

Aussetzen einer Raspberry PI Umgebung für pyMLL (MobaLedLib)

1. Hardware

Folgende Hardware wird benötigt

- Raspberry PI 4 Model B
- SSD-Karte zwischen 32 GB und 128 GB
- Netzteil mit USB C, 5V und mind. 3 A Leistung

2. Installation Raspberry

Für die Grundinstallation der SSD Karte wird Raspberry PI Imager genutzt

<https://www.raspberrypi.com/software/>

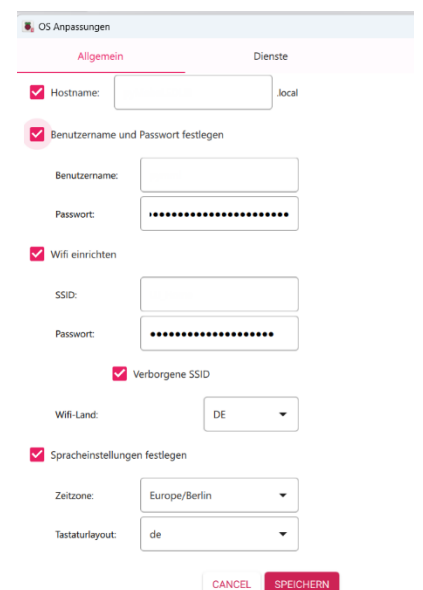
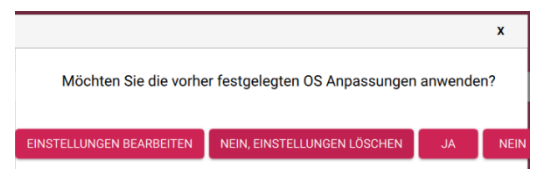
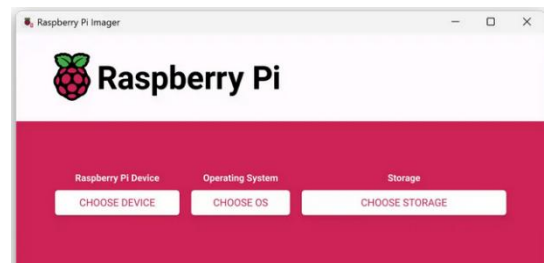
Unter **Choose Device** wird das Modell ausgewählt – Raspberry PI 4 Model B

Unter **Choose OS** wird die Version Raspberry PI OS (64-bit) ausgewählt

Unter **Choose Storage** wird das Laufwerk mit der SSD Karte ausgewählt

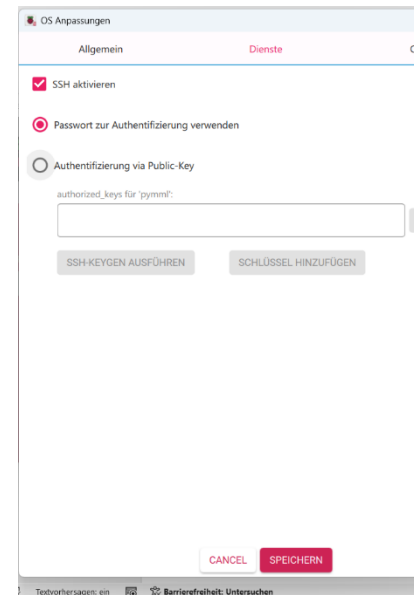
Danach sollten die folgenden Anpassungen vorgenommen werden.

Unter dem Punkt **Einstellungen bearbeiten** im Reiter **Allgemein** sollten alle Punkte ausgefüllt werden. Das ist erforderlich um den Raspberry später ohne aufwendiges anschließen von Tastatur und Monitor per RDP einrichten zu können.



Im Reiter **Dienste** sind SSH aktivieren und Passwort zur Authentifizierung verwenden auszuwählen. Beide Punkte müssen ausgewählt werden, um später mit RDP auf die graphische Oberfläche zu kommen.

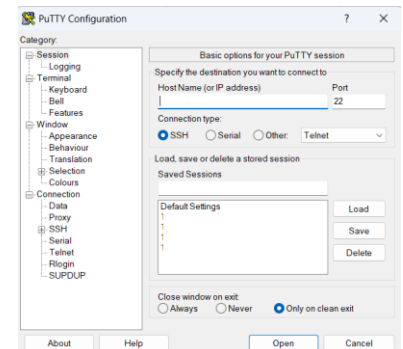
Danach kann die Installation der SSD Karte gestartet werden.



3. Erster Start Raspberry

Nachdem der Raspberry das erste Mal gestartet wurde erfolgt der Zugriff zunächst mit Putty. Putty kann von der folgenden Web-Seite geladen werden: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Im Putty wird die IP Adresse oder der Host-Name eingetragen. Als Port wird 22 ausgewählt und SSH muss aktiviert sein.



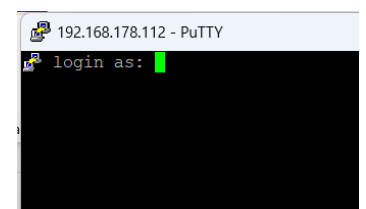
Wenn Putty die Verbindung hergestellt hat, können der User / Passwort eingegeben werden.

Danach sind folgende Schritte erforderlich:

Zunächst das System aktualisieren / `sudo apt-get update`

Dann folgt die Installation von xrdp / `sudo apt-get install xrdp`

Die Frage wird mit „Y“ beantwortet



Wenn der xrdp daemon sauber installiert ist sollte er auf folgenden Befehl mit Active: ative (running) antworten. Mit STRG-C Abfrage beenden / `sudo systemctl status xrdp`

Als nächste wird die Änderung der Bootreihenfolge vorgenommen. Das ist zwingend erforderlich, da sonst der Remote Desktop Zugriff nicht funktioniert.

Configurationsmenü aufgerufen / `sudo raspi-config`

Wenn das Menü aufgerufen wurde folgende Schritte durchführen:

S1 System Operation – „enter“

S6 Auto Login – „enter“

Would you like to automatically log in tot he console? <No> - „enter“

Console autologin ist disabled <Ok> [ggf. ein zweites Mal] <Finish> - enter

Would you like to reboot now? <Yes> - „enter“

Danach erfolgt ein Reboot

Mit folgendem Befehl kann er auch von der Console aus durchgeführt werden

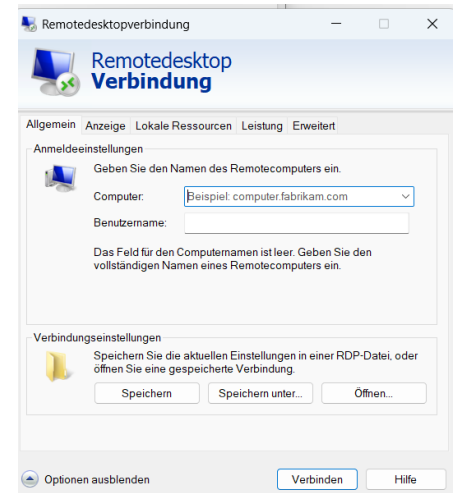
/ `sudo shutdown -r now`

Der Raspberry startet danach automatisch neu.

4. Start der RDP Verbindung

Mit der von Microsoft mitgelieferten Software Remotedesktop kann (wenn die Rechner im gleiche WLAN sind) nun eine RDP-Verbindung aufgebaut werden.

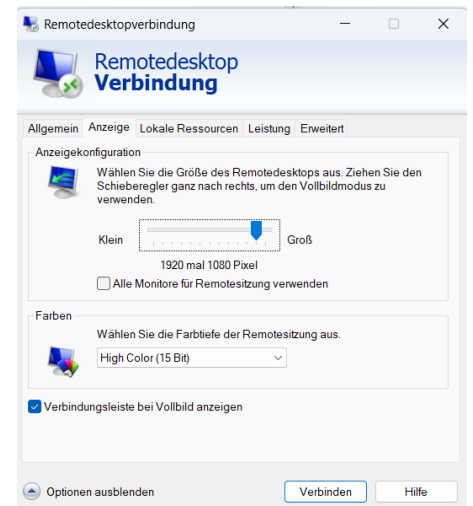
Dazu wird die IP Adresse oder der Computername eingetragen. Der Username kann in dieser Ansicht vorbelegt werden.



Danach die **Optionen einblenden** und auf den Reiter **Anzeige** wechseln

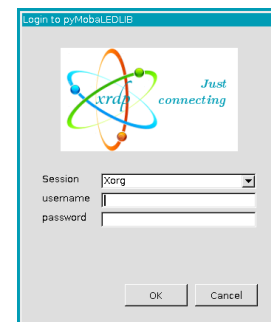
Hier muss unter Farbe **High Color (15 Bit)** eingestellt sein.

Danach kann über **Verbinden** eine RDP Verbindung zum Raspberry gestartet werden.



5. Erste RPD Verbindung

Bei der Anwahl erfolgt als erstes die Abfrage von Username / Passwort.



Danach startet der Desktop des Raspberry.

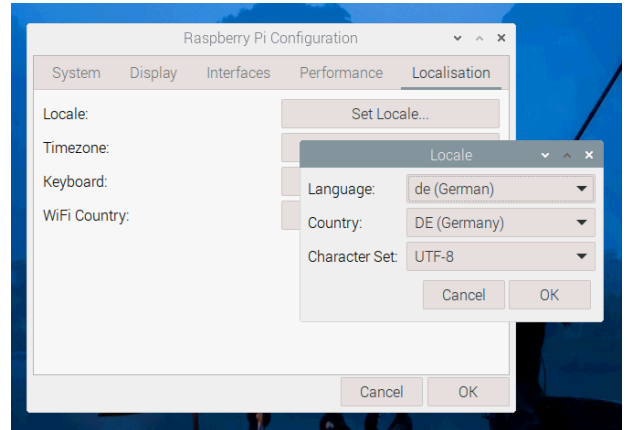
Nach jedem Reboot erfolgt im Desktop eine zweite Abfrage des Passwortes.



Wer möchte kann jetzt unter Raspberry Pi Configuration Anpassungen wie z.B. Browser und die länderspezifischen Einstellungen vornehmen.

Wichtig: Bei den länderspezifischen Einstellungen muss bei Auswahl von LOCAL – LANGUAGE: de(Germany) und COUNTRY: DE(Germany) zwingend der CHARACTER SET: **UTF-8** eingestellt sein.

Danach ist ein Reboot erforderlich.

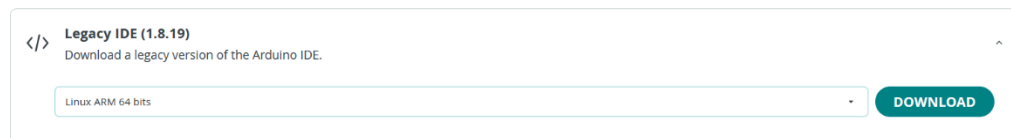


6. Installation der Arduino IDE

Die Software wird von folgender Web-Seite geladen:

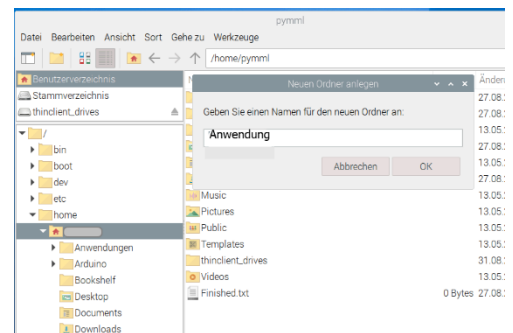
<https://www.arduino.cc/en/Main/Software>

Es muss die Version Legacy IDE (1.8.9) Linux ARM 65 bits (das Raspberry auch 64 Bit ist) geladen werden.



Den Download finden man nach Abschluss im Verzeichnis: /home/<Benutzername>/Downloads.

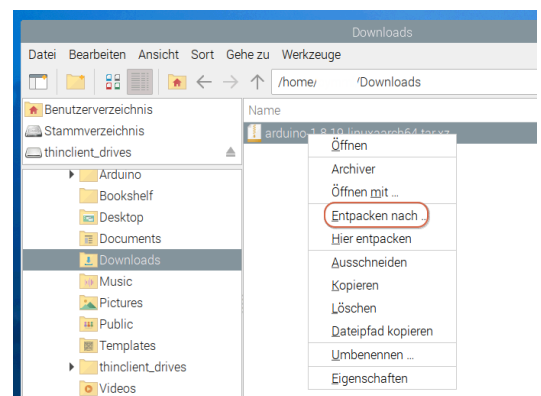
Zunächst legen wir ein Verzeichnis /home/<Benutzername>/Anwendungen an. Das erfolgt mit dem Dateimanager.



Die Datei *arduino-1.8.19-linuxaarch64.tar.xz* wird in das Verzeichnis entpackt.

Dazu nutzen wir den Dateimanager und navigieren in das Verzeichnis **.../Downloads**.

Über die rechte Mouse-Taste bekommen wir das Kontextmenü in dem wir **Entpacken nach** auswählen.



Dann wechseln wir in das Verzeichnis

.../home/<Benutzername>/Anwendungen/arduino-1.8.19 und starten (ausführen) die Datei **install.sh**.

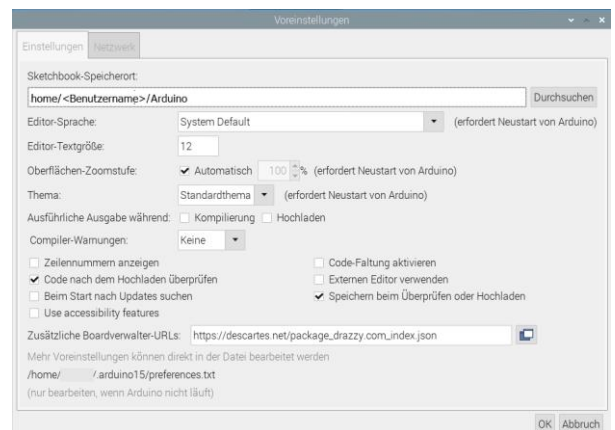
Danach erscheint auf dem Desktop das Arduino Icon auf dem Desktop.

7. Erster Start Arduino

Arduino IDE wird über das Icon auf dem Desktop gestartet. Es erfolgt eine Abfrage bei der wir **Ausführen** wählen

Zunächst kontrollieren wir die Voreinstellungen. Dazu wählen wir unter **DATEI – VOREINSTELLUNG – aus**.

Als Sketchbook-Speicherort sollte **home/<Benutzername>/Arduino** eingetragen sein.



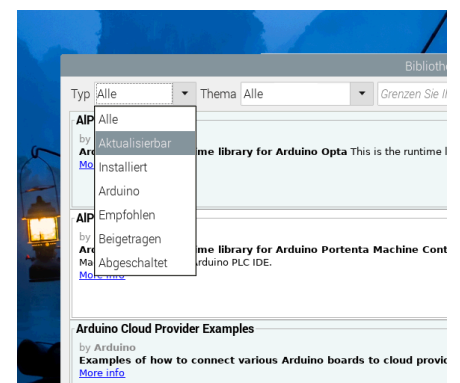
Unter **Zusätzliche Boardverwaltungs-URL** kann die folgende Web-Adresse https://descartes.net/package_drazzy.com_index.json eingetragen werden.

Danach wird unter **SKETCH – BIBLIOTHEK – BIBLIOTHEKVERWALTER** aufgerufen (das nimmt etwa Zeit in Anspruch).

Wenn die Liste aktualisiert wurden **TYP = AKTUALISIERBAR** auswählten (und warten).

Alle dort aufgeführten Bibliotheken sollten dann über Update aktualisiert werden. Keine zusätzlich angebotenen Bibliotheken installieren.

Das Fenster sollte danach leer sein.



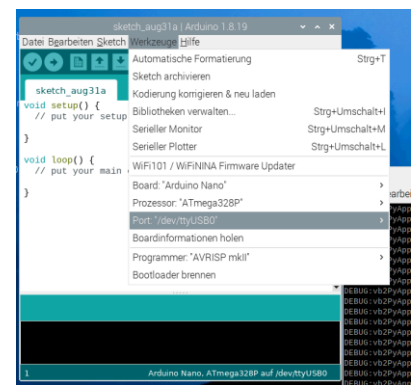
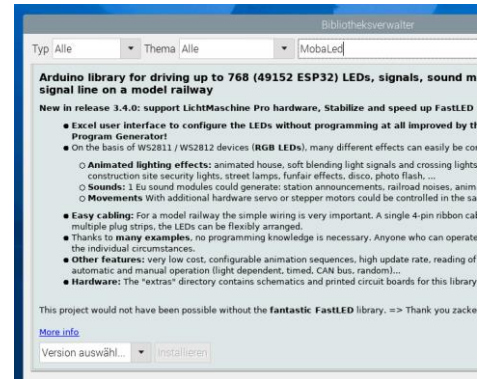
Wir wechseln wieder auf **TYP =ALLE** und Suchen (Suche eingrenzen) MobaLedLib (dauert auch etwas).

MobaLedLib wird in der aktuellen Version 3.4.0 mit allen weiteren angebotenen Library installiert.

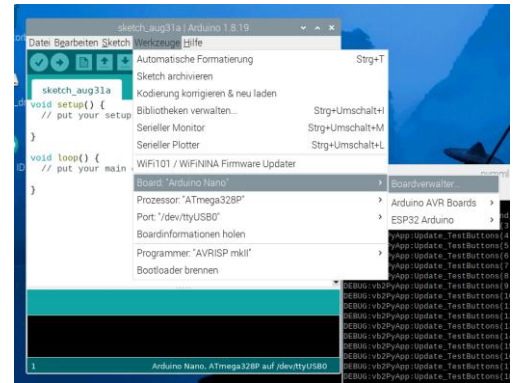
Dann kann das Fenster geschlossen werden.

Nun kann ein Nano angeschlossen werden.

Wir wechseln zu **WERKZEUGE** und richten dort das Board „Arduino Nano“ und den Port /dev/ttyUSB0 ein.

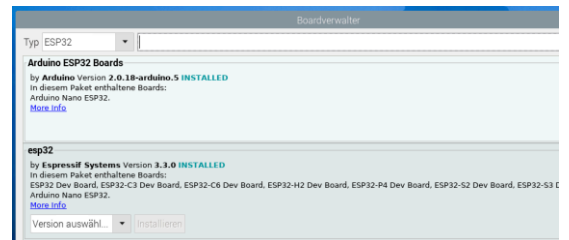


Dann rufen wir unter **WERKZEUGE – BOARD – BOARDVERWALTER** auf.



Hier wählen wir ebenfalls den **TPY = AKTUALISIERBAR** aus und aktualisieren alle angezeigten BOARDS (kostet auch etwas Zeit).

Zusätzlich laden wir noch ESP32 und ARDUINO ESP32 BOARDS in der aktuellen Version (etwas abwarten).



8. Zugriffsrechte zuweisen

Über die Console werden noch zusätzliche Zugriffsrechte zugewiesen (ob erforderlich nicht ganz klar, kann jedoch nicht schaden)

```
sudo usermod -a -G dialout <BENUTZERNAME>
```

9. Python prüfen

Es kann sein, dass Paket bei python fehlen. Folgende sollten installiert sein

- tpython-tk
- python-is-python3
- python-serial

Zunächst prüfen wir die python Version. Dazu starten wir die Console und führen folgende Befehle aus:

```
python3 --version
```

- es sollte die Version 3.11.2 installiert sein

```
sudo apt update
```

```
sudo apt install python3-tk
```

```
sudo apt install python3-serial
```

```
sudo apt install python-is-python3
```

Wenn alles aktuell ist, erfolgt folgende Rückmeldung auf die Commands

```
pymml@pyMobaLEDLIB:~$ sudo apt install python-is-python3
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
python-is-python3 ist schon die neueste Version (3.11.2-1+deb12u1).
0 aktualisiert, 0 neu installiert, 0 zu entfernen und 218 nicht aktualisiert.
```

Dann Reboot des Raspberry

```
sudo shutdown -r now
```

10. Installation pyMLL

Die Software wird von Github geladen

<https://github.com/haroldlinke/pyMobaLedLib/releases>

Geladen wird die Version *pyMobaLedLib-5.3.6b.tar.gz*

Die Datei kopieren wir in das Verzeichnis */home/pymml/Arduino* und entpacken sie dort.

Start pyMLL

Zunächst sollte das neue Verzeichnis *pyMobaLedLib-5.3.6b.tar.gz* in *pyMobaLedLib* umbenannt werden.

Dann kann von der Console aus pyMLL gestartet werden.

```
python3 /home/<user>/Arduino/pyMobaLedLib/python/pyMobaLedLib.py
```

Pfade einstellen – stoppen

Neustart – Aktualisierung

Folgende Fehlermeldung:

```
16:53:55 Board-Definitionen werden heruntergeladen. 293 kB von 344 kB heruntergeladen.
16:53:55 Board-Definitionen werden heruntergeladen. 320 kB von 344 kB heruntergeladen.
16:53:55 Board-Definitionen werden heruntergeladen. 342 kB von 344 kB heruntergeladen.
16:53:55 *****
16:53:55 * Exception in start_ARDUINO_program_Popen 'utf-8' codec can't decode byte 0xe4 in
position 14: invalid continuation byte-
16:53:55 *****
```

pyMLL gestoppt und gesichert (trotz der Meldung)

Neustart

Verbindung konnte hergestellt werden.

Laden des LED Nano lief mit drei Zeilen Test Servo

Servo konnte danach angesteuert werden