

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. It features the Arduino logo, the version number 1.8.12, and a description of the IDE. On the right, there are links for Windows installer, Windows ZIP file, Windows app, Mac OS X, Linux 32 bits, Linux 64 bits, Linux ARM 32 bits, Linux ARM 64 bits, Release Notes, Source Code, and Checksums. Below this, there are sections for Hourly Builds and Beta Builds. The entire screenshot is crossed out with a large red X.

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.

Previous IDE Releases

ARDUINO 1.8.11

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.
[See the release notes.](#)

Windows [Installer](#)
Windows [ZIP file for non admin install](#)

Mac OS X [10.8 Mountain Lion or newer](#)

Linux [32 bits](#)
Linux [64 bits](#)
Linux [ARM 32](#)
Linux [ARM 64](#)

Source

ARDUINO 1.0.6

Classic Arduino IDE, to be used with any Arduino board, but Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.
[See the release notes.](#)

Windows [Installer](#)
Windows [ZIP file for non admin install](#)

Mac OS X

Linux [32 bits](#)
Linux [64 bits](#)

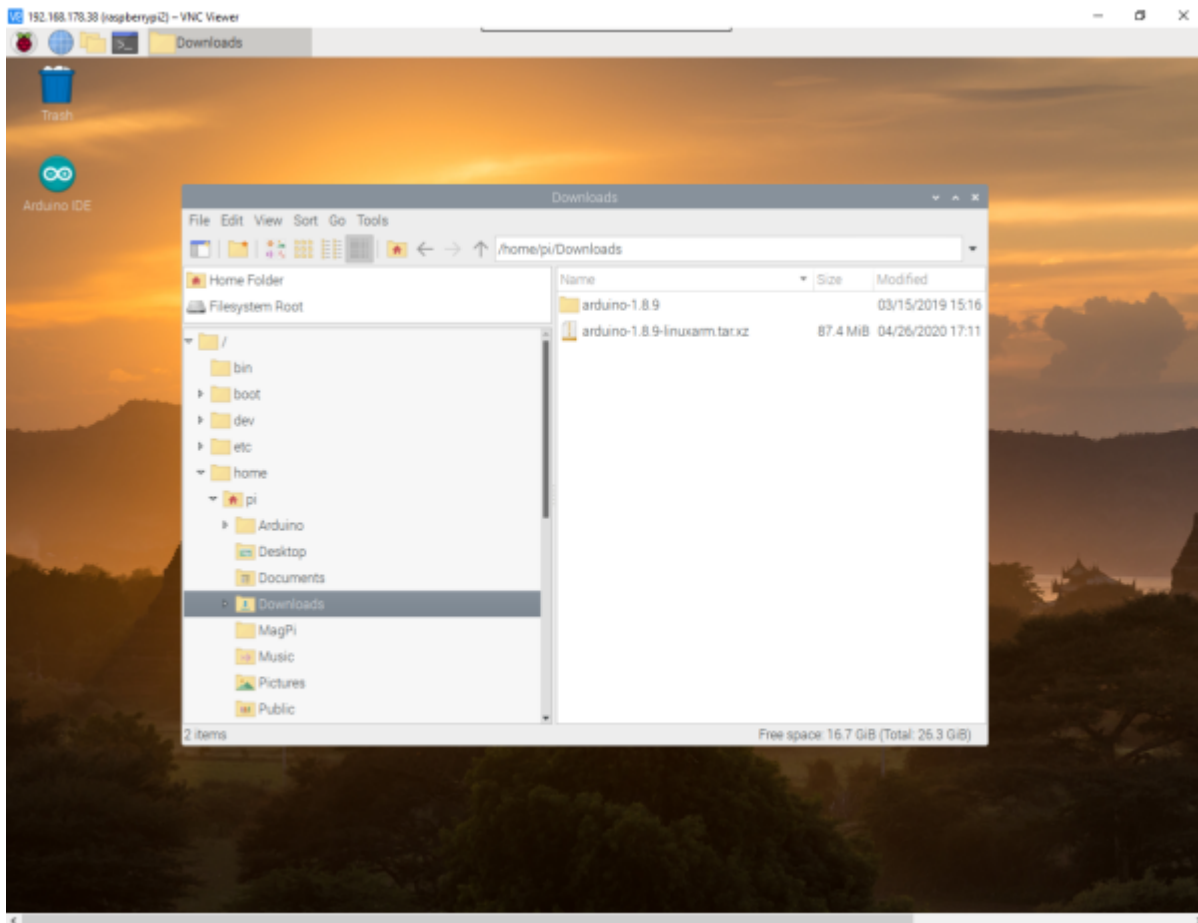
Source

Arduino 1.8.x, 1.6.x, 1.5.x BETA

These packages are no longer supported by the development team.

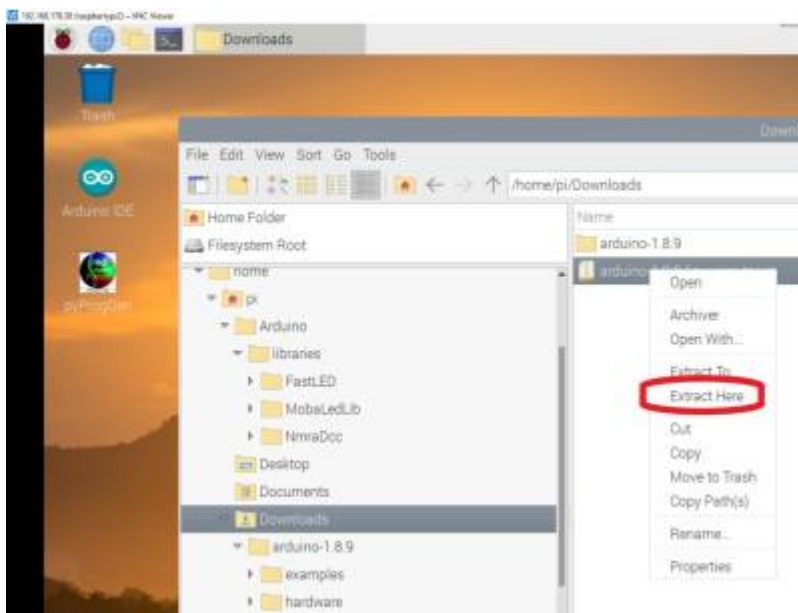
1.8.10	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit Linux ARM 32 Linux ARM 64	Source code on Github
1.8.9	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit Linux ARM 32 Linux ARM 64	Source code on Github
1.8.8	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit Linux ARM 32 Linux ARM 64	Source code on Github
1.8.7	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit Linux ARM 32 Linux ARM 64	Source code on Github
1.8.6	Windows	MAC OS X	Linux 32 Bit Linux 64 Bit	Source code

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.

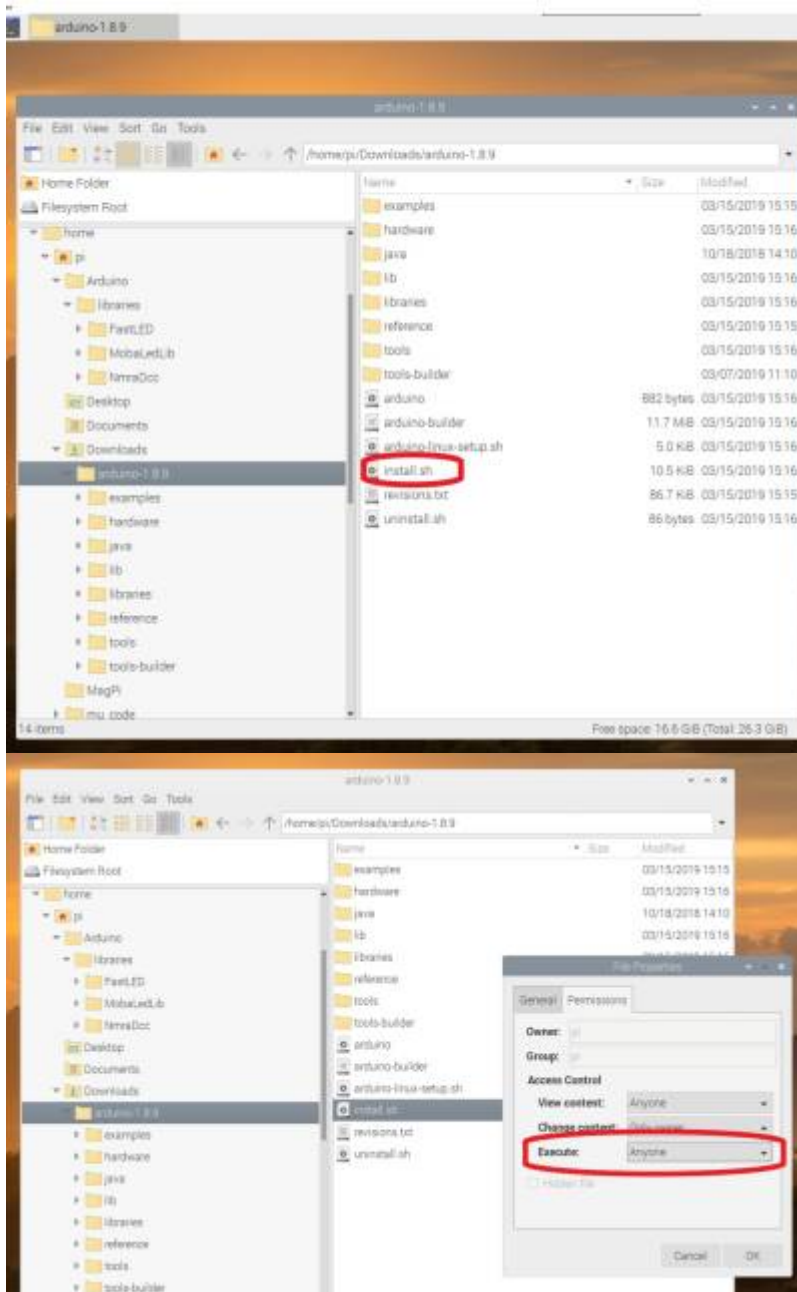


Zur Installation wird das Programm entpackt. Ich habe es gleich im Downloads Ordner entpackt. Ist nicht optimal, aber funktioniert.

Auf den Dateinamen Rechts-klicken und „Extract Here“ auswählen.



Nach dem Entpacken in das entpackte Verzeichnis wechseln und die Datei install.sh doppelt anklicken. Es kann sein, dass die Datei nicht ausgeführt werden kann. Dann auf die Datei Rechts-klicken. Die Eigenschaften (properties) auswählen. Und bei „Execute“ „Anybody“ auswählen.

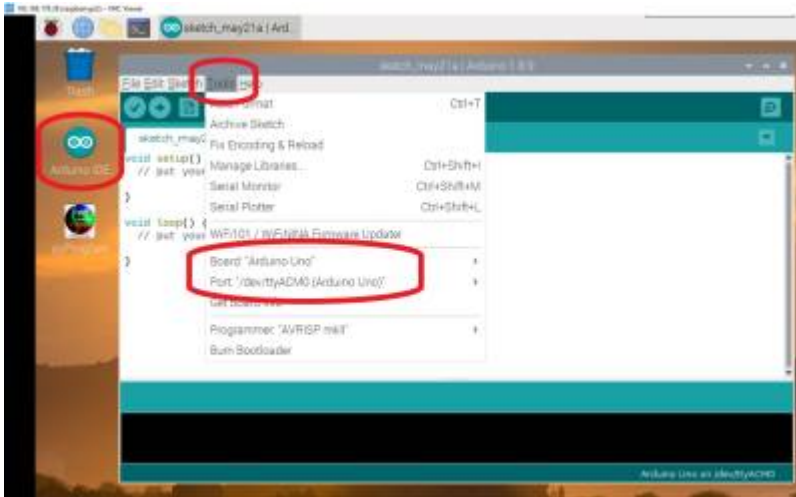


Das Installationsscript erzeugt ein Desktop Icon für die ARDUINO Benutzeroberfläche.

Schritt 2: Die MobaLedLib installieren

Die ARDUINO IDE aufrufen durch Doppelklick auf das ARDUINO-Icon auf dem Desktop.

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



Falls das nicht der Fall ist, sollte zunächst überprüft werden, ob der Arduino erkannt wird und an welcher Schnittstelle er angeschlossen wurde. Das können wir nachher mit dem PyProggenerator machen.

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](#).

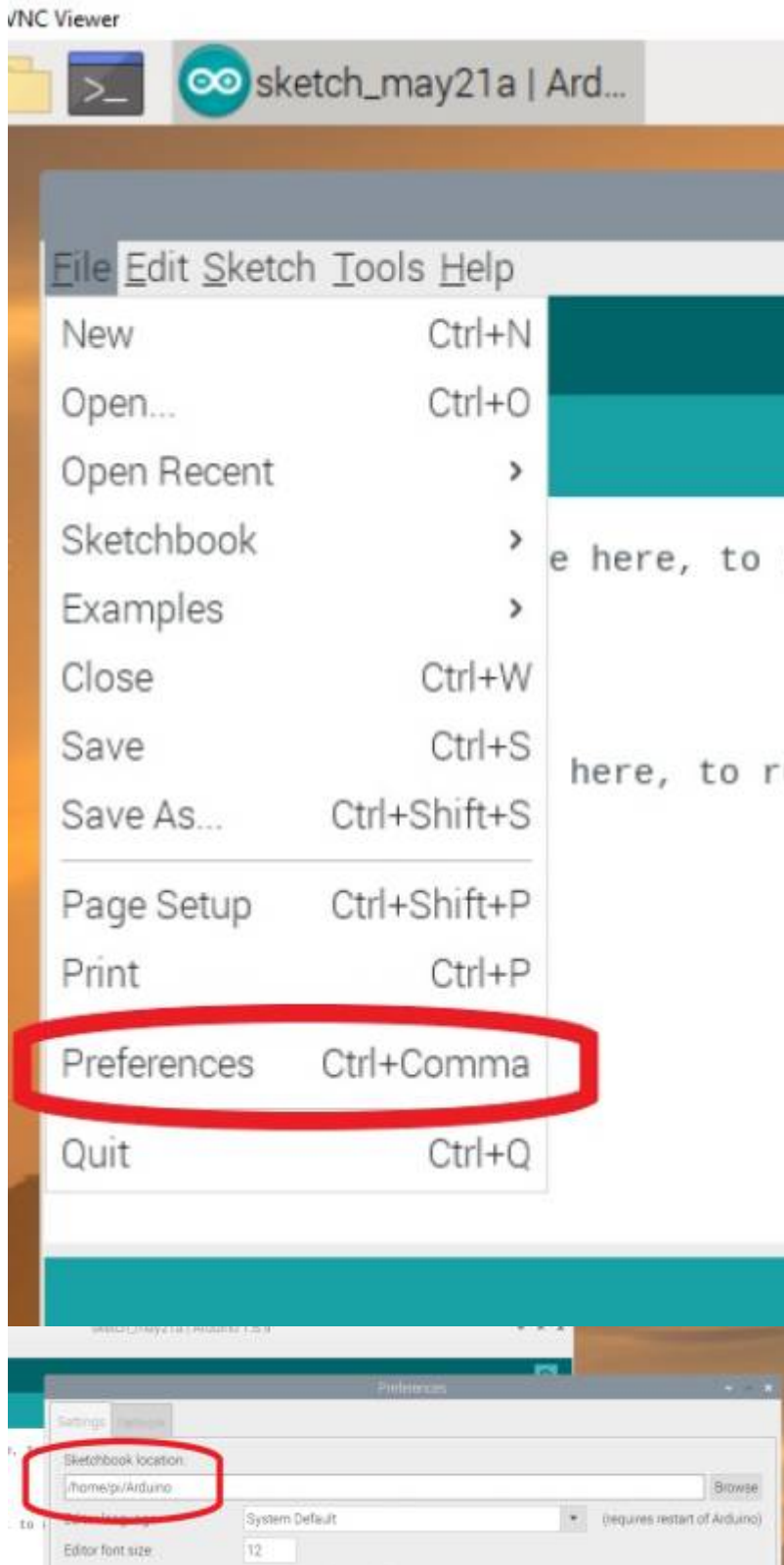
Für Einsteiger ist möglicherweise ein kleiner zusätzlicher Test interessant. Mehr dazu am Ende der Anleitung unter [Test](#).

Überprüfung Sketch-Ordner

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

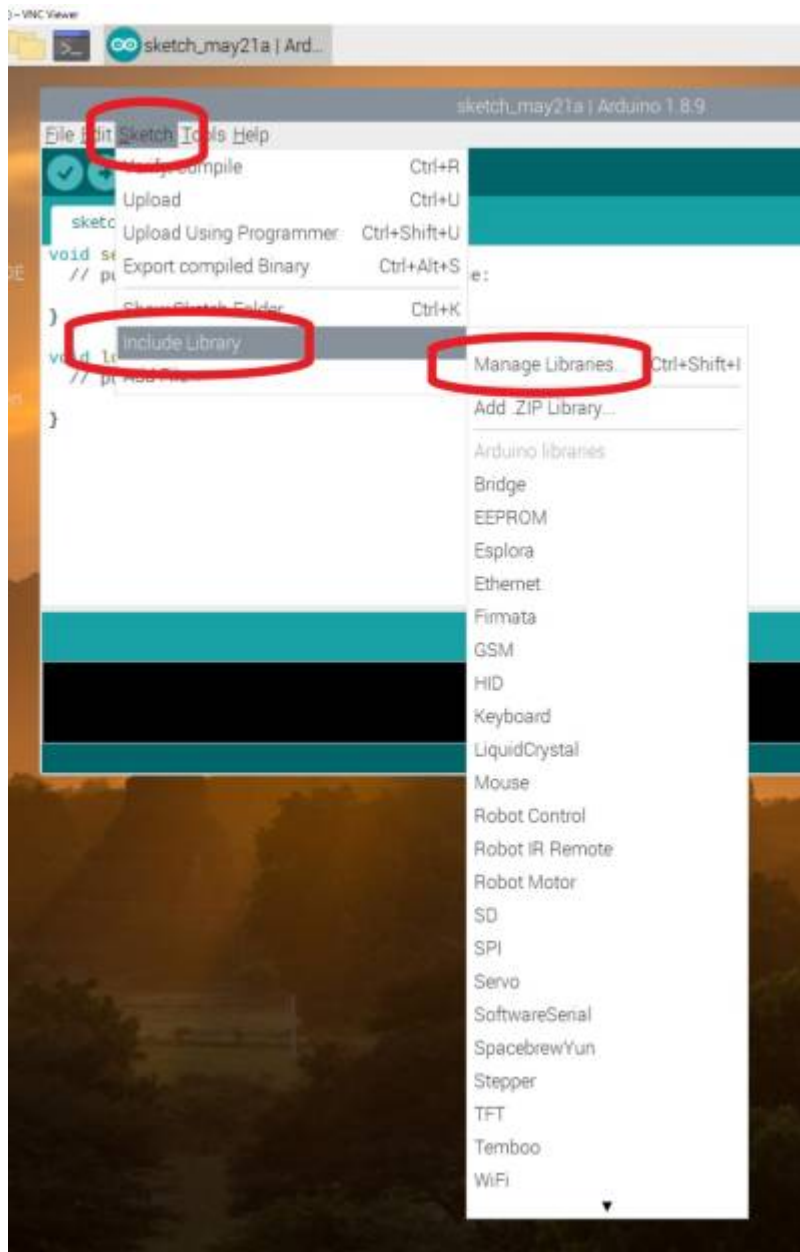
```
/home/pi/Arduino
```

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



MobaLedLib (MLL) einrichten

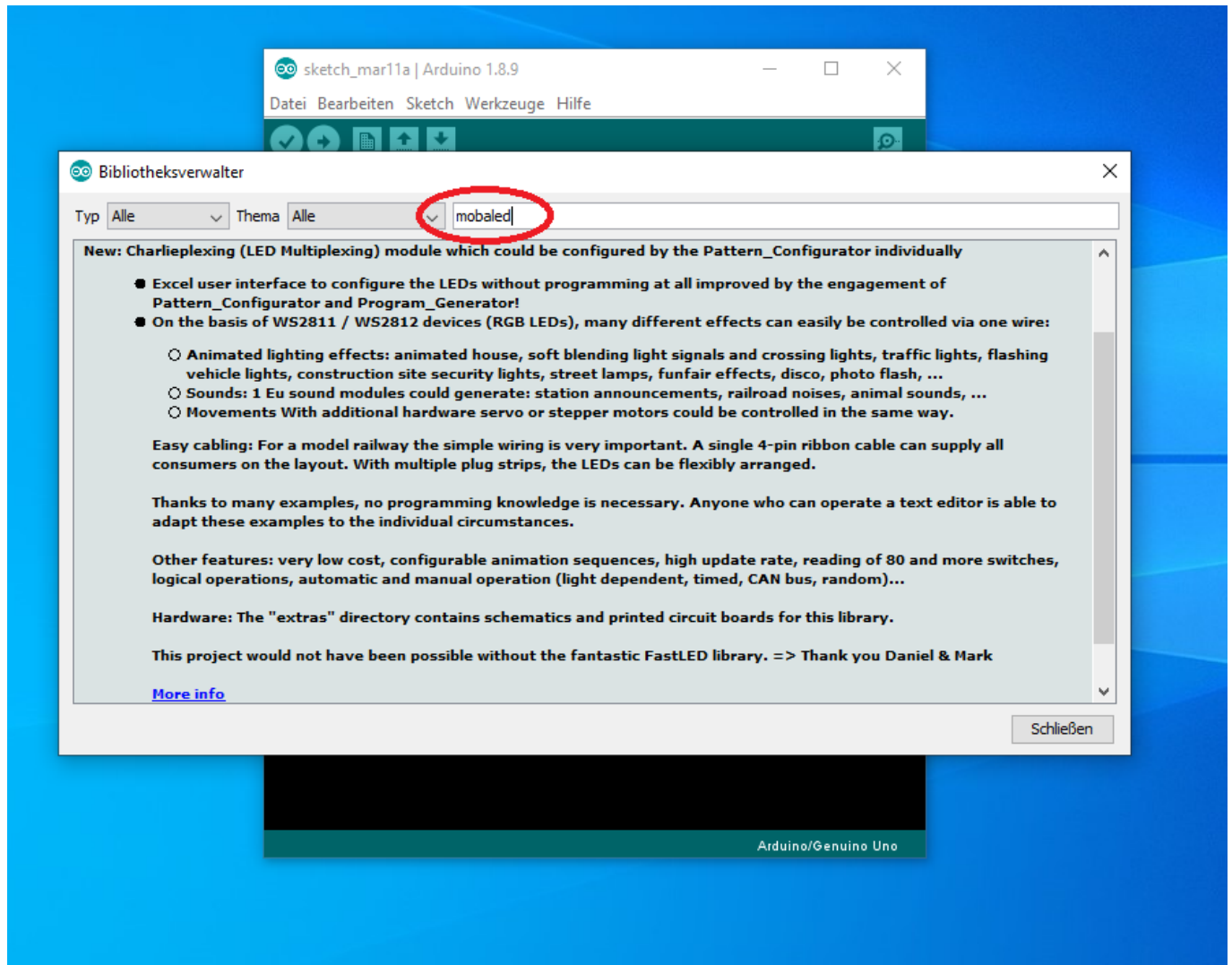
Zunächst muss die Bibliotheksverwaltung der IDE über Sketch → Bibliothek einbinden → Bibliothek verwalten geöffnet werden.



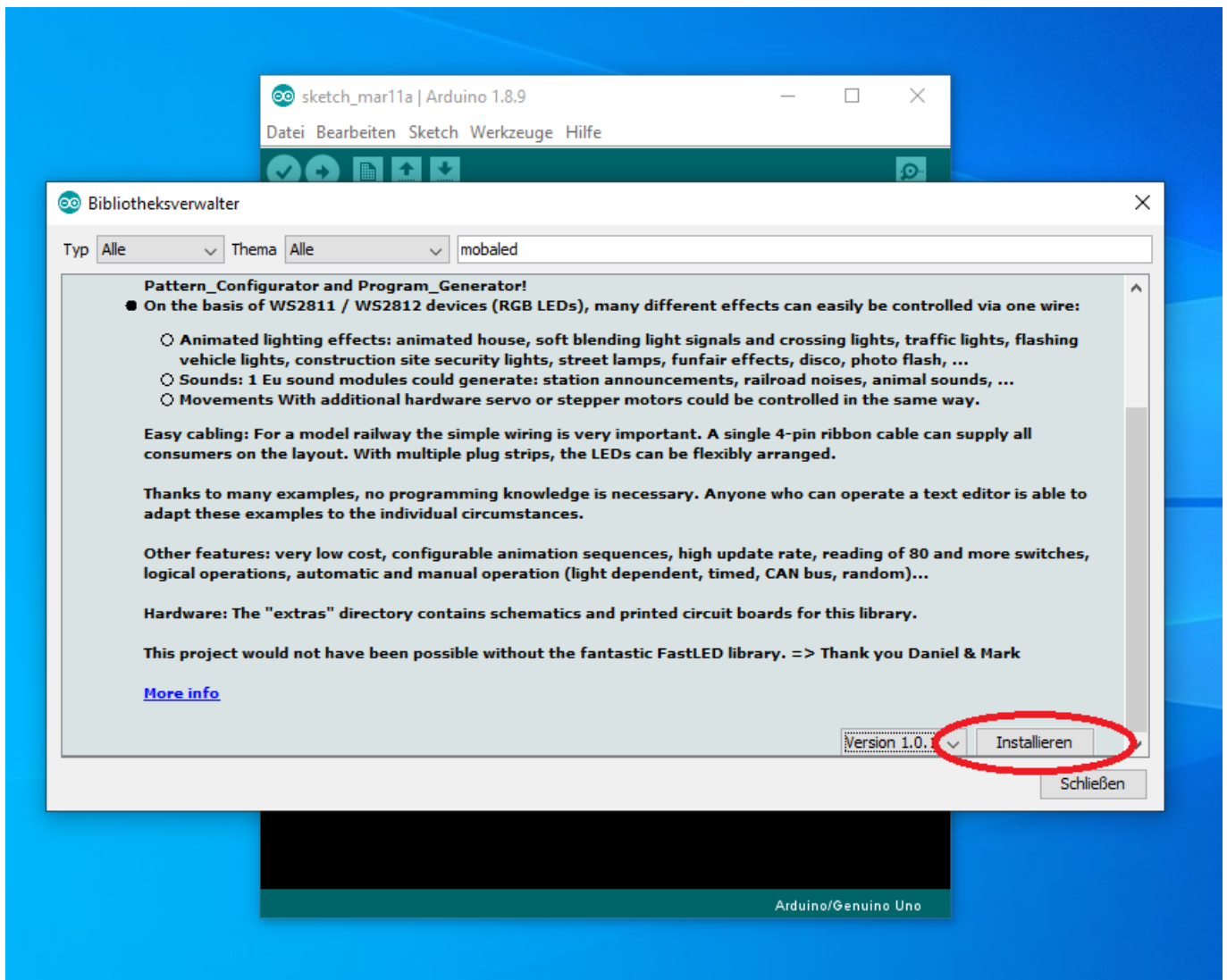
Es öffnet sich der Bibliotheksverwalter.



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.

Aussdem fehlt noch die NMRA-DCC-Komponente, wenn man den DCC ARDUINO benutzen will.

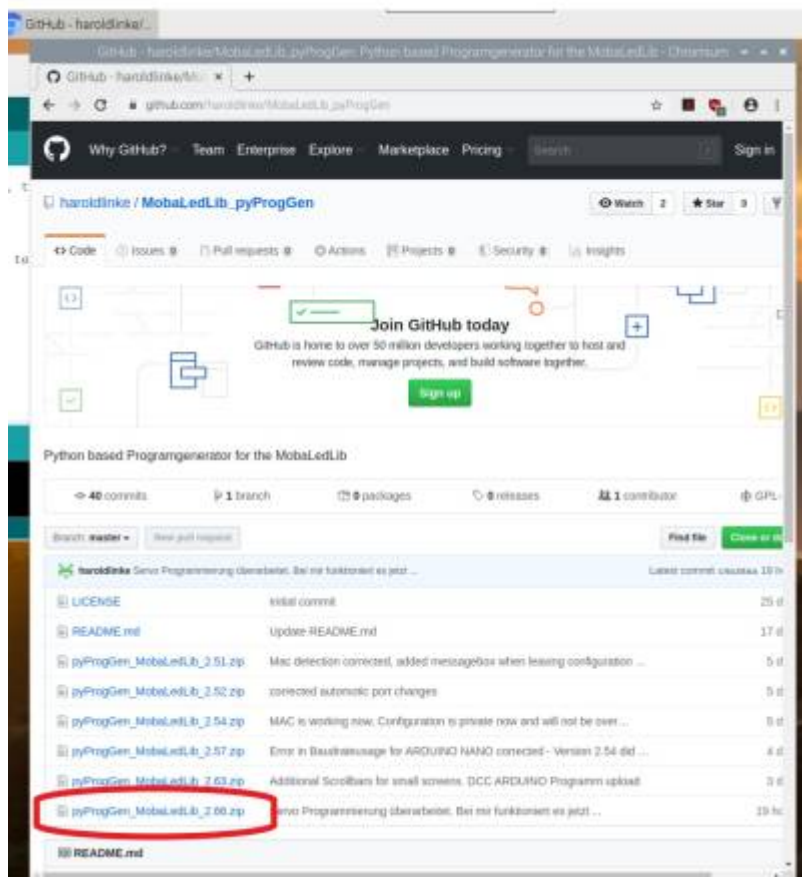
Bitte nun noch mit der gleichen Vorgehensweise in das Suchfeld „nmradcc“ eingeben und installieren.

(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.)

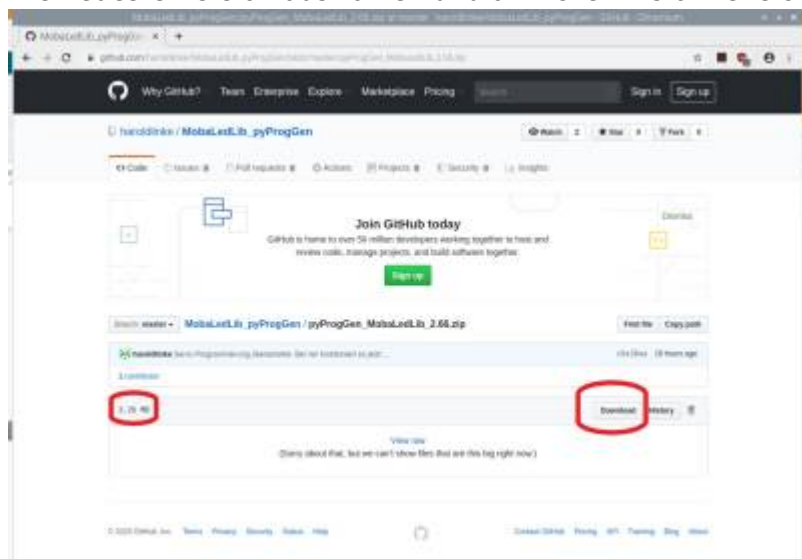
PyProgGen installieren

Die neueste Version des pyProgGen findet Ihr in GitHub unter:

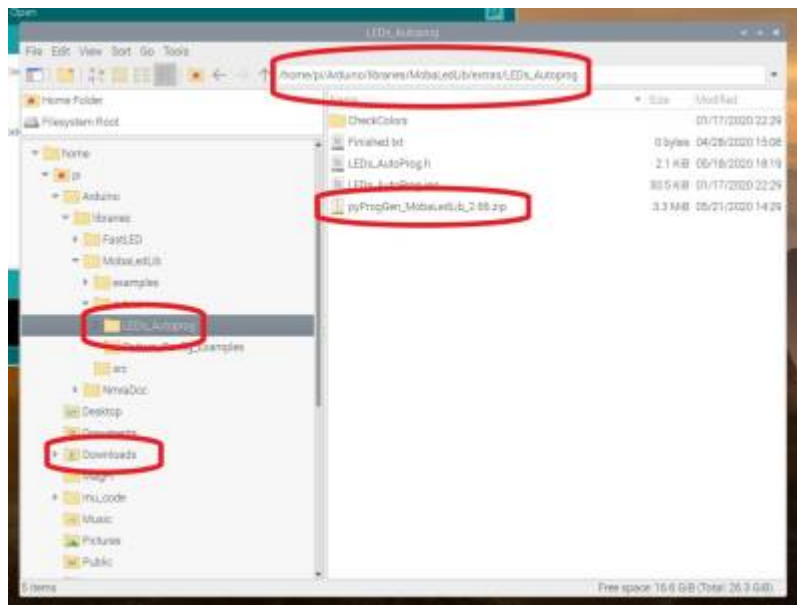
https://github.com/haroldlinke/MobaLedLib_pyProgGen



Die neueste Version auswählen und anklicken. Es öffnet sich das Download Fenster.

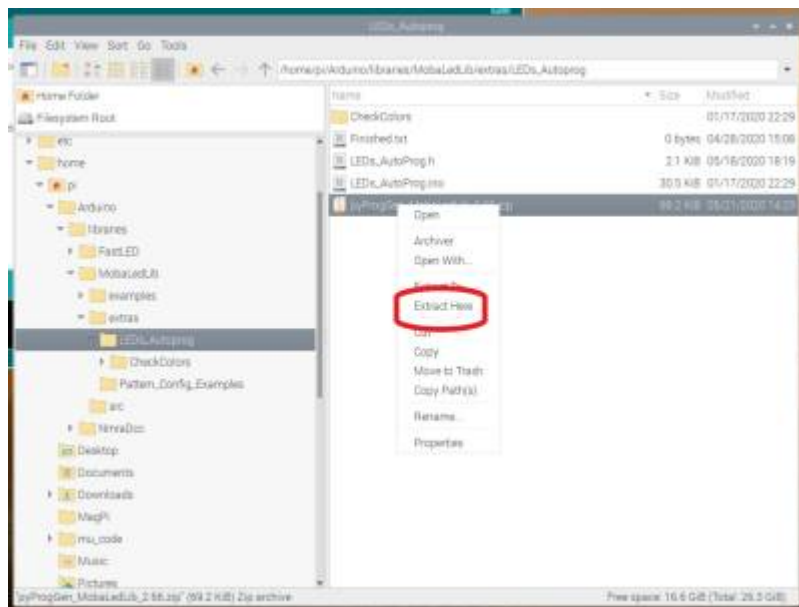


Den „Download“-Button klicken. Die gespeicherte Datei befindet sich danach im Ordner „Downloads“. Diese Datei von dort in das Verzeichnis `home/pi/Arduino/libraries/MobaLedLib/extras/LEDs_Autoprogramm`



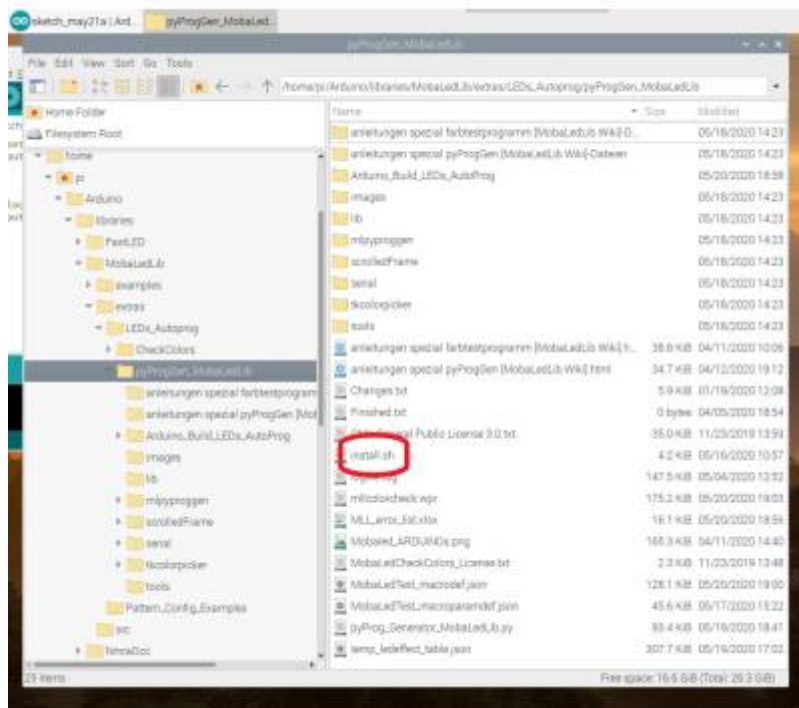
verschieben.

Die Datei dort entpacken (rechts-klick - Extract Here)

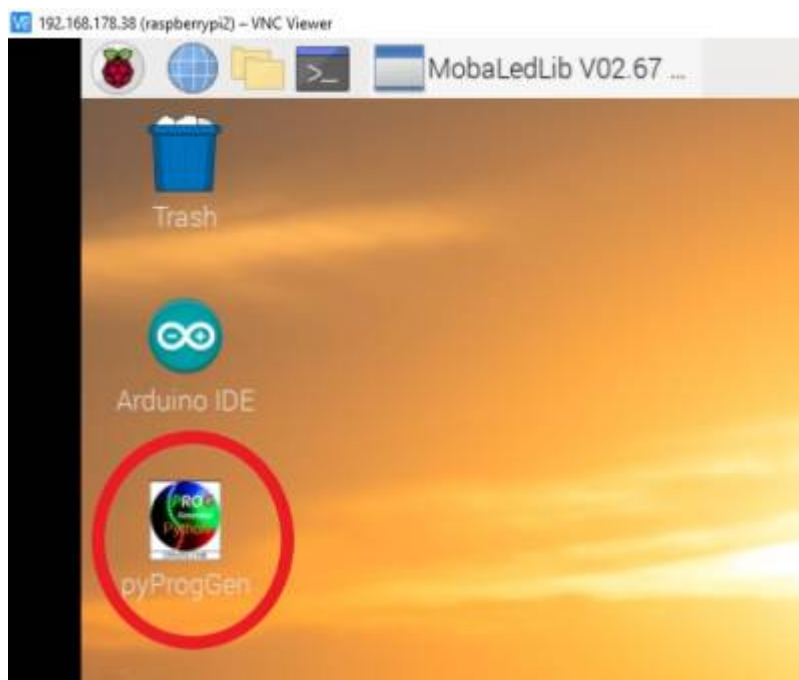


Es wird ein Verzeichnis „pyProgGen_MobaLedLib“ erzeugt.

In das Verzeichnis wechseln und die Datei install.sh ausführen. (Vorher die Berechtigung zum Ausführen setzen, wie oben beschrieben)



Das Script erzeugt ein Icon auf dem Desktop des Raspi.



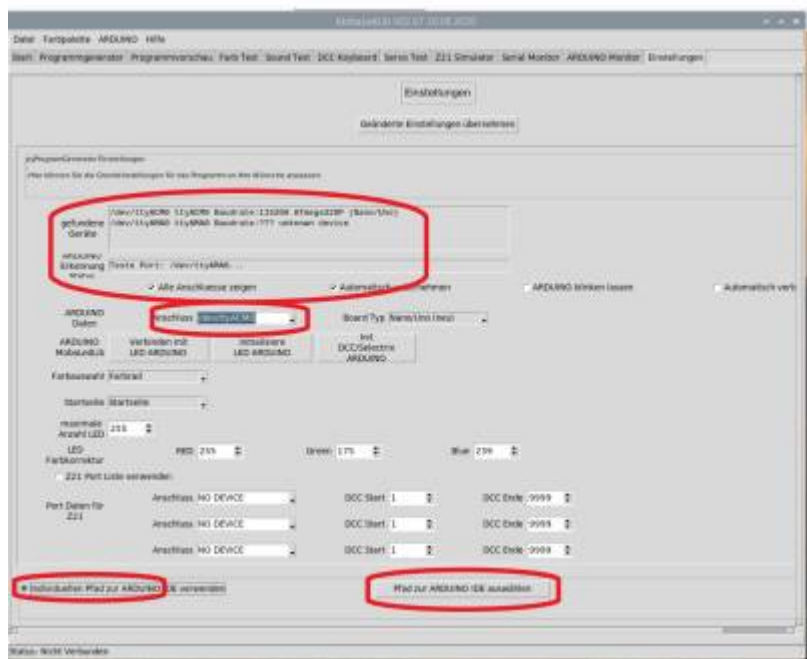
Das Icon doppelt-klicken und nach einiger Zeit erscheint das Startfenster von PyProgGen.



Herzlichen Glückwunsch. Du hast das Programm erfolgreich installiert.

PyProgGen Einstellungen anpassen

Der wichtigste erste Schritt um mit pyProgGen auf dem Raspberry arbeiten zu können ist die Auswahl der Schnittstelle für den ARDUINO und die Auswahl des Pfades zur ARDUINO IDE.

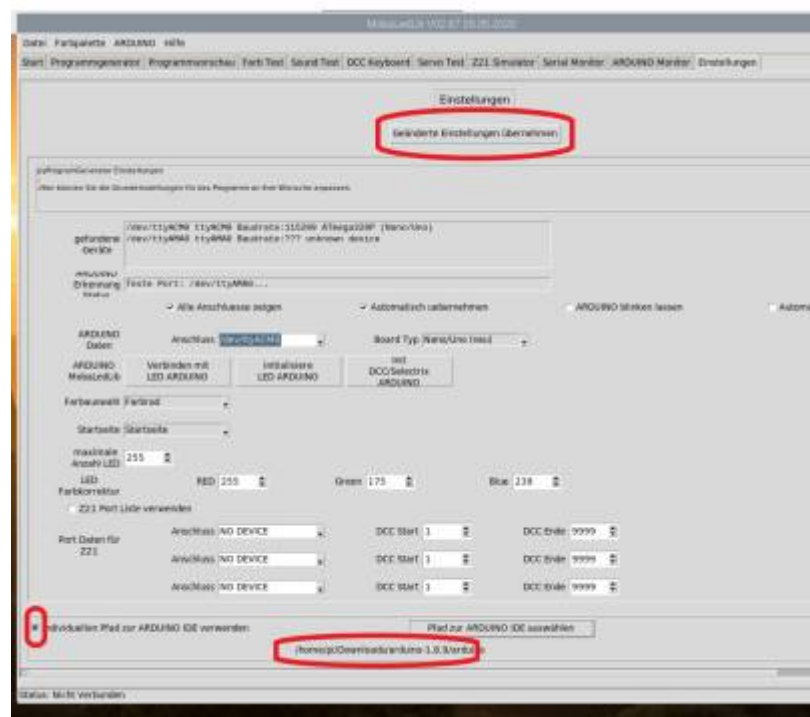
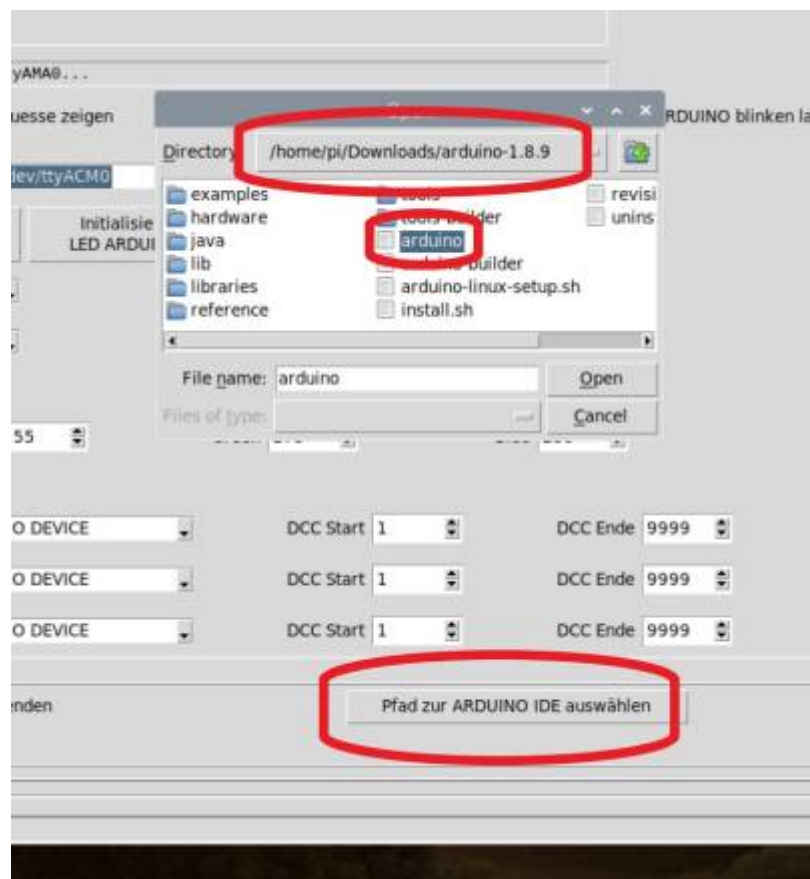


Den Reiter „Einstellungen anklicken“:

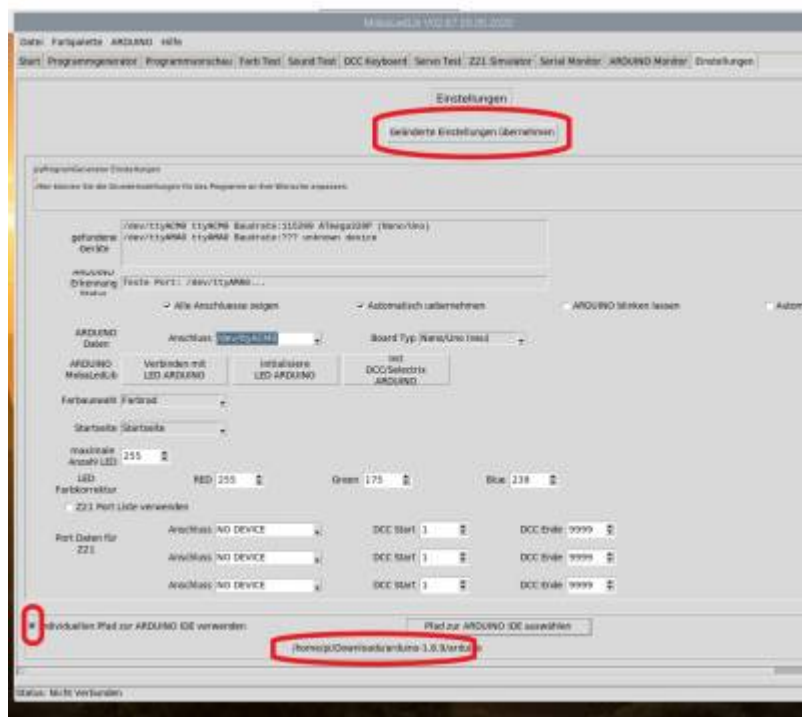
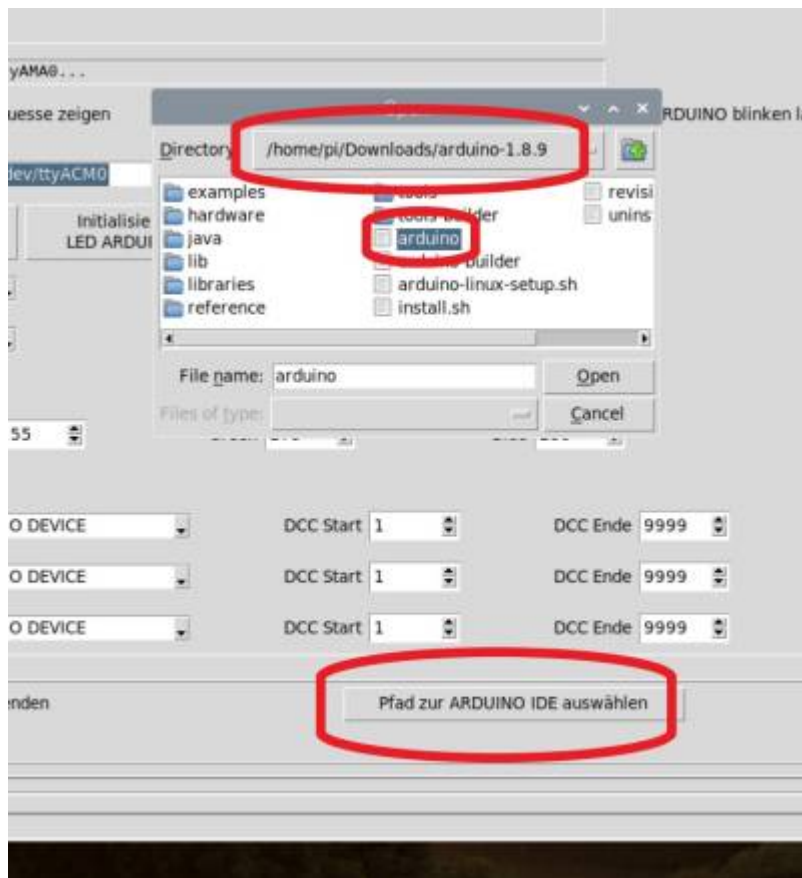
Im Feld gefundene Geräte werden alle Geräte angezeigt, die eine serielle Schnittstelle benutzen. Wenn der ARDUINO angeschlossen ist, wird hier der ARDUINO mit seiner Bezeichnung und der Schnittstelle angezeigt.

Wenn „Automatischübernehmen“ angeklickt ist, werden die Daten des ARDUINO automatisch eingestellt. In bestimmten Fällen kann es vorkommen, dass der Anschluss nicht automatisch übernommen wird, dann kann man ihn von Hand einstellen.

Im unteren Bereich des Fensters muss der Pfad zur ARDUINO IDE eingestellt werden, da das Programm dies nicht automatisch finden kann.



Nicht vergessen, die Checkbox „Individuellen Pfad zur ARDUINO IDE zu aktivieren, und die geänderten Einstellungen zu speichern!“



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

Permanent link:
<https://wiki.mobaledlib.de/anleitungen/spezial/pyprogramgenerator/raspberrypiinstallation?rev=1590066455>

Last update: 2020/05/21 14:07

