

CAN / UART / MODBUS Quick Reference

Industrielle Schnittstellen unter Linux auf einer Seite

Konkrete Beispiele mit `can0`, `/dev/beiis-uart-i-a` und `/dev/beiis-modbus-ii-a`.

Änderungen an Interfaces und Benutzerrechten benötigen normalerweise `sudo`.

1. SocketCAN

Interface finden und konfigurieren

```
ip -br link show type can    CAN-
                             Interfaces
                             auflisten

sudo ip link set can0
down                        Vor Änderun-
                             gen stoppen
sudo ip link set can0
type can                    Classic CAN
                             mit 500
                             kbit/s
bitrate 500000
sudo ip link set can0
type can                    CAN FD: 1 / 8
                             Mbit/s
bitrate 1000000 dbitrate
8000000 fd on
sudo ip link set can0 up    Interface
                             starten
```

Empfangen, Senden, Last erzeugen

```
candump -tz -x can0        Frames mit
                             Zeitstempel
                             anzeigen

candump can0,123:7FF       Nur CAN-ID
                             0x123
cansend can0               Classic-CAN-
123#DEADBEEF              Frame senden
cansend can0               CAN-FD-Frame
123##1DEADBEEF            mit BRS
cangen can0 -g 10 -I i     Alle 10 ms Test-
-L 8 -D i                 frame senden
```

Status und Fehleranalyse

```
ip -details -statistics    Bitrate, Zu-
link show can0             stand und
                             Zähler
candump -e can0            Error-Frames
                             lesbar anzei-
                             gen
sudo ip link set can0
type can                    Bus-off au-
                             tomatisch
                             verlassen
restart-ms 100              Kernelmeldungen
journalctl -k -g can0      zum Inter-
                             face
```

Keine Frames? Bitrate, Terminierung, Massebezug und CAN-H/CAN-L prüfen.

2. UART / TTY

Gerät finden

```
ls -l /dev/serial/by-id/    Stabile
                             USB-Seriell-
                             Namen
ls -l /dev/beiis-uart-i-a    Ziel und
                             Rechte des
                             Alias prüfen
journalctl -k -g tty         Erkannte
                             TTY-Geräte
udevadm info -q property    Udev-
-n                            Eigenschaften
/dev/beiis-uart-i-a          anzeigen
```

115200 Baud, 8N1, ohne Flow Control

```
stty -F /dev/beiis-uart-i-a Port konfi-
115200 cs8 -cstopb -parenb   gurieren
-ixon -ixoff -crtstcts raw
-echo
stty -F /dev/beiis-uart-i-a Aktuelle
-a                             Einstellun-
                             gen
picocom -b 115200            Interaktives
/dev/beiis-uart-i-a          Terminal
printf 'AT\r\n' >           Text mit
/dev/beiis-uart-i-a          CR/LF
                             senden
timeout 5 hexdump -C <      Fünf Se-
/dev/beiis-uart-i-a          kunden
                             binär le-
                             sen
```

Rechte und belegte Ports

```
groups                       Eigene
                             Gruppen
                             anzeigen
sudo usermod -aG dialout     Seriellen
$USER                        Zugriff er-
                             lauben;
                             neu anmel-
                             den
fuser -v                     Prozess
/dev/beiis-uart-i-a          finden, der
                             den Port
                             nutzt
sudo systemctl stop          Konsolen-
serial-getty@                Getty bei
ttyAMA0.service              Bedarf
                             stoppen
Unlesbare Zeichen bedeuten meist falsche Baudrate, Pa-
rität, Daten- oder Stopbits.
```

3. Modbus mit mbpoll

Modbus RTU lesen und schreiben

```
mbpoll -m rtu -a 1 -b 19200 Holding-
-P even -r 1 -c 4 -t 4       Register
/dev/beiis-modbus-ii-a       1-4 le-
                             sen
mbpoll -m rtu -a 1 -b 19200 Input-
-P even -r 1 -c 4 -t 3 -1    Register
/dev/beiis-modbus-ii-a       einmal
                             lesen
mbpoll -m rtu -a 1 -b 19200 Holding-
-P even -r 10 -t 4           Register
/dev/beiis-modbus-ii-a 123   10 auf
                             123 set-
                             zen
```

Modbus TCP

```
mbpoll -m tcp -a 1 -r 1      Vier
-c 4 -t 4 192.168.1.50      Holding-
                             Register
                             lesen
nc -vz 192.168.1.50 502      TCP-Port
                             502 prüfen
sudo tcpdump -ni eth0 tcp    Modbus-
port 502                     TCP-
                             Verkehr
                             beobach-
                             ten
```

Datentypen und Adressierung

```
-t 0                         Coils lesen/schreiben
-t 1                         Discrete Inputs lesen
-t 3                         Input-Register lesen
-t 4                         Holding-Register le-
                             sen/schreiben
-1                            Nur einen Poll ausführen
-0                            PDU-Adresse ab 0 statt
                             Referenz ab 1
-v                            Ausführliche Diagnose
```

Typische Modbus-Fehler

Timeout: Slave-ID, Baudrate, Parität, A/B, Masse und Antwortzeit prüfen.

CRC: Serielle Parameter, Terminierung, Schirmung und Störungen prüfen.

Exception: Funktion, Registerbereich und Schreibbe-
rechtigung prüfen.

Wichtig: Vier häufige Ursachen für Kommunikationsfehler

SocketCAN

Alle Teilnehmer brauchen dieselbe Bitrate. Das Interface vor Änderungen immer auf `down` setzen.

Modbus

Dokumentationen mischen oft 40001-Notation, Referenzen ab 1 und PDU-Adressen ab 0. `-0` bewusst verwenden.

UART

Baudrate, Datenbits, Parität, Stopbits und Flow Control müssen auf beiden Seiten übereinstimmen.

RS-485

Nur die beiden Busenden terminieren. A/B-Bezeichnung ist zwischen Herstellern nicht immer konsistent.