MobaLedLib Studio – Fehlersuche

Seitenstand 1.14 (2026-11-24), gilt für Version 5.12

Auf dieser Seite stehen die häufigsten Probleme mit ihren Lösungen. Suche den Bereich, in dem es hakt, und die Beobachtung, die zu deinem Problem passt. Probleme, die nur eine bestimmte Installation betreffen (zum Beispiel „Python wurde nicht gefunden“), stehen am Ende der Seite für dein System: Windows, macOS, Linux, Raspberry Pi.

Zuerst prüfen

- **Welche Version läuft?** Die Nummer steht unter dem Titel der Oberfläche. Nach einem Update zeigt das Programm manchmal noch die alte Version (siehe unten).
- **Steht oben eine rote oder orange Meldung?** Sie sagt meist selbst, was fehlt. Zum Beispiel „Nicht alle Dateien passen zur Version …“ oder „Das Update … ist abgebrochen“. Siehe Wenn nichts mehr geht.
- **Das Protokoll:** Das Programm schreibt alle Meldungen in die Datei **webmobaledlib.log**. Du siehst die letzten Zeilen unter **Hilfe → Logfile anzeigen**. **Ganze Datei herunterladen** gibt dir die komplette Datei. Bei einer Fehlermeldung im Forum bitte diese Datei mitschicken (und *webmobaledlib.log.1*, falls es sie gibt). Wo die Datei liegt, steht auf der Seite Aktualisieren (Datenordner).

Start und Browser

Beobachtung	Ursache und Hilfe
„Port 8080 ist belegt“	Das Programm läuft schon. Dann öffnet das Symbol nur den Browser. Sonst läuft ein anderes Programm auf dem Port. Starte mit einem anderen Port, zum Beispiel `–port 8081`.
Der Browser öffnet sich nicht	Das Programm probiert mehrere Browser. Gib die Adresse von Hand ein: `http://localhost:8080`. Der Start dauert auf langsamen Rechnern bis zu einer Minute: **nicht noch einmal** auf das Symbol klicken.
Nach dem Aktualisieren zeigt das Programm noch die **alte Version**	Die **alte Instanz läuft noch**, und das Symbol öffnet nur den Browser darauf. Beende sie mit dem Knopf **BEENDEN** oben rechts und starte neu. Bleibt sie hängen: Linux und macOS `pkill -f webmobaledlib.py`. Windows: `netstat -ano` zeigt hinter :8080 die Nummer des Prozesses, `taskkill /PID <Nummer> /F` beendet ihn.
Meldung **„Connection lost“**, die Seite baut sich immer wieder neu auf	Meist ist der Rechner **langsam**: Der Browser braucht nach dem Seitenaufbau noch 10 bis 30 Sekunden, bis er sich mit dem Programm verbindet, und das Programm wartet nur 60 Sekunden. Schließe andere Programme, benutze einen leichteren Browser und wähle in den Programm Einstellungen das **kompakte Layout**. Auf dem **iPad oder iPhone** siehe den Abschnitt Raspberry Pi und Netz.
Nach jeder Änderung bleibt die Tabelle einen Moment **weiß**	Meist zeichnet der Browser ohne Grafikbeschleunigung. Schalte im Browser die **Hardwarebeschleunigung** ein, verkleinere Fenster und Browser-Zoom oder wähle das kompakte Layout. Hilft das nicht, aktualisiere auf die neueste Version.

Beobachtung	Ursache und Hilfe
Rote Meldung **„Nicht alle Dateien passen zur Version …“**	Es wurden nicht alle Dateien einer Version eingespielt. Jede Zeile nennt Datei, gefundene und verlangte Version. Mache **Hilfe → Update MobaLedLib Studio …** (das Update stellt fehlende und beschädigte Dateien wieder her) oder siehe Wenn nichts mehr geht.
Das Programm startet gar nicht	Starte es in einem Terminal oder einer Eingabeaufforderung und lies die Meldung: Windows `install\start_windows.bat`, macOS `install/start_mac.command`, Linux `install/start_linux.sh`. Das Protokoll liegt im Datenordner.
Die Konfiguration wird nicht gespeichert	Der Datenordner ist nicht beschreibbar. Dann nimmt das Programm einen Ersatzordner und warnt im Protokoll. Einen anderen Ort stellt `MLL_DATA_DIR` ein.

Die Lichtmaschine verbinden und das Programm übertragen

Beobachtung	Ursache und Hilfe
Mit Lichtmaschine verbinden klappt nicht, oben steht **Nicht verbunden**	Bei einer **neuen** Lichtmaschine ist das normal: Es läuft noch **kein MobaLedLib-Programm**. Stelle Board und Anschluss ein und übertrage zuerst das Programm, siehe Verbinden und Übertragen.
Bei **USB Port erkennen** erscheint kein Anschluss	Prüfe das **USB-Kabel** (Daten, nicht nur Laden) und stecke die Lichtmaschine neu an. Seit 3.10 erkennt die Liste **CH340, FTDI und Arduino** an der Hersteller-Kennung; erscheint er trotzdem nicht, setze **Unbekannte Ports anzeigen**. **Windows:** Bei Nachbauten mit dem Chip **CH340** fehlt oft der CH340-Treiber. **Linux und Raspberry Pi:** Du musst in der Gruppe `dialout` sein (`groups` zeigt es; sonst ab- und wieder anmelden).
Das **Hochladen bricht ab**	Prüfe das **Board** in **Hardware Optionen → Lichtmaschine**. Bei vielen Nano-Nachbauten hilft **Nano Normal (old Bootloader)**. Unter Linux und macOS erkennt das Programm den Typ nicht selbst: Wähle ihn immer richtig.
Die Statuszeile zeigt „**Headerfileerstellung abgebrochen – Die Headerdatei wurde nicht geschrieben**“ gleich nach **Setup übertragen** (Version vor 3.22)	Im Fenster **ARDUINO-IDE Einstellungen** (Optionen, Reiter Entwicklungsumgebung) ist der oberste Haken **Test ARDUINO Programm Creation** gesetzt. Das ist ein Testmodus für Entwickler: Nach jedem Schritt der Erzeugung kommt eine Frage. Bis 3.21 erschien sie im Browser nicht und galt als „Nein“. **Nimm den Haken weg**, drücke **Speichern** (oder aktualisiere: Seit 3.22 erscheint die Frage als Fenster, seit 3.23 wirkt der Haken sofort nach dem Speichern; in 3.22 erst nach einem Neustart).
Meldung **„Hochladen: Prüfung fehlgeschlagen“** (im Protokoll `stk500_paged_load … failed to read all of flash memory`)	Das Programm wurde **geschrieben**, nur die anschließende **Prüfung** (Rücklesen des Speichers) ist gescheitert. Prüfe zuerst, ob die **Heartbeat LED** blinkt: Dann läuft die Lichtmaschine mit dem neuen Programm. Hilfen: Probiere **beide** Einstellungen **Nano Normal (old Bootloader)** und **Nano (neue Version)** (Hardware Optionen → Lichtmaschine) und übertrage noch einmal. Probiere ein anderes **USB-Kabel** und einen anderen **USB-Anschluss**. Versorge die **LEDs nicht über den USB-Anschluss** des Computers (auf dem Raspberry Pi besonders wichtig).

Beobachtung	**Ursache und Hilfe**
Meldung **„Hochladen: Keine Antwort“** (im Protokoll `stk500_getsync … not in sync`, `programmer is not responding`; seit 3.10)	Der Nano antwortet dem Bootloader-Programm nicht. Meist passt die Einstellung **Nano Normal (old Bootloader)** / **Nano (neue Version)** nicht zum Bootloader des Nano (alt: 57600 Baud, neu: 115200 Baud): **Probiere beide.** Sonst: USB-Kabel (es muss Daten übertragen), anderer Anschluss, und der Anschluss darf von keinem anderen Programm benutzt werden (zum Beispiel dem Seriellen Monitor). Ohne Antwort versucht es avrdude zehnmal, das dauert fast eine Minute.
Beim Knopf **Firmware herunterladen** kommt „**Die Liste der verfügbaren Firmware konnte nicht ermittelt werden**“ oder „**Das Herunterladen der Firmware ist fehlgeschlagen**“ (seit 3.15)	Das Programm erreicht GitHub nicht. Prüfe die **Internetverbindung** (Firewall, Proxy, Uhrzeit des Rechners) und versuche es später noch einmal; der Fehlertext steht in der Meldung. Eine **kaputte ZIP-Datei** (Meldung „konnte nicht entpackt werden“) bleibt im Ordner *MobaLedLib/firmware* liegen: Du kannst sie von Hand entpacken. Die fertigen Dateien (*.uf2*) stehen nach dem Entpacken im selben Ordner.
Windows: Beim **USB Port erkennen** erscheint „**Der COM Port … wird bereits von einem anderen Programm benutzt**“, obwohl kein anderes Programm läuft; **Nein** und danach **Setup übertragen** geht	Das Programm hatte den Anschluss kurz zuvor selbst benutzt, und Windows (CH340) gibt ihn erst nach einem Moment frei (Protokoll: `PermissionError(13, 'Zugriff verweigert')`). **Seit 3.10** schließt das Programm den Anschluss nach jeder Prüfung und versucht das Öffnen bis zu dreimal. Kommt die Meldung weiter, ist er wirklich belegt: Serieller Monitor, Farb Test oder ARDUINO IDE schließen.
Bei der **Bootloader-Erkennung** (seit 3.13 voreingestellt) kommt nach ein paar Sekunden „**Das Gerät am COM Port … wurde nicht als LED Arduino erkannt**“	Der Nano antwortet dem Bootloader-Programm weder mit 57600 noch mit 115200 Baud. Klicke **Nein**: Dann bleibt deine Wahl **Nano Normal (old Bootloader)** / **Nano (neue Version)**, und das Übertragen klappt. Die Prüfung endet nach etwa 6 Sekunden. Wer die Erkennung nicht möchte, nimmt in den **Programm Einstellungen** (Spezialeinstellungen) den Haken **Bootloader des Nano beim Prüfen erkennen** weg.
Linux/Raspberry Pi: Der Anschluss erscheint erst mit **Unbekannte Ports anzeigen**, danach „**Es ist kein LED Arduino an COM Port /dev/ttyUSB0 angeschlossen**“	Der Nano meldet sich als „USB Serial“, das kannte die Liste nicht. **Seit 3.10** erkennt sie CH340, FTDI und Arduino an der **Hersteller-Kennung**. Zum Nachsehen: `python3 -m serial.tools.list_ports -v` (`hwid:` mit `VID:PID=1A86:…` ist ein CH340).
„**Der … am COM Port … lässt sich nicht ansprechen (Fehlercode -2 oder -3)**“	Der Anschluss ließ sich nicht zuweisen oder der Arduino nicht zurücksetzen. Das Programm hat es dreimal versucht. USB-Kabel abziehen, wieder anstecken und noch einmal versuchen.
Im Protokoll und im Fenster stehen **tausende Zeilen** `javax.jmdns`, `dns[response` und Hexdumps (Raspberry Pi)	Meldungen der **ARDUINO IDE** (Suche nach Boards im Netz); der Rechnername des Pi ist durch Namenskonflikte mit der FRITZ!Box verlängert (zum Beispiel `RaspiMLL-fritz-box-fritz-box-…`). **Seit 3.10** blendet das Programm sie aus. Setze zusätzlich einen **kurzen, festen Rechnernamen** (`sudo raspi-config` → System Options → Hostname).
Das Übertragen dauert **sehr lange**	Beim **ersten Mal** ist das normal (bis zu 10 Minuten): PlatformIO lädt Werkzeuge aus dem Internet nach. Prüfe sonst die Internetverbindung.
Meldung **„PlatformIO nicht gefunden“**	Die Installation wurde nicht ganz beendet. Starte das Installationsskript noch einmal (Installation).
Beim **Raspberry Pico** meldet das Hochladen *picotool was unable to connect* (Linux)	Der USB-Zugriff für den Pico ist nicht eingerichtet. Klicke in **Hardware Optionen → ENTWICKLUNGSUMGEBUNG** auf **USB-Zugriff für den Pico einrichten** (einmalig, das Programm fragt nach dem Passwort).

Beobachtung	Ursache und Hilfe
ESP32: Nach dem Hochladen blinkt die Platine dauernd, der serielle Monitor zeigt immer wieder *assert failed: lfs_fs_grow_*	Im Flash liegt ein altes Dateisystem eines früheren Projekts. Ab **4.97** löscht das Programm es nach dem Hochladen per USB selbst. Ein Gerät, das schon abstürzt, lädst du einmal **per USB** neu hoch (nicht per WLAN).
ESP32: Das Hochladen bleibt bei *Connecting.....____* stehen und endet mit *Failed to connect to ESP32*	Der Chip kommt nicht in den Bootloader: **Halte auf dem ESP32-Board die Taste BOOT gedrückt**, sobald im Fenster „Connecting…“ erscheint, bis „Writing at …“ kommt. Viele Nachbauten haben keinen zuverlässigen Auto-Reset. Hilft es nicht: ein Datenkabel direkt am Rechner ohne Hub und einen anderen USB-Port probieren.
Meldung **„Die ARDUINO IDE wurde nicht gefunden“**	Mit **PlatformIO** (Standard) sollte sie nicht kommen. Klicke in **Hardware Optionen → ENTWICKLUNGSUMGEBUNG** auf **PlatformIO prüfen**. Benutzt du die ARDUINO IDE, siehe ARDUINO Einstellungen.
Beim **Übertragen** bleibt das Fenster stehen, nichts passiert mehr	Hängt ein Befehl länger als **25 Sekunden**, schreibt das Programm den Zustand in die Datei `hang_dump.txt` im Datenordner (seit 4.28). Schick die Datei mit dem Protokoll (**Hilfe → Logfile anzeigen**) ins Forum. **Vorgang abbrechen** beendet den Vorgang, der Anschluss ist danach sofort wieder frei.
Meldung **„Es läuft noch ein Vorgang“** bleibt stehen und blockiert alle Eingaben	Bis 4.34 möglich (nach einem Fehler beim Verbinden). Aktualisiere auf die neueste Version.
Dialoge und Menüs erscheinen nicht über der Seite, sondern **unter der Tabelle** (bisher nur bei einem Mac gemeldet)	Seit 4.77 liegen sie durch eine CSS-Absicherung fest über der Seite. Aktualisiere, starte das Programm neu und lade die Seite mit **Strg+F5** (am Mac **Cmd+Umschalt+R**) neu. Hilft es nicht, melde im Forum oder als Issue: Browser mit Version, Vollbild oder Zoom, Meldungen der Browser-Konsole.

Das Programm erzeugen: Makros und Tabelle

Beobachtung	Ursache und Hilfe
Ein **Makro** wird beim Doppelklick nicht vorausgewählt, oder in der Liste stehen **Makros doppelt**	Im Ordner **libraries** darf nur stehen, was dort hingehört (**MobaLedLib** und die **Erweiterungen**). Jeder Ordner mit einer Datei `MobaLedLib.properties` wird als Erweiterung eingelesen. Entferne fremde oder alte Ordner (zum Beispiel `ServoAnim`, `LEDColorAnim`, `MultiLEDColorAnim`, `Pattern`, `ServoAnim2B`).
Meldung beim Start: **„Das Makro … ist schon eingebaut … und wird ignoriert“**	Eine Erweiterung bringt ein Makro mit einem Namen, den es schon gibt. Benenne den Ordner der Erweiterung um oder entferne ihn.
Beim Kompilieren fehlt **MobaLedLib.h** oder eine Bibliothek	Die Bibliotheken müssen installiert sein. Das Programm bietet das beim Start an.
Ein **Text** ist in meiner Sprache noch **deutsch**	Mache ein Bildschirmfoto und schreibe Stelle und Sprache ins Forum. Die Übersetzungen werden laufend ergänzt.

Raspberry Pi und Netz

Beobachtung	Ursache und Hilfe
Das Symbol meldet „Der Dienst konnte nicht gestartet werden“	Führe `sudo bash install_pi.sh –no-check` noch einmal aus oder starte den Dienst mit `sudo systemctl start mobaledlib-web`.
Die Seite lädt nicht	Prüfe mit `systemctl status mobaledlib-web` den Dienst. Hänge den Port an die Adresse (`http://<Adresse>:8080`). Warte beim ersten Start eine Minute.
`raspberrypi.local` wird nicht gefunden	Dieser Name funktioniert nicht überall (Gastnetz, WLAN mit Geräteisolation, Firmennetz, Android). Nimm die **IP-Adresse** (steht im Router).
Das Anmeldefenster nimmt das Passwort nicht an	Benutzer ist `mobaledlib`. Das Passwort ist das Standardpasswort `mobaledlib` oder das im Programm gesetzte. Vergessen? Lösche die Datei `web_password.json` im Datenordner.
Der Dienst startet nicht	`journalctl -u mobaledlib-web -e` zeigt die Meldung. `pi_check.py` zeigt fehlende Pakete.
Auf dem **iPad oder iPhone** bricht die Verbindung ab, zum Beispiel beim Wechsel der **Tabelle** („Connection lost“)	Alle Browser auf dem iPad und iPhone senden die Anmeldung nicht beim Aufbau der Dauerverbindung mit. Aktualisiere das Programm auf Version **3.05 oder neuer**: Seitdem merkt es sich die Anmeldung mit einem **Cookie**. **Cookies müssen erlaubt sein.**
Firefox oder Chrome auf dem iPad zeigt **gar keine Verbindung** und keine Anmeldung	iOS verlangt die Berechtigung **Lokales Netzwerk**. Schalte sie ein: **Einstellungen → Datenschutz & Sicherheit → Lokales Netzwerk** (oder bei den Einstellungen des Browsers). Oder benutze **Safari**.
Ein anderer Rechner im Netz erreicht die Oberfläche nicht	Meist die **Firewall** (Windows: Netzwerkprofil „Öffentlich“). Siehe Zugriff aus dem Netz.
Der Pi **hängt sich auf**, oder das Programm verschwindet ohne Meldung	Meist ist die **Stromversorgung** zu schwach (Netzteil, USB-Festplatte, Arduino mit vielen LEDs am USB-Anschluss) oder der Pi zu heiß. Benutze das offizielle Netzteil, versorge LEDs und Festplatten getrennt und lass das Gehäuse frei stehen. Das Protokoll enthält ab Version 2.22 Zeilen zu Temperatur und Spannung.

Wenn nichts mehr geht

Ein Update kann abbrechen (Stromausfall, Programm beendet, ein Virenprogramm sperrt eine Datei) oder Dateien können fehlen. Probiere der Reihe nach:

1. **Orange Hinweis „Das Update … ist abgebrochen“:** Das Programm ist **automatisch zur vorherigen Version zurückgegangen**. Deine Tabellen und Einstellungen sind nicht betroffen. Schließe den Hinweis mit **OK** und versuche das Update später noch einmal. Scheitert es wieder an derselben Datei, beende das Programm, halte das Sicherungs- oder Virenprogramm kurz an und wiederhole das Update.
2. **Rote Meldung „… Dateien fehlen oder haben eine andere Größe als in der Prüfliste“:** Mache **Hilfe → Update MobaLedLib Studio …**. Auch bei gleicher Version **stellt das Update fehlende und beschädigte Dateien wieder her**. Starte danach neu.
3. **Das Programm startet gar nicht, oder rot „Das automatische Zurückgehen war nicht vollständig“: Reparaturskript.** Es braucht das Programm nicht und läuft auch, wenn es beschädigt ist. Beende das Programm vorher.

System	Aufruf
Windows	im Ordner **python\install** die Datei **repair_windows.bat** doppelklicken
macOS	im Ordner **python/install** die Datei **repair_mac.command** doppelklicken
Linux	`bash install/repair_linux.sh` im Ordner **python**
Raspberry Pi	`bash install/repair_linux.sh` im Ordner **python**, danach `sudo systemctl restart mobaledlib-web`

Das Skript macht ein abgebrochenes Update rückgängig, vergleicht die Dateien mit der Prüfliste, sucht ein vollständiges Release (zuerst die zwischengespeicherten, sonst von GitHub) und schreibt nur die fehlenden oder abweichenden Dateien neu. Die ersetzten Dateien sichert es vorher. Es fragt vor dem Schreiben (**j**). **Ohne Internet:** Lade `webmobaledlib.zip` von GitHub und rufe das Skript mit `–zip <Pfad zu webmobaledlib.zip>` auf. `–check-only` prüft nur.

1. **Das Update bricht mit „keine Prüfliste“, „unvollständig“ oder „würde … Dateien löschen“ ab:** Das Paket ist nicht das vollständige Release. Es wurde **nichts verändert**. Melde es im Forum.
2. **Auch das Reparaturskript fehlt oder geht nicht (von Hand):**
    1. Beende das Programm.
    2. Lade das aktuelle `webmobaledlib.zip` von GitHub.
    3. **Benenne den Ordner python um** (zum Beispiel in *python_defekt*), entpacke das ZIP (es enthält den Ordner **python**) und starte wie gewohnt. Auf dem Raspberry Pi startest du den Dienst neu: `sudo systemctl restart mobaledlib-web`.
    4. Deine Tabellen, Einstellungen und das Passwort liegen im **Datenordner** und bleiben dabei erhalten. Die Python-Umgebung (**venv_web** bzw. **venv**) gehört nicht zum ZIP: Liegt sie im alten Ordner, verschiebe sie in den neuen.
3. **Dateien zurückholen:** Ersetzte und gelöschte Dateien liegen im Datenordner unter `update_backup/<Datum-Uhrzeit>/` (letzte fünf), die von der Reparatur ersetzten unter `repair_backup/<Datum-Uhrzeit>/`.

Alles neu installieren (zum Beispiel auf dem Raspberry Pi): Ein Skript namens *uninstall.bat* gibt es nicht. Auf dem Pi entfernt `sudo bash ~/mobaledlib/python/pi/uninstall_pi.sh` den Dienst, das Desktop-Symbol und die sudo-Regel, **nicht** den Ordner *mobaledlib*. Lösche ihn danach selbst, entpacke das ZIP neu (`unzip webmobaledlib.zip` in einem leeren Ordner *mobaledlib*) und führe im Ordner **mobaledlib/python/pi** `sudo bash install_pi.sh` aus. Deine Daten bleiben erhalten, denn der Datenordner liegt außerhalb.

Wenn nichts hilft

Schreibe ins Forum der MobaLedLib. Nenne dein **System**, die **Programmversion** (unter dem Titel der Oberfläche), **was du getan hast** und die **Fehlermeldung**. Hänge die Datei **webmobaledlib.log** an (siehe oben).

From:
https://wiki.mobaledlib.de/ - **MobaLedLib Wiki**

Permanent link:
https://wiki.mobaledlib.de/anleitungen/mobaledlibstudio/fehlersuche

Last update: **2026/11/24 17:08**