Junsi iCharger 4010Duo - Linux Setup Guide (Debian/LMDE)

Diese Dokumentation beschreibt die Einrichtung und Fehlerbehebung für den Zugriff auf Junsi iCharger Hardware unter Linux (getestet mit LMDE 7). Da der iCharger ein Composite-Gerät ist, wird der Zugriff über **PyUSB** anstelle von hidapi durchgeführt.

1. System-Voraussetzungen

Zuerst müssen die notwendigen System-Bibliotheken für die USB-Kommunikation und die Python-Umgebung installiert werden:

```
sudo apt update
sudo apt install libusb-1.0-0 libusb-1.0-0-dev python3-venv
```

Innerhalb der virtuellen Umgebung (venv) des Projekts muss PyUSB installiert sein:

```
pip install pyusb
```

2. USB-Architektur des iChargers

Der iCharger meldet sich am USB-Bus als **Composite Device** mit zwei Schnittstellen:

- [cite_start]**Interface 0 (HID):** Zuständig für die Datenübertragung (Modbus via HID). [cite: 178, 179]
- [cite_start]**Interface 1 (Mass Storage):** Stellt die SD-Karte als Laufwerk bereit. [cite: 140]

3. udev-Regeln konfigurieren

Um den Zugriff für Nicht-Root-Benutzer zu erlauben, muss eine udev-Regel angelegt werden. Diese sorgt dafür, dass die Gerätedatei unter `/dev/bus/usb/` die richtigen Rechte erhält.

Erstellen Sie die Datei `/etc/udev/rules.d/99-junsi.rules`:

```
# Junsi iCharger 4010Duo - VID:0483 PID:5751
# 1. Erlaubt Schreib-/Leserechte für die Gruppe plugdev
# 2. Weist uaccess zu, um dem aktuellen User direkten Zugriff zu gewähren
SUBSYSTEM=="usb", ATTR{idVendor}=="0483", ATTR{idProduct}=="5751",
MODE="0666", GROUP="plugdev", TAG+="uaccess"
```

Aktivieren Sie die Regeln:

```
sudo udevadm control --reload-rules
sudo udevadm trigger
```

4. Diagnose-Verfahren

Sollte das Gerät nicht erkannt werden, prüfen Sie die folgenden Punkte:

A. Kernel-Treiber Check

Prüfen Sie, ob ein Treiber das Interface blockiert:

```
lsusb -t
```

Achten Sie auf die Zeile für den iCharger. Wenn dort `Driver=usbhid` steht, muss der Treiber im Python-Code via `detach_kernel_driver()` gelöst werden.

B. Rechte-Check

Suchen Sie Bus und Device Nummer:

```
lsusb | grep 0483
```

Prüfen Sie dann die Rechte (Beispiel für Bus 002, Device 016):

```
ls -l /dev/bus/usb/002/016
```

Soll-Ergebnis: `crw-rw-rw-` oder `crw-rw—-+` (ACL).

C. Live-Monitor

Beobachten Sie das System beim Einstecken des Geräts:

```
udevadm monitor --environment --udev
```

Achten Sie darauf, ob die Zeile TAGS=`:uaccess:seat:` erscheint.

5. Integration im Framework

[cite_start]Der Transport nutzt das Modbus-Protokoll über HID-Reports. [cite: 418, 420]

- [cite_start]**Anfrage:** Paket wird in einen 64-Byte Report verpackt. [cite: 418, 419]
- [cite_start]**Antwort:** Der Lader antwortet mit Function Codes wie 0x03 oder 0x04. [cite: 184, 203]

6. Troubleshooting

Fehler	Ursache	Lösung
—	—	—
`Access Denied`	udev Regel nicht aktiv	udevadm trigger ausführen & neu einstecken.
`Resource Busy`	Interface 1 (SD-Karte) wird gerade gemountet	SD-Karte entfernen oder kurz warten.
`open failed`	Kernel-Treiber (usbhid) blockiert	Sicherstellen, dass der Transport `claim_interface` nutzt.

</code>

From:
https://drklipper.de/ - **Dr. Klipper Wiki**

Permanent link:
https://drklipper.de/doku.php?id=ionpy:hardware:junsi_icharger

Last update: **2026/01/26 21:02**