

Anleitung Installation AliceLite Software auf Windows 10 oder Ubuntu Linux (16, 18 oder 20) für das M1K-Modul (ADALM1000) von Analog Devices

Die Benutzeroberfläche für das M1K ist in Python (Version 3.7) geschrieben und besteht aus acht Python-Modulen. Das Hauptmodul hat den Dateinamen `aliceLite.py`.

Damit dieses Skript auf dem PC gestartet werden kann, müssen vorher die Windowsbibliothek `libsmu` sowie das Python-Modul (= Bibliothek) `pysmu` installiert werden. `pysmu` ist eine Python-Schnittstelle zu eigentlichen Bibliothek `libsmu`.

Außerdem muss (für Windows 10) der Gerätetreiber für das M1K installiert werden.

Weiter muss ein Python-Interpreter auf dem PC vorhanden sein.

Windows 10

Installation des Python-Interpreters, der Bibliotheken und des Gerätetreibers unter Windows 10

Unter Windows ist meistens kein Python-Interpreter vorhanden. Dieser muss also zusätzlich installiert werden, wobei bisher nur die Python-Version 3.7 erfolgreich getestet wurde.

Es gibt zwei verschiedene Möglichkeiten a) und b) der Installation von Python 3.7:

Bevorzugt wird die Möglichkeit b).

a) Installation von Python 3.7, siehe www.python.org/downloads/release/python-379/. Hier am besten die Installationsdatei `windows x86-64 executable installer` verwenden.

Da AliceLite das Python-Modul NumPy benötigt, muss dies mit `pip install numpy` anschließend nachinstalliert werden.

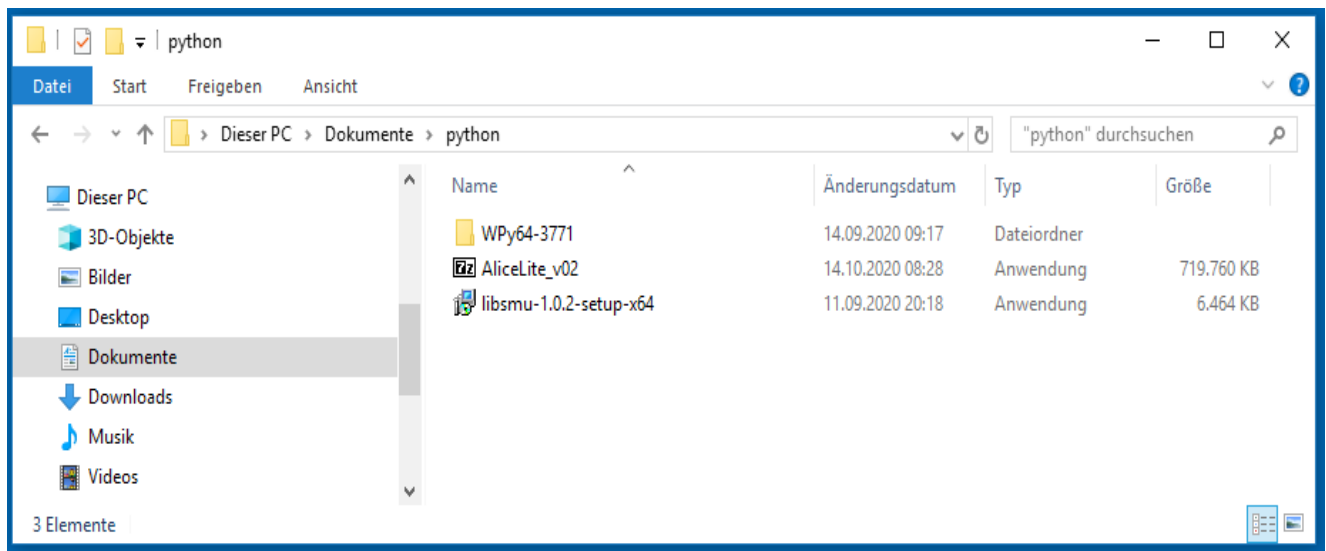
Die weiteren Installationsschritte sind nahezu identisch zu b)

b) Die Module AliceLite, Numpy und der Python-3.7-Interpreter können auch als Portable App auf den Rechner „installiert“ werden. Dafür wird die Python-Distribution `winPython 3.771` verwendet. (Siehe sourceforge.net/projects/winpython/, die unter dieser Webadresse zum Download bereitgestellte Installationsdatei jedoch nicht verwenden!).

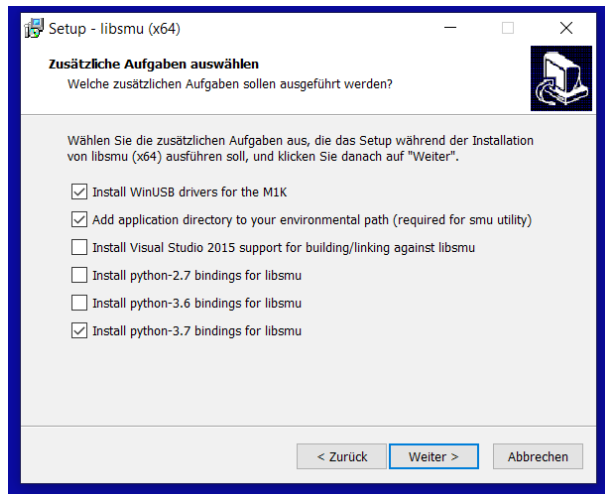
WinPython 3.771 ist inklusive der nötigen AliceLite-Python-Quellcodes in dem Archiv `aliceLiteWinPyth_v0x.exe` (selbstentpackend) bzw. `aliceLiteWinPyth_v0x.zip` enthalten.

Dieses Archiv wird an einem Ort entpackt, auf den der Benutzer uneingeschränkte Schreib- und Leserechte hat wie z.B. unter `\Dokumente\python\` siehe Beispiel-Screenshot.

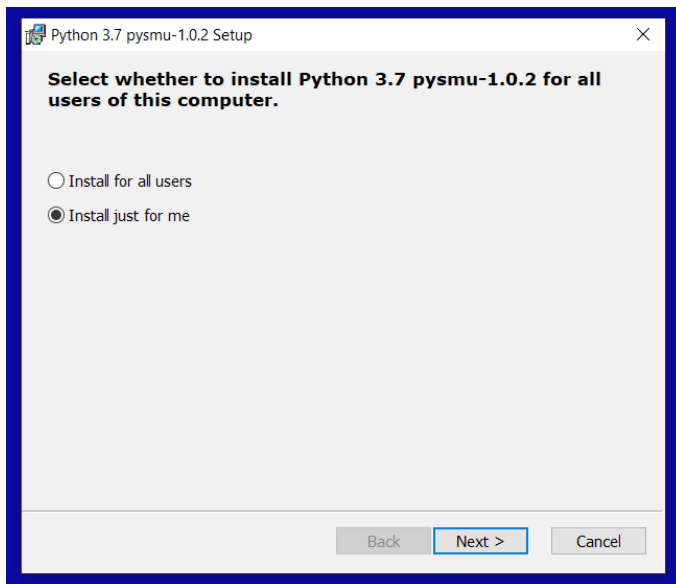
Achtung: Der Pfad zu dem WinPython-Verzeichnis `WPY64-3771` darf keine Leerzeichen oder Umlaute enthalten. Außerdem sollte der Pfad nicht zu lang sein. Wenn Sie auf Nummer Sicher gehen wollen: Unter C : ein Verzeichnis `tmp` anlegen und darin das Archiv entpacken.



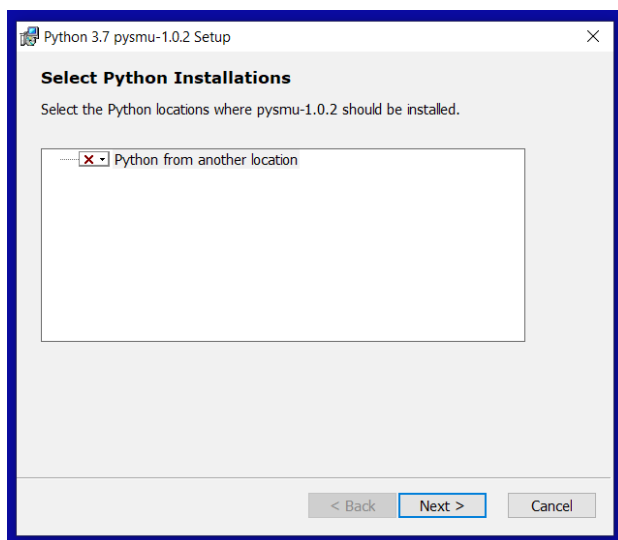
Über das Installationsprogramm `libsmu-1.0.2-setup-x64.exe` werden nun die nötigen Bibliotheken sowie der Gerätetreiber installiert:



Hierbei wie oben im Bild mindestens die Treiber und die `python-3.7` bindings auswählen.

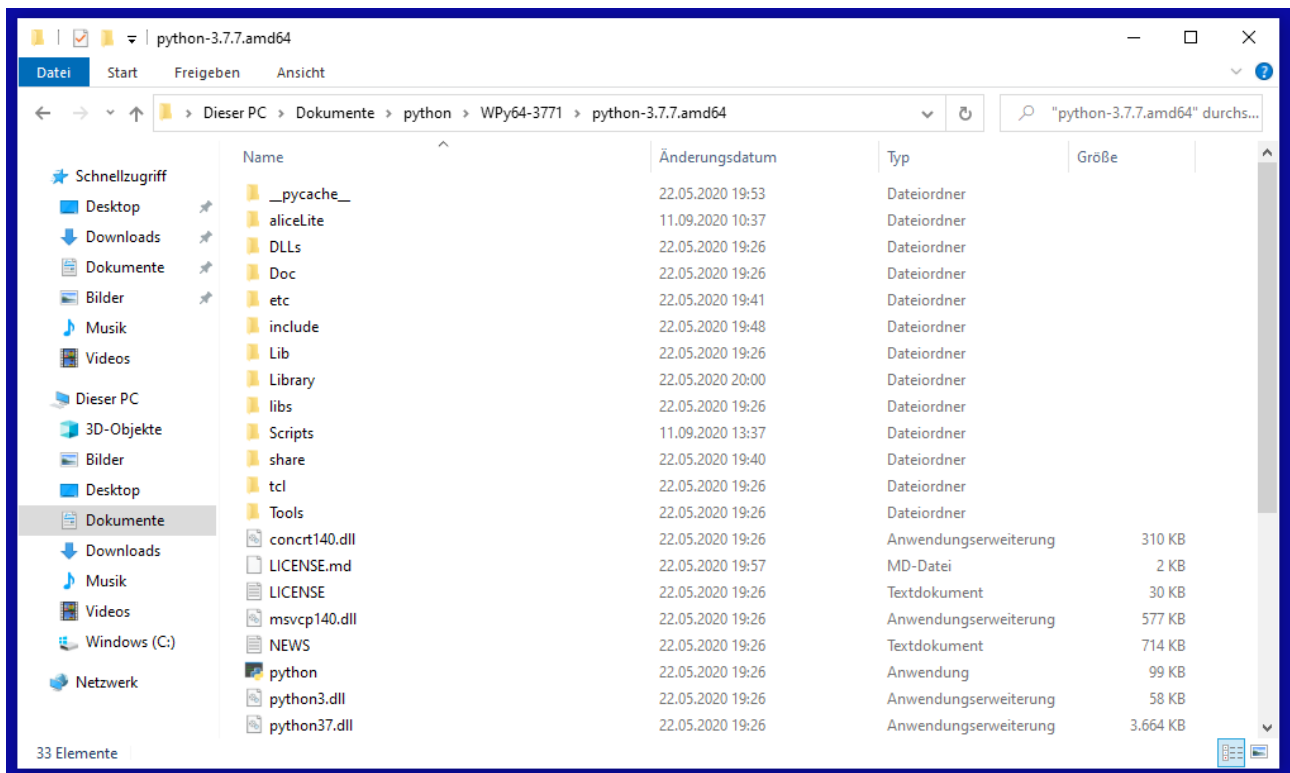


Nun ist wichtig, dass das Installationsprogramm den Python-Interpreter findet. Dazu muss dessen Verzeichnis angegeben werden, da bei WinPython ja kein systemweiter in der Windows Registry notierter „Standard-Python-Interpreter“ existiert, sondern WinPython seinen eigenen isolierten Python-Interpreter mitbringt.



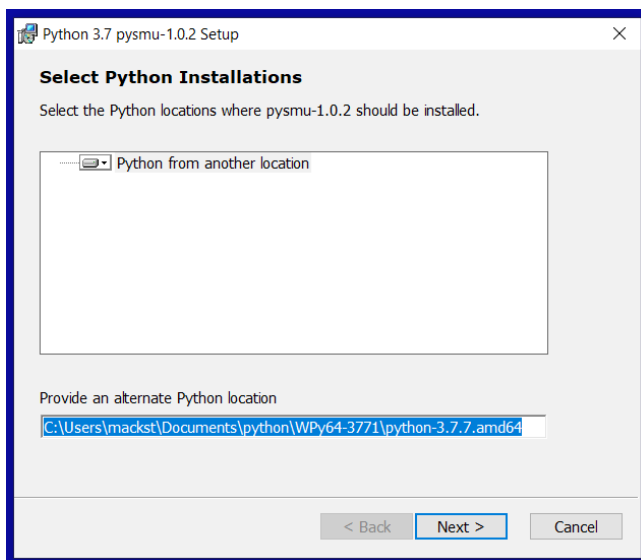
Dazu im nächsten Fenster Dateipfad des Python-Interpreters angeben.

Den Pfad erfährt man, wenn man in das entpackte Archiv mit dem Explorer hinein navigiert bis der Python-Interpreter mit dem Namen `python` im Verzeichnis `python-3.7.7.amd64` erscheint:



Um den absoluten Pfad per Copy/Paste zu übertragen oben in der Zeile > Dieser PC > Dokumente > ... auf den Pfeil nach Unten klicken.

Diesen Pfad dann in das nächste Fenster unten eintragen nachdem auf das Festplattensymbol neben Python from another location geklickt wurde (im Vergleich zum Screenshot unten statt mackst den entsprechenden Usernamen angeben):



Für den Fall a) erscheint oberhalb von „Python from another location“ als Auswahl der Systemweite Pythoninterpreter, welcher statt des WinPython-Interpreters in dem Fenster oben ausgewählt wird.

Die folgenden Schritte/Fenster des Installationsprogramms installieren den M1K-Gerätetreiber. Hier immer auf OK bzw. Next klicken bis das Installationsprogramm zu Ende ist.

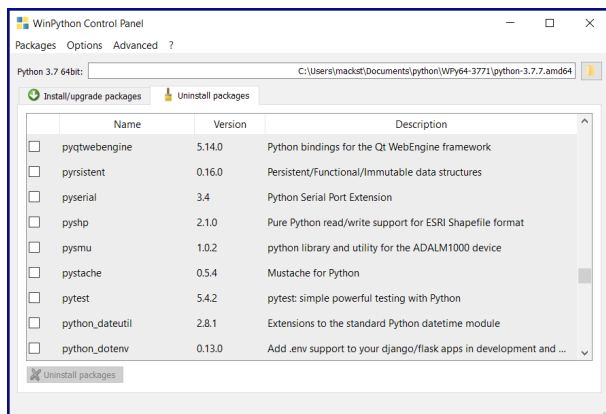
Aufruf von AliceLite unter Windows 10

a) Aus dem GitHub Repository github.com/StefanMack/PraktMesstADALM1000 die Python-Module von AliceLite in ein beliebiges Verzeichnis kopieren.

Gestartet wird AliceLite mit dem Befehl `python3 aliceLite.py` aus dem Windows Command-Fenster.

b) Die AliceLite-Dateien befinden sich im entpackten WinPython-Archiv. Im Verzeichnis `WPY - 3771` befindet sich eine Verknüpfung mit dem Namen `aliceLite`. Doppelklicken darauf starte das Programm AliceLite.

Bemerkung zu dem Festplatten-Speicherbedarf von WinPython



Die hier verwendete WinPython-Distribution enthält eine sehr große Anzahl von Python-Modulen, welche alle außer Numpy und pysmu für AliceLite nicht zwingend erforderlich sind. über das WinPython Control Panel (ein Programm im Verzeichnis `\WPY64 - 3771`) können die nicht benötigten Module deinstalliert werden um Speicherplatz freizugeben.



Ubuntu-Linux

Installation der Bibliotheken unter Ubuntu-Linux

Hierbei wird der unter Ubuntu schon vorhandene Python-Interpreter verwendet. Es wurden Ubuntu 18.04 und 20.04 erfolgreich getestet.

Die Bibliothek `libsmu` 1.0.3 für Linux (also eine Version höher als für Windows) gibt es als Debian-Installationspaket: <https://github.com/analogdevicesinc/libsmu/releases>

Weiter gibt es unter der gleichen Adresse auch ein Debian-Installationspaket für das Python-Modul `pysmu`.

Name	Größe
 <code>python3-pysmu_1.0.3-1-ubuntu-20.04-amd64.deb</code>	120,2 kB
 <code>libsmu-1.0.3-ubuntu-20.04-amd64.deb</code>	246,9 kB

Beide Debian-Dateien herunterladen und diese dann mit folgenden Befehlen im Kommandofenster installieren:

```
sudo apt install -f ./libsmu-1.0.3-ubuntu-20.04-amd64.deb
```

```
sudo apt install -f ./python3-pysmu_1.0.3-1-ubuntu-20.04-amd64.deb
```

Wie unter <https://github.com/analogdevicesinc/libsmu> beschrieben wird pysmu nur für die Standard-Python-Version der jeweiligen Ubuntu-Version installiert: Bei Ubuntu 18 ist das Python 3.6 und bei Ubuntu 20 Python 3.8.

Ob die Installation von libsmu und pysmu erfolgreich war, wird über das Importieren von pysmu im Python-Interpreter getestet:

```
stefan@LenovoLaptop:~/Downloads$ python3
Python 3.8.2 (default, Jul 16 2020, 14:00:26)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pysmu
>>>
```

Im Beispiel oben wurde dies für Ubuntu 20 durchgeführt. Funktioniert das Importieren wie hier ohne Fehlermeldung, dann wurden die beiden Bibliotheken korrekt installiert.

Falls folgende Fehlermeldung auftaucht, dass libsmu nicht gefunden wurde:

```
ImportError: libsmu.so.1: cannot open shared object file: No such file or directory)
```

Muss die Liste der installierten Bibliotheken mit folgendem Befehl aktualisiert werden:

```
sudo ldconfig
```

Wenn Aufgrund fehlender Berechtigung auf das M1K nicht zugegriffen werden kann, muss der User mit Namen username in die entsprechende Gruppe aufgenommen werden:

```
sudo usermod -a -G plugdev username
```

Alternativ kann für eine unter Ubuntu schon vorhandene beliebige Python-Version (3.6 und 3.7 wurden erfolgreich getestet) das Python-Modul pysmu auch selbst erstellt (kompiliert) werden, siehe: <https://github.com/analogdevicesinc/libsmu>. Der vorher beschriebene Weg über das Debian-Installationspaket ist jedoch wesentlich einfacher und weniger fehlerträchtig.

Starten von AliceLite unter Ubuntu

Sind die Bibliotheken installiert, müssen noch die acht AliceLite-Quellcode-Dateien von der GitHub-Seite <https://github.com/StefanMack/M1K/tree/master/aliceLite> in ein Verzeichnis kopiert werden.

Ist dies geschehen, dann wird mit dem Befehl `python3 aliceLite.py` das Hauptmodul aufrufen und damit das Programm AliceLite gestartet.

Aktualisieren von AliceLite unter Windows 10 oder Ubuntu Linux

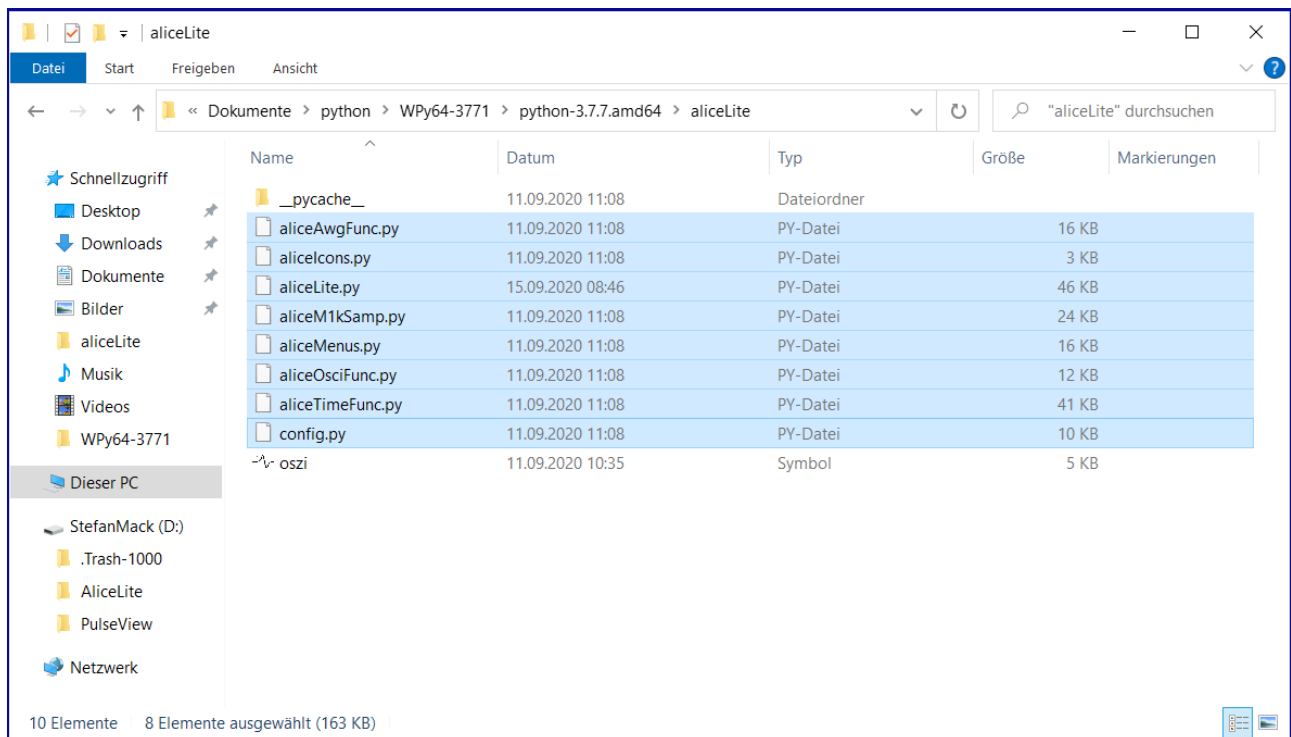
Die aktuellen Quellcodedateien sind unter <https://github.com/StefanMack/M1K/tree/master/aliceLite> zu finden.

Für ein Update müssen die vorhandenen Quellcodedateien durch die aktuellen ersetzt werden. Unter Ubuntu Linux oder bei der Windows-Installationsvariante a) wurde das Quellcodeverzeichnis explizit erstellt und ist somit bekannt.

Im Fall der Windows-Installationsvariante b) (WinPython) befinden sich die Quellcodedateien unter `\WPY64-3771\python-3.7.7.amd64\aliceLite`

Für die hier dargestellte Beispielinstallation zeigt der nachfolgende Screenshot die zu ersetzenden Quellcodedateien im o.g. Ordner.

Das Ersetzen erfolgt ganz normal z.B. im Dateixplorer via Copy/Paste.



Firmware des M1K flashen und M1K kalibrieren

Hierfür gibt es unter Windows wie auch unter Linux das Kommandozeilenprogramm smu.

Achtung: Die in den Praktika verwendeten M1K besitzen die aktuelle Firmware 2.17 und sind für die Spannungsmessung kalibriert. Unbedingt an diesen M1K-Platinen keine Änderungen an der Firmware oder Kalibrierung vornehmen.

Firmware flashen

Die Software alceLite startet nur unter der M1K-Firmware 2.17.

Falls sich auf dem M1K nicht diese Firmwareversion befindet muss die zugehörige Binärdatei zuerst unter github.com/analogdevicesinc/m1k-fw herunter geladen werden.

Nach Entpacken dieses Archivs im Terminalfenster in das Verzeichnis mit der Binärdatei m1000.bin navigieren und diese mit folgendem Befehl auf das M1K übertragen:

```
smu -f m1000.bin
```

Dieser Befehl muss in der Regel zweimal ausgeführt werden, da beim ersten Mal das M1K lediglich in den Programmiermodus wechselt.

Kalibrieren

Für das Kalibrieren muss eine Kalibrierdatei in Textform vorliegen, siehe wiki.analog.com/university/tools/m1k/calibration.

Im Anhang ist die für die M1K im Praktikum verwendete Kalibrierdatei `calib.txt` wiedergegeben. Mit der Kalibrierdatei `calib_zero.txt` können vor den Kalibriermessungen alle Kalibrierwerte auf Null gesetzt werden.

Die aktuell im M1K verwendete Kalibrierdatei wird mit `smu -d` ausgelassen.

Eine neue Kalibrierdatei wird mit `smu -w calib.txt` in das M1K geschrieben.

Anhang:

Standardkalibrierdatei der M1K im Praktikum

```
# Channel A, measure V
</>
<0.0000, 0.0009>
<2.4355, 2.3784>
<\>

# Channel A, measure I
</>
<0.0000, 0.0000>
<0.1000, 0.1000>
<-0.1000, -0.1000>
<\>

# Channel A, source V
# Eingaben hier werden von Firmware falsch in Gain/Offset umgerechnet
# Daher besser unverändert lassen, weil AWG 5 V an sich genauer als 0,2 %
</>
<0.0000, 0.0000>
<2.5000, 2.5000>
<\>

# Channel A, source I
</>
<0.0000, 0.0000>
<0.1000, 0.1000>
<-0.1000, -0.1000>
<\>

# Channel B, measure V
</>
<0.0000, 0.0006>
<2.4355, 2.3776>
<\>

# Channel B, measure I
</>
<0.0000, 0.0000>
<0.1000, 0.1000>
<-0.1000, -0.1000>
<\>

# Channel B, source V
# Eingaben hier werden von Firmware falsch in Gain/Offset umgerechnet
# Daher besser unverändert lassen, weil AWG 5 V an sich genauer als 0,2 %
</>
<0.0000, 0.0000>
<2.5000, 2.5000>
<\>

# Channel B, source I
</>
<0.0000, 0.0000>
<0.1000, 0.1000>
<-0.1000, -0.1000>
<\>
```