

Protokollierung einer seriellen Schnittstelle auf Basis eines RaspberryPi 4

Samuel Brunner

March 16, 2021

1 Funktionsumfang

Dieses System dient zur Protokollierung einer seriellen RS232 Ausgabeschnittstelle. Logfiles werden dabei automatisch auf einen USB-Stick mit Timestamp gesichert. Um eine unterbrechungsfreie, zuverlässige und autarke Funktion zu gewährleisten wird eine Powerbank als USV vorgeschaltet. Es können Einwegschnittstellen, wie z.B. Debugausgabeports, aber auch Kommunikation zwischen Systemen, ohne diese zu beeinflussen, mitgeschnitten werden.

2 Hardwarevorbereitung

Figure 1: Gesamtansicht



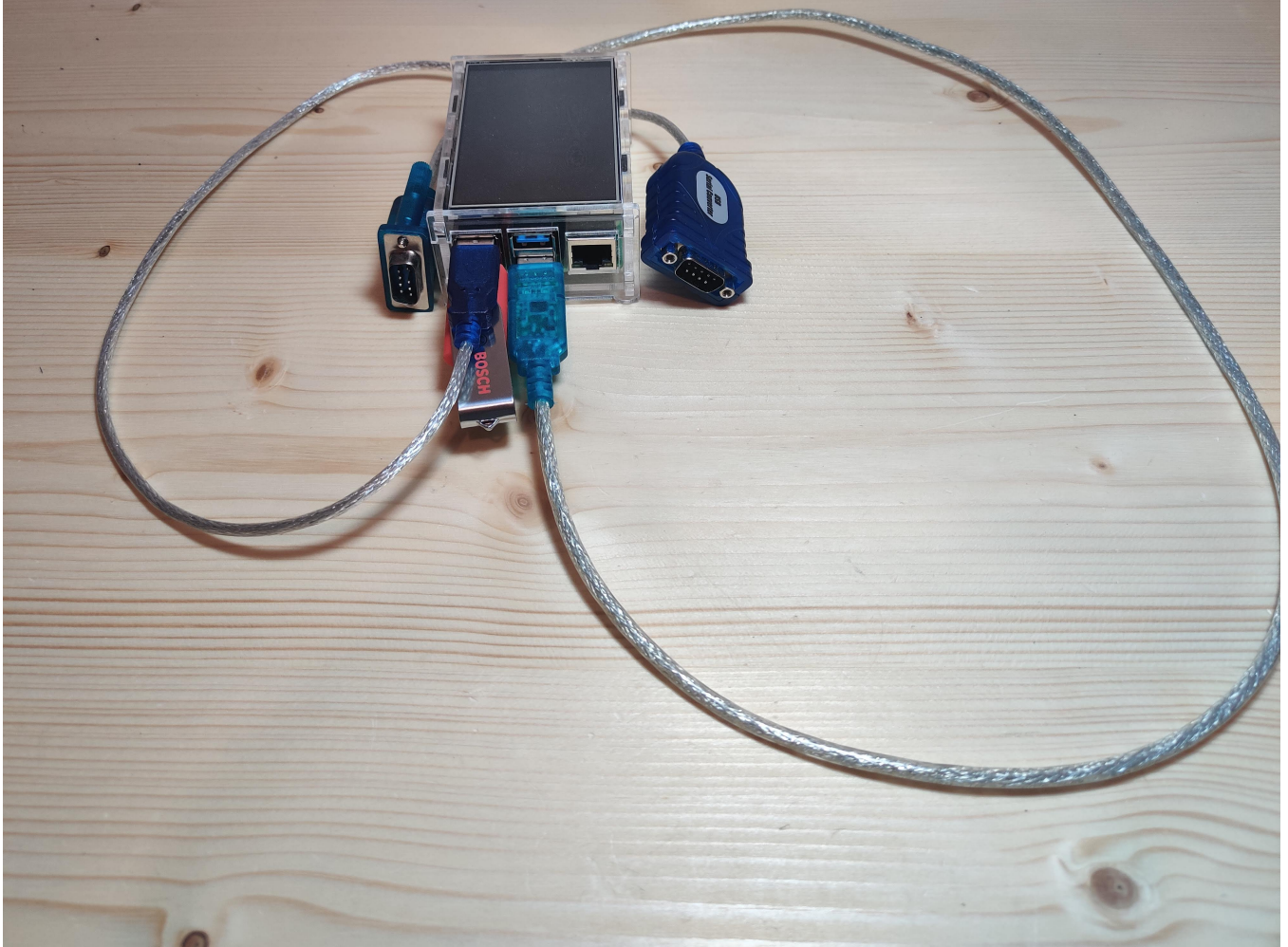
Figure 2: Da muass no de richtige Powerbank nei! De hat koa pass-through-charging



Figure 3: USB-RS232-Adapter und USB-Stick an beliebige USB-Ports anstecken



Figure 4: Nur für bidirektionales logging: Zwei USB-RS232-Adapter und USB-Stick an beliebige USB-Ports anstecken



3 Konfiguration

Um eine serielle Schnittstelle auslesen zu können, müssen zuerst einige Parameter im Logging-Skript angepasst werden. Dazu gehören unter anderem, Speed(Baudrate), Anzahl der Datenbits, Anzahl der Stopbits und Parität der seriellen Schnittstelle. Diese Konfigurationen werden über SSH oder VNC vorgenommen.

In beiden Fällen muss zuerst die IP-Adresse des eigenen Laptops auf eine IP-Adresse geändert werden, die im Subnet des RaspberryPi liegt und nicht mit anderen Adressen kollidiert.

IP des RaspberryPi 192.168.0.2/24

Freie IP-Adressen 192.168.0.3 - 192.168.0.254

Zugangsdaten user: pi; password: 13571357

3.1 Anpassung des Logging-Skriptes an serielle Schnittstelle und Anwendungsfall

3.1.1 Möglichkeit 1: SSH

Für diese Variante sollten wissen, wie eine SSH-Verbindung aufgebaut wird. Außerdem wird ein Verständnis über die Bedienung von Linux per SSH-Terminal vorausgesetzt. Erfüllen Sie diese Kriterien nicht fahren Sie bitte mit "Möglichkeit 2: VNC" fort.

- Die SSH Verbindung wird über den Standardport 22 und mit obiger IP hergestellt.
- Hierfür muss die Software Putty installiert werden. Putty liegt auf dem beiliegenden USB-Stick im Ordner Software/Putty.
- Verbinden Sie sich bitte mit dem Raspi. (user: pi; passwort: 13571357)
- Jetzt muss ihr Anwendungsfall ausgewählt werden.
- Zum Mitschneiden einer Output-Schnittstelle führen Sie bitte folgendes Kommando aus:

```
cd /home/pi/Desktop/serial_logging/ && git stash && git checkout master && cd /home/pi/Desktop/
```

Moeglicher Output; beides ist OK:

```
1.  
Switched to branch 'master'  
Your branch is up to date with 'origin/master'.
```

```
2.  
Already on 'master'  
Your branch is up to date with 'origin/master'.
```

- Zum Mitschneiden einer Schnittstelle zwischen zwei Teilnehmern führen Sie bitte folgendes Kommando aus:

```
cd /home/pi/Desktop/serial_logging/ && git stash && git checkout bidirectional && cd  
/home/pi/Desktop/
```

Moeglicher Output; beides ist OK:

```
1.  
Switched to branch 'bidirectional'  
Your branch is up to date with 'origin/bidirectional'.
```

```
2.  
Already on 'bidirectional'  
Your branch is up to date with 'origin/bidirectional'.
```

- Die Parameter der zu protokollierenden Schnittstelle werden direkt in das Skript eingetragen (Zeilen 11-19; siehe unten).
- Um das Skript zu bearbeiten stehen *vim* und *nano* zur Verfügung.
- Siehe Anhang 5.1 für alle möglichen Parameter.

```
#!/home/pi/Desktop/logging.py  
#!/home/pi/Desktop/logging_bidirectional.py  
#Line 11-19
```

```
#####  
# Hier Parameter fuer die serielle Schnittstelle eintragen!!!
```

```
#=====
baudrate = 9600
parity = serial.PARITY_NONE
stopbits = serial.STOPBITS_ONE
bytesize = serial.EIGHTBITS
timeout = .1
#=====
```

- Das Skript muss als root mit python3 ausgeführt werden.

```
sudo python3 /home/pi/Desktop/serial_logging/logging.py
```

- Damit das Skript nicht mit dem Ende der SSH-Session beendet wird, kann *screen* verwendet werden.

3.1.2 Möglichkeit 2: VNC mit grafischer Oberfläche

- Installation der Software VNC Viewer. VNC Viewer liegt auf dem beiliegenden USB-Stick im Ordner Software/vnc_viewer.
- Starten Sie bitte nun den VNC Viewer.
- Um nun die Verbindung zum RaspberryPi herzustellen, folgen Sie bitte den Anweisungen der Screenshots auf den nächsten Seiten.

Figure 5: IP-Adresse eingeben und mit Enter bestätigen. Nicht rechts auf Anmelden!

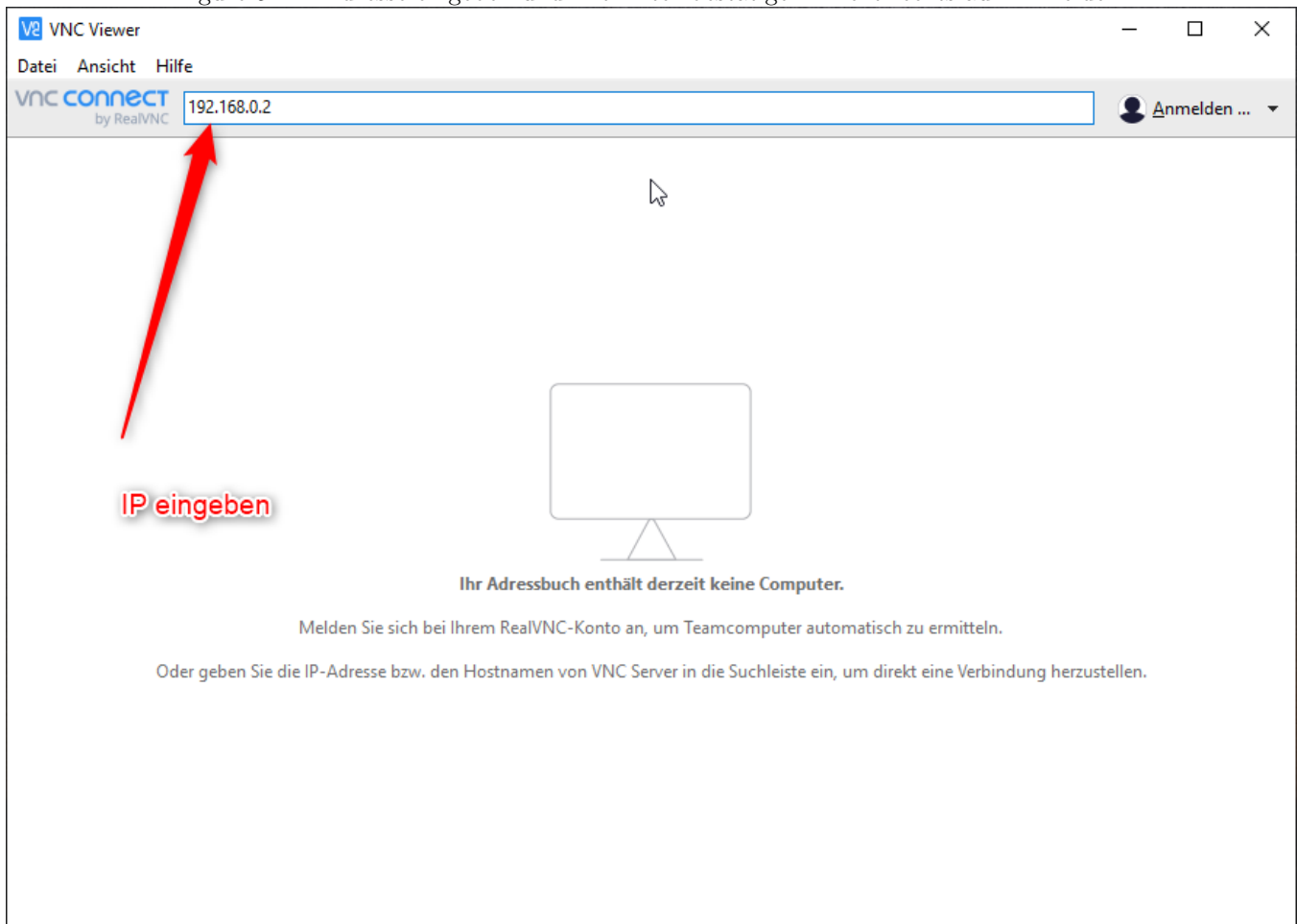


Figure 6: Als Benutzer "pi" eingeben; als Passwort "13571357" eingeben; mit ENTER bestätigen.

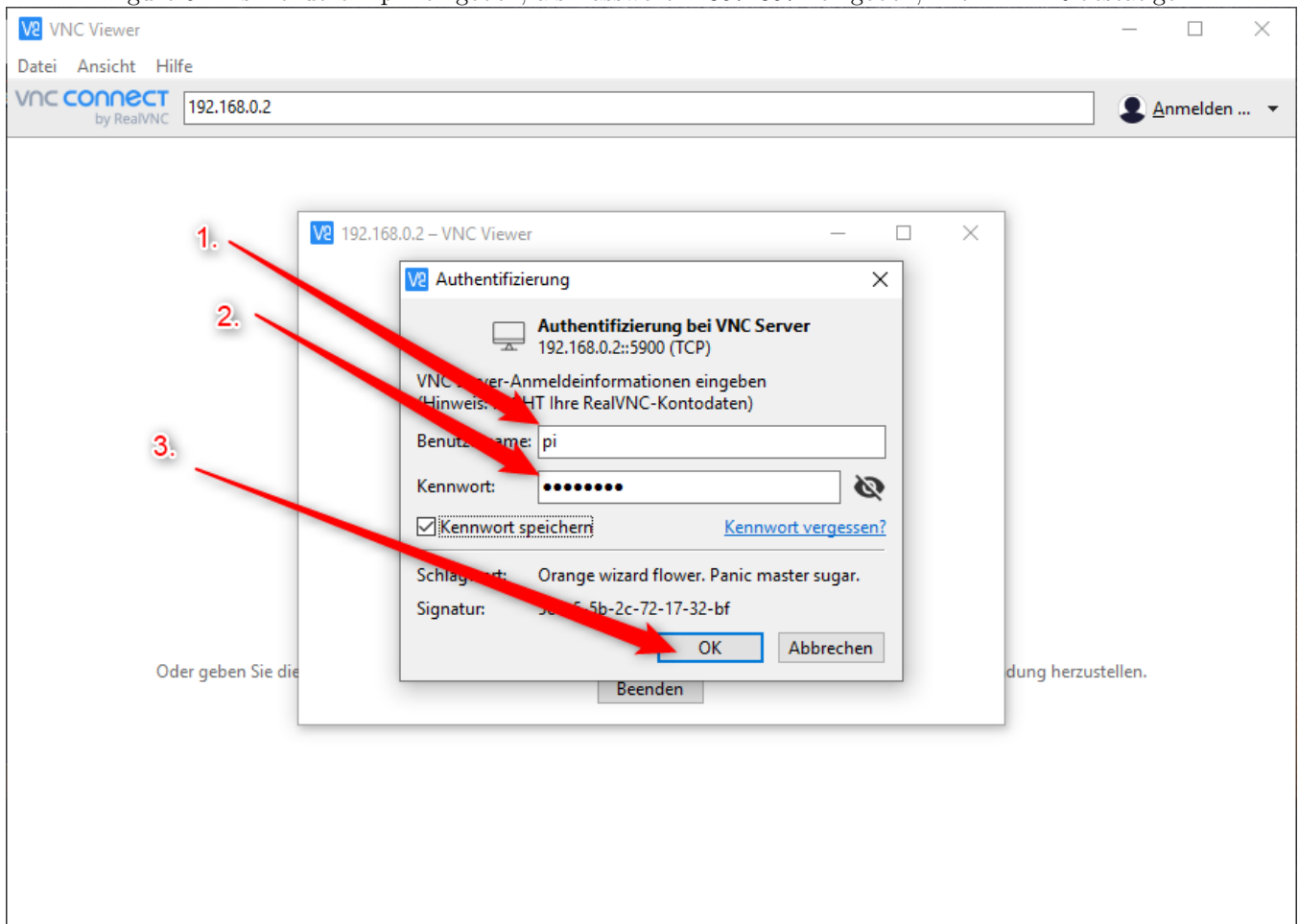
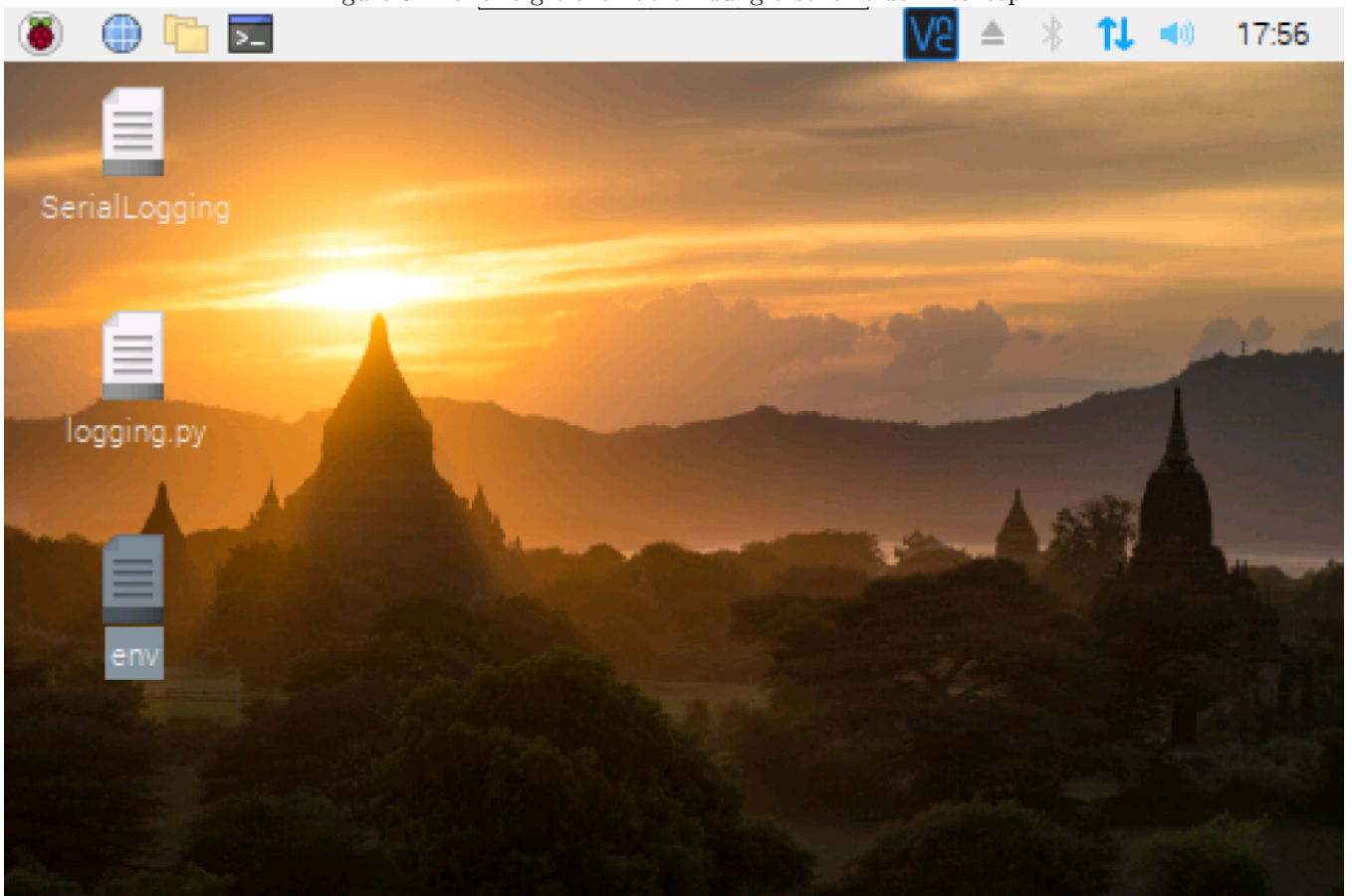


Figure 7: Bei erfolgreicher Verbindung erscheint der Desktop.



- Jetzt muss ihr Anwendungsfall ausgewählt werden.
- Öffnen Sie bitte ein Terminal mit STRG+ALT+T
- Zum Mitschneiden einer Output-Schnittstelle führen Sie bitte folgendes Kommando aus:

```
cd /home/pi/Desktop/serial_logging/ && git stash && git checkout master && cd /home/pi/Desktop/
```

Möglicher Output; beides ist OK:

```
1.
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

2.
Already on 'master'
Your branch is up to date with 'origin/master'.
```

- Zum Mitschneiden einer Schnittstelle zwischen zwei Teilnehmern führen Sie bitte folgendes Kommando aus:

```
cd /home/pi/Desktop/serial_logging/ && git stash && git checkout bidirectional && cd /home/pi/Desktop/
```

Möglicher Output; beides ist OK:

```
1.
Switched to branch 'bidirectional'
Your branch is up to date with 'origin/bidirectional'.

2.
Already on 'bidirectional'
Your branch is up to date with 'origin/bidirectional'.
```

- Das Terminal kann nun wieder geschlossen werden.
- Die Parameter der zu protokollierenden Schnittstelle werden direkt in das Skript eingetragen (Zeile 11-19).
- Wechseln Sie in den Ordner serial_logging auf dem Desktop
- Rechtsklick auf die Datei logging.py
- Änderungen Durchführen wie unten beschrieben. Siehe Anhang 5.1 für alle möglichen Parameter.
- Editor mit "X" beenden
- Änderungen mit "save" übernehmen
- Dateixplorer mit "X" beenden

```
#!/home/pi/Desktop/logging.py
#!/home/pi/Desktop/logging_bidirectional.py
#Line 11-19
```

```
=====
# Hier Parameter fuer die serielle Schnittstelle eintragen!!!
=====
baudrate = 9600
parity = serial.PARITY_NONE
stopbits = serial.STOPBITS_ONE
```

```
bytesize = serial.EIGHTBITS
timeout = .1
```

```
#=====
```

4 Ausführen des Logging-Skriptes

VNC

- Doppelklick auf SerialLogging auf dem Desktop
- "Execute" im sich öffnenden Fenster.

SSH

- Das Skript wird bei normaler Ausführung bei Beendigung der SSH-Session terminiert.
- Um das Skript weiterlaufen zu lassen, kann z.B. Screen verwendet werden.

```
screen sudo python3 /home/pi/Desktop/serial_logging/logging.py
```

Detach

To detach this terminal session, press

CTRL + A release, **and** then press D

Then you are back **in** the original terminal screen with the other one running detached **in** the background.

List **all** Instances

You can **list all open** screen instances **and** their status by typing...

```
screen -list
```

Reconnect

...**and** you can reconnect to an instance with...

```
screen -r
```

If you only have one screen instance **open**, just **-r** will be enough. If you have more than one, you have to specify which one you want to reconnect with by typing its name after the **-r**

5 Anhang

5.1 Parameter für serielle Schnittstellen

Standardparameter

- baudrate (int): Baud rate such as 9600 or 115200 etc.
- bytesize: Number of data bits. Possible values: FIVEBITS, SIXBITS, SEVENBITS, EIGHTBITS
- parity: Enable parity checking. Possible values: PARITY_NONE, PARITY_EVEN, PARITY_ODD, PARITY_MARK, PARITY_SPACE
- stopbits: Number of stop bits. Possible values: STOPBITS_ONE, STOPBITS_ONE_POINT_FIVE, STOPBITS_TWO

Weitere Parameter (nur über direkten Eintrag ins Skript)

- `timeout` (float): Set a read timeout value.
- `xonxoff` (bool): Enable software flow control.
- `rtscts` (bool): Enable hardware (RTS/CTS) flow control.
- `dsrdtr` (bool): Enable hardware (DSR/DTR) flow control.
- `write_timeout` (float): Set a write timeout value.
- `inter_byte_timeout` (float): Inter-character timeout, None to disable (default).
- `exclusive` (bool): Set exclusive access mode (POSIX only). A port cannot be opened in exclusive access mode if it is already open in exclusive access mode.