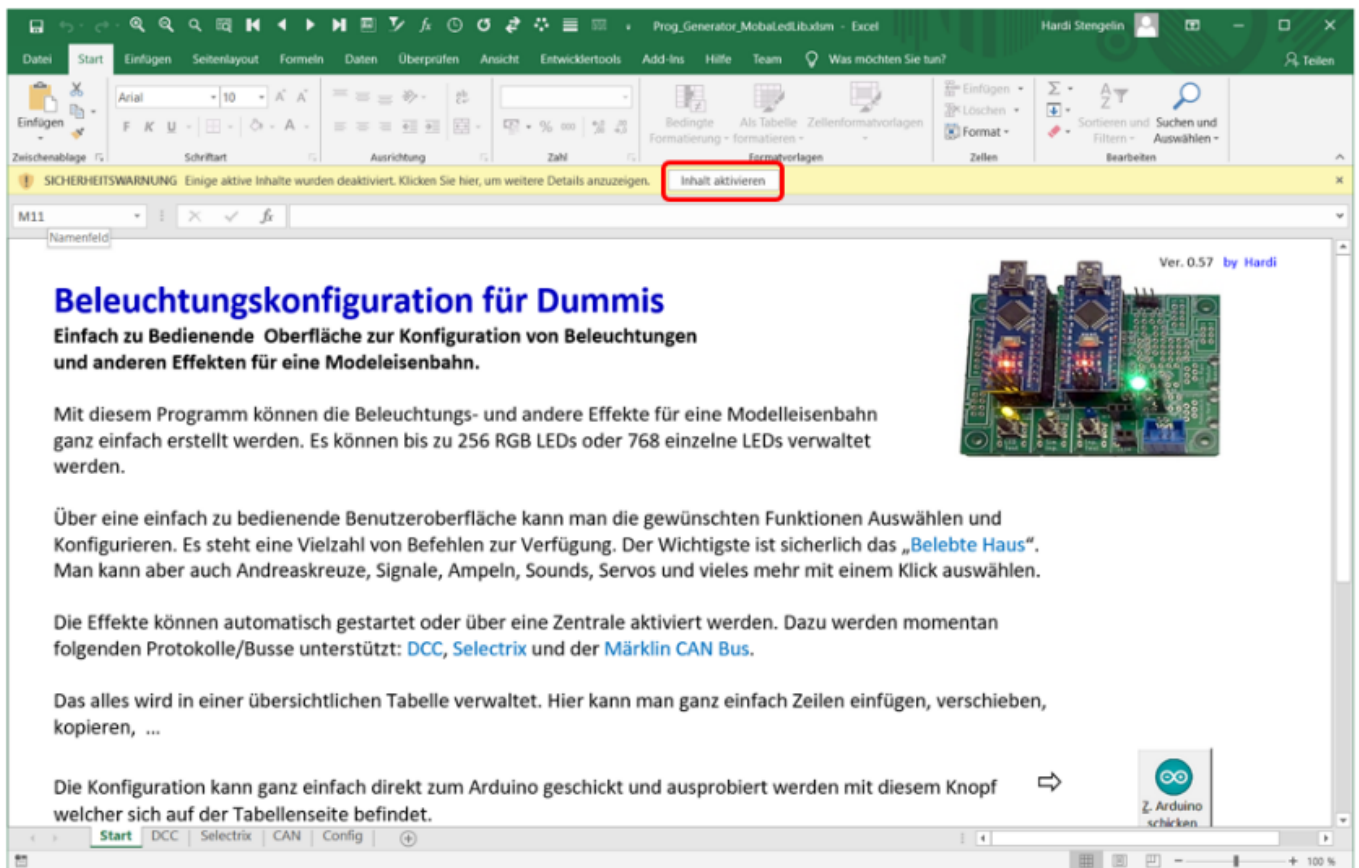
<https://wiki.mobaledlib.de/anleitungen/fehlermeldungen/fastled> Hilfestellungen.) Jetzt kann endlich das MoBaLedLib-Excel Programm geöffnet werden. Dazu auf der Tastatur die WINDOWS und die „R“ Taste gleichzeitig drücken. In den erscheinenden „Ausführen“ Dialog die folgende Zeile kopieren:

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

Achtung: Die Groß- und Kleinschreibung muss exakt stimmen.



Nun sollte sich endlich die aktuelle Version des Programm Generators der MobaLedLib öffnen.

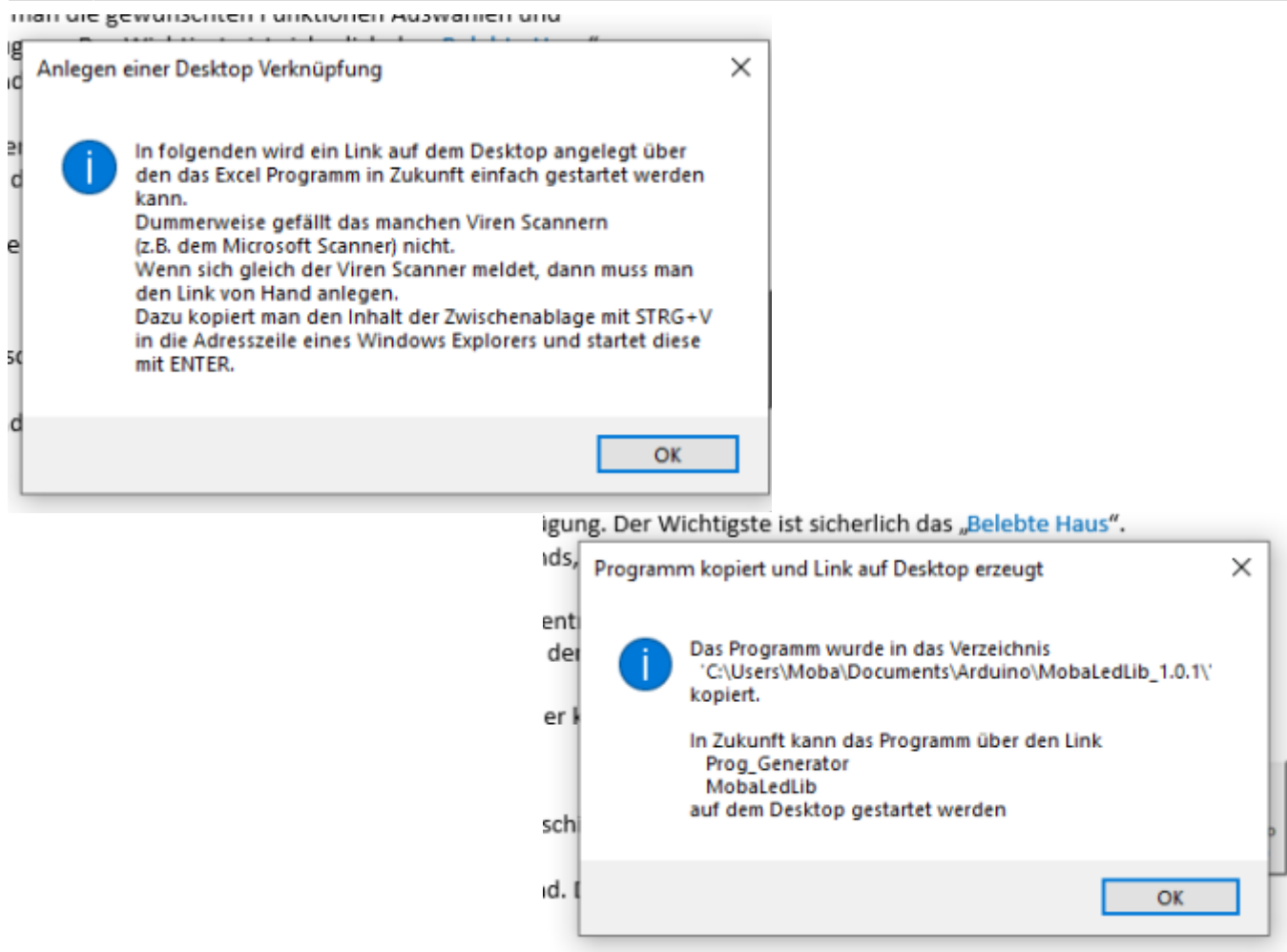


Da es sich um ein Excel Programm mit eingebetteten Makros handelt, wird beim ersten Start eine Sicherheitswarnung angezeigt. Mit einem Klick auf „Inhalte aktivieren“ werden die Makros ausführbar gemacht.

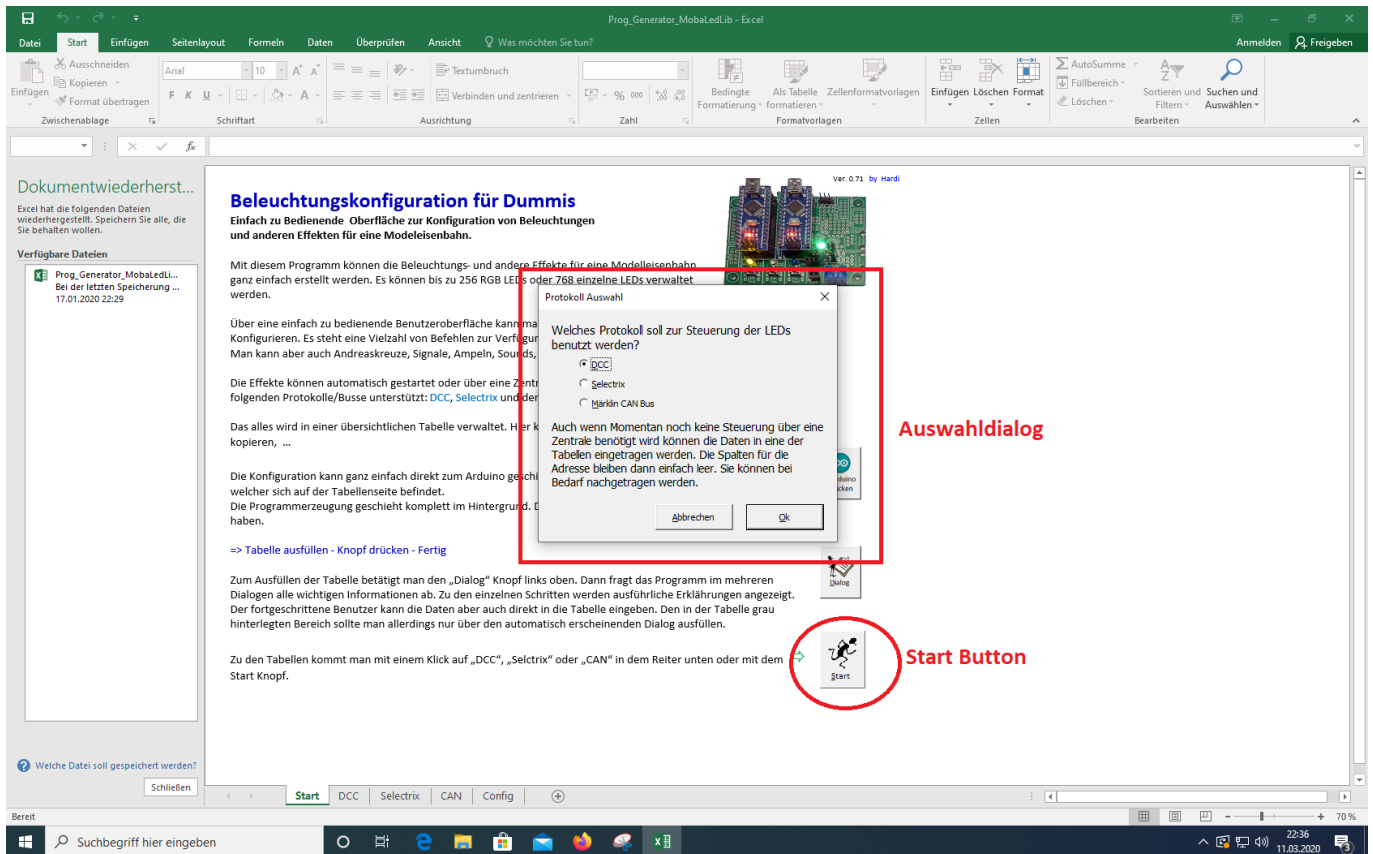
Danach wird das Programm automatisch in das Verzeichnis

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

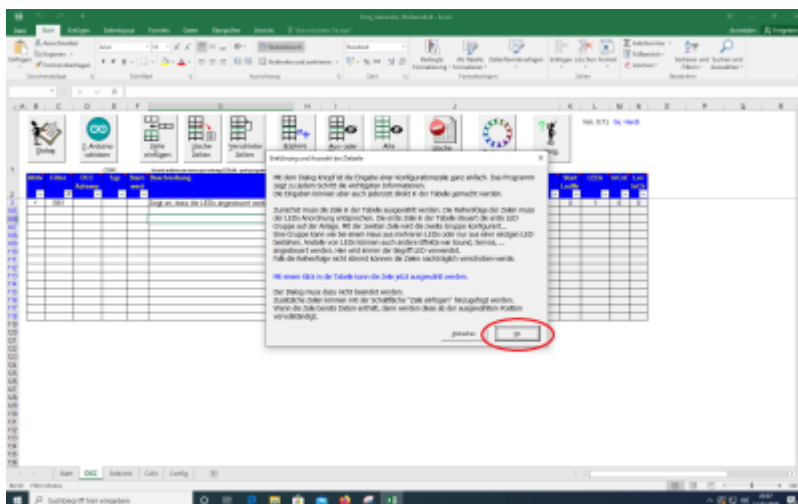
kopiert und ein Icon auf dem Desktop angelegt (x.y.z entspricht der Versionsnummer). Darüber kann der Prog-Generator künftig gestartet werden.

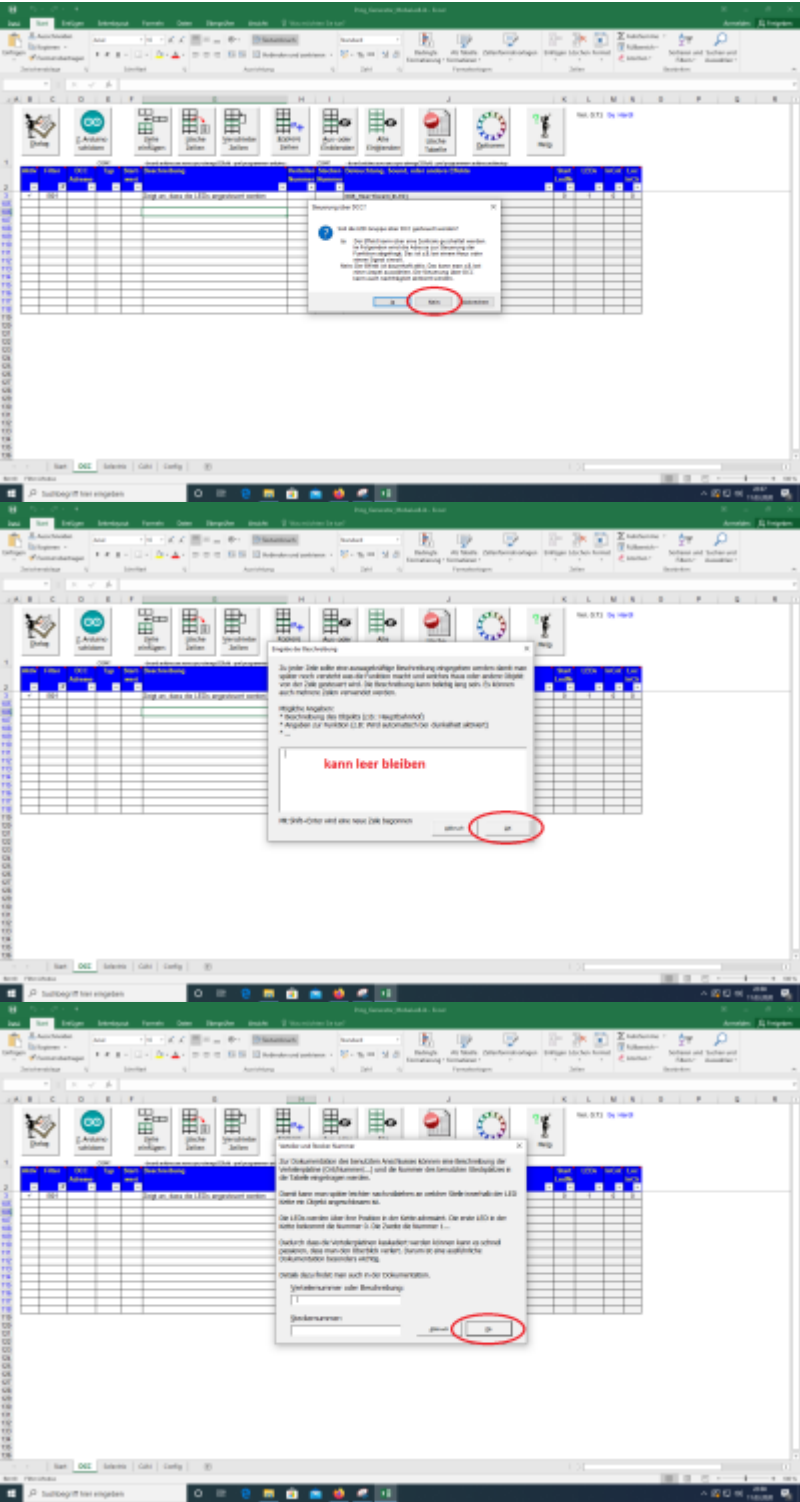


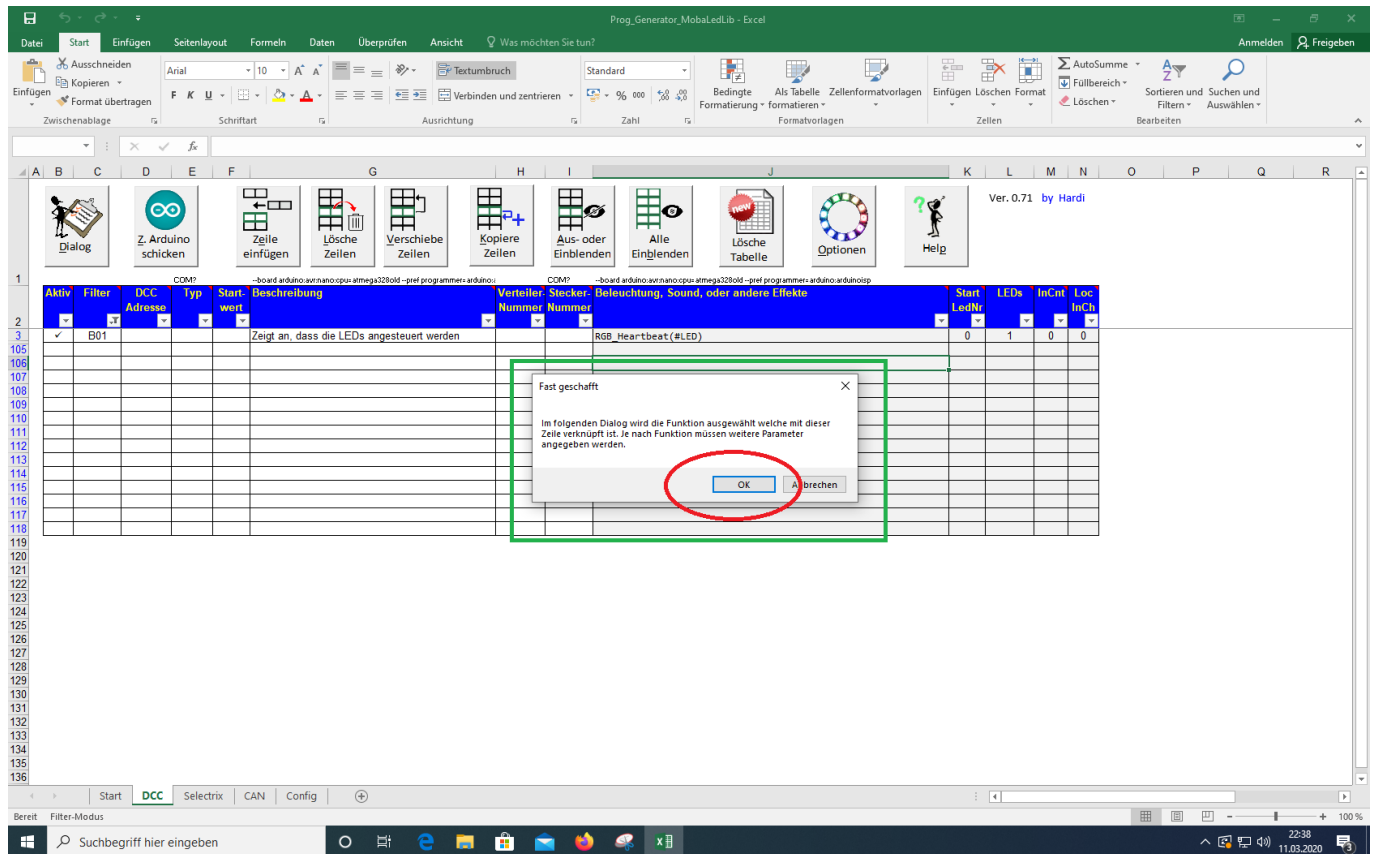
Ein Klick auf OK führt zu einem ersten Auswahldialog. Für erste Versuche ist es zunächst unerheblich welche der drei angebotenen Möglichkeiten (DCC-, Selektrix- oder CAN-Steuerung) ausgewählt wird.



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. \







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.
Leuchtfeuer	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

Abort

Select

Die ARDUINO Software von der ARDUINO Webseite herunterladen

Verbinde Deinen Raspberry Pi mit dem Internet, öffne den Webbrowser und gib die folgende URL ein:

<https://www.arduino.cc/en/Main/Software>

Du must die ARDUINO Software für LINUX für 32-bit ARM Prozessoren herunterladen. Vorsicht:

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

Permanent link:
<https://wiki.mobaledlib.de/anleitungen/spezial/pyprogramgenerator/raspberrypiinstallation?rev=1588700258>

Last update: 2020/05/05 18:37

