MobaLedLib Studio – Erweiterungen (MLX)

Seitenstand 1.04 (2026-11-24), gilt für Version 5.12

Erweiterungen (englisch *MobaLedLib Extensions*, kurz **MLX**) bringen **zusätzliche Makros** in die Makroauswahl, zum Beispiel einen **Zeitplan**, ein **Kaminfeuer** oder den Empfang von **Infrarot-Fernbedienungen**. Sie gehören nicht zum Kern der MobaLedLib, sondern sind kleine Programmpakete, die jemand zusätzlich geschrieben hat. Du schaltest sie im Fenster **Erweiterungen** (Menü **Einstellungen, Erweiterungen …**) im Reiter **Erweiterungen** ein und aus.

Du brauchst Erweiterungen nicht für ein erstes Programm. Sie sind **optional**, manche brauchen **Zusatzhardware**. Deshalb zeigt die Makroauswahl nur die Erweiterungen, die du eingeschaltet hast.

Voraussetzung: Erweiterungen gibt es nur mit **PlatformIO**. Wenn du die Arduino IDE benutzt, siehe Zusätzlich: Arduino IDE; stelle für Erweiterungen auf PlatformIO um (Reiter **Entwicklungsumgebung**).

Das Fenster öffnen

1. Öffne im Programm Generator das Menü **Einstellungen** und wähle **Erweiterungen …**.
2. Setze oder entferne den **Haken** vor einer Erweiterung. Die Änderung gilt für die Makroauswahl, sobald du das Fenster mit **Schließen** schließt. Du musst nichts speichern und nichts neu starten.

Steht in der Erweiterung eine Datei `libraries.json` mit `name` und `description`, zeigt das Programm diesen **Namen** und die **Beschreibung** an. Sonst nimmt es die Angaben aus der Liste.

Die Haken gelten **für alle Dateien** und bleiben beim Programm-Update erhalten.

Die bekannten Erweiterungen

Das Fenster zeigt drei Gruppen. Neben jeder Erweiterung steht, ob sie **im Programmpaket**, schon **geladen** oder **noch nicht geladen** ist.

Gruppe	Erweiterung	Wofür
Standard (vorab eingeschaltet, im Programmpaket)	**P_ScheduleMLX**	Schaltet Beleuchtungen nach der **Uhrzeit** (ein Zeitbereich).
	P_ScheduleDuoMLX	Wie oben, mit **zwei getrennten Zeitbereichen** A und B.
Optional (vorab ausgeschaltet)	**AddressLearnMLX**	**Adressen einlernen**, zum Beispiel für den Verkehrssicherungsanhänger.
	FireplaceMLX	**Kaminfeuer** mit einer RGB-LED.
	InfraredMLX	Steuern mit einer **Infrarot-Fernbedienung** (RC5 und NEC). Braucht einen **IR-Empfänger**.
Eigene Erweiterungen	(von dir hinzugefügt)	siehe unten

Eine optionale Erweiterung einschalten

1. Setze den Haken vor der Erweiterung.
2. Beim **ersten Mal** lädt das Programm sie aus dem **Internet** (von GitHub) und legt sie im **Datenordner** des Programms ab (Unterordner `extensions`). Der Status wechselt von *noch nicht geladen* auf *geladen*.
3. Danach arbeitet sie **ohne Internet**. Aus- und wieder Einschalten lädt nichts neu.
4. Die Makros der Erweiterung erscheinen in der Makroauswahl in der Gruppe **Erweiterungen**.

Klappt der Download nicht (kein Internet, GitHub nicht erreichbar), meldet das Programm es, und der Haken springt zurück.

Eigene Erweiterungen hinzufügen

Wer eine neue Erweiterung entwickelt oder eine fertige benutzen möchte, die noch nicht in der Liste steht, trägt sie unten im Fenster ein:

1. Schreibe in das Feld **Eigene Erweiterung hinzufügen** eine **Zeile** in einer dieser Formen:

Eingabe	**Bedeutung**
`MeineNeueExtensionMLX`	Paket aus der **PlatformIO-Registry**, aktuelle Version
`MeineNeueExtensionMLX@=1.1.0`	genau die **Version 1.1.0**
`MeineNeueExtensionMLX=https://github.com/Besitzer/MeineNeueExtensionMLX`	Code aus dem **GitHub-Repository**
`https://github.com/Besitzer/MeineNeueExtensionMLX`	wie oben, der Name ist der Name des Repositorys
`Name=https://github.com/Besitzer/Repo#v1.2`	wie oben, hinter # steht ein **Tag** oder **Zweig**

1. Klicke auf **Hinzufügen**. Beim **ersten Mal** erscheint der Hinweis zu fremdem Code (siehe unten); bestätige ihn mit **Hinzufügen**.
2. Das Programm lädt die Erweiterung, liest ihre Beschreibung (die Datei `MobaLedLib.properties`) und zeigt sie unter **Eigene Erweiterungen** an, schon **eingeschaltet**.

Mit **Entfernen** nimmst du sie wieder aus der Liste und löschst die geladene Kopie.

Gut zu wissen:

- Statt einer GitHub-Adresse geht auch die Adresse einer **ZIP-Datei** (`https://…/datei.zip`). Andere Adressen werden abgelehnt.
- Der Name darf nur Buchstaben, Ziffern, `_`, `-` und `.` enthalten. Er darf nicht schon vergeben sein.
- Die Erweiterung muss die Regeln für *MobaLedLib Extensions* einhalten, also eine `MobaLedLib.properties` mitbringen. Sonst meldet das Programm einen Fehler.
- Das Laden aus der **PlatformIO-Registry** setzt voraus, dass PlatformIO gefunden wurde (Reiter **Entwicklungsumgebung**, Knopf **PlatformIO prüfen**).

Wichtig: fremder Code

Eine Erweiterung ist **Programmcode**. Er wird beim Übersetzen in **dein Programm** eingebunden und läuft auf der Lichtmaschine. Füge deshalb nur Erweiterungen aus **Quellen** hinzu, **denen du vertraust**, zum Beispiel die bekannten aus der Liste oder die eines Entwicklers, den du kennst. Das Programm erinnert dich beim ersten Hinzufügen einer Adresse daran.

Wenn eine Datei eine ausgeschaltete Erweiterung benutzt

Öffnest du eine Datei, in der Zeilen einer **ausgeschalteten** Erweiterung stehen (zum Beispiel von einem Freund), fragt das Programm: *Die Datei benutzt die Erweiterung … Jetzt einschalten?*

- **Einschalten:** Das Programm lädt sie bei Bedarf und schaltet sie ein. Die Zeilen sind dann wieder gültig.
- **Nein:** Die Erweiterung bleibt aus. Dann **fehlen die zugehörigen Zeilen** in der erzeugten Datei. Gefragt wird pro Erweiterung **einmal je Sitzung**.

Wenn es nicht klappt

Was passiert	**Was hilft**
Die Gruppe **Erweiterungen** fehlt in der Makroauswahl	Im Fenster **Erweiterungen** (Menü **Einstellungen**) ist keine Erweiterung eingeschaltet, oder du benutzt die Arduino IDE statt PlatformIO.
Die Erweiterung … konnte nicht geladen werden	Prüfe die **Internetverbindung**. Bei einer eigenen Erweiterung prüfe auch die **Schreibweise** der Adresse und ob das Repository **öffentlich** ist.
MobaLedLib.properties fehlt oder *nicht lesbar*	Das ist keine gültige MobaLedLib-Erweiterung, oder die Beschreibungsdatei hat einen schweren Formfehler. Frage den Entwickler der Erweiterung. Kleine Formfehler (fehlende Anführungszeichen oder Kommas) toleriert das Programm.
Der Übersetzungsvorgang meldet einen Fehler bei einer Erweiterung	Schalte die Erweiterung aus und übersetze noch einmal. Wenn es dann geht, melde den Fehler dem Entwickler der Erweiterung, siehe Fehler melden.
Ich habe eine Erweiterung von Hand in den Ordner `libraries` gelegt	Sie wird wie bisher **immer** benutzt, der Reiter schaltet sie nicht aus. Ist sie auch in der Liste, hat die Liste Vorrang.

Weitere Hilfe: Fehlersuche.

Für Entwickler von Erweiterungen

- Die Beschreibung steht in der Datei `MobaLedLib.properties` im Hauptordner der Erweiterung (`id`, `types`, `macros`, `parameters`, `includes`). Die **Id** in der Datei ist der Name, unter dem das Programm die Erweiterung in der Datei `platformio.ini` einträgt.
- Zum Testen trägst du die GitHub-Adresse deiner Erweiterung ein (mit `#Zweig` für einen Entwicklungszweig). Nach einer Änderung **entfernst** du den Eintrag und fügst ihn **neu hinzu**, damit das Programm den neuen Stand lädt.

- Ist deine Erweiterung ausgereift und weit verbreitet, kann sie in die Liste der bekannten aufgenommen werden (Datei `extensions_known.json`); dazu am besten im Projekt auf GitHub melden.

Siehe auch: Die Werkzeugleiste und das Fenster Optionen.

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

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

Last update: **2026/11/24 17:20**