Diskstation

Ich habe vor meine Diskstation als SMI-Server zu benutzen.
Die Verbindung erfolgt über USB-Kabel. Für den SMI-Bus sind es zwei „USB→TTL-Serial“-Kabel die an einem selbstgebauten Interface angeschlossen sind. Die Schaltaktoren sind über ein „USB→RS485“-Kabel angebunden.

Hier erst mal mein Stand der Dinge (DS212+ mit DSM 5.2):

==== Bootstrap ===== (gilt nicht für Diskstation Pakete (SPK) Erst muss einmal IPKG installiert werden um weitere Pakete zu installieren. Diese Anleitung hat schon mal ganz gut funktioniert.

Allerdings muss dann der Pfad noch um ':/opt/bin:/opt/sbin' erweitert werden damit die installierten Programme auch ohne /opt/bin/ gestartet werden können: `export PATH='$PATH:/opt/bin:/opt/sbin'`

oder besser gleich in /root/.profile mit `vi /root/.profile` ändern

Pakete zum Testen

```
IPKG install stty
```
~~IPKG install screen~~
…

Erster Testlauf

Ich habe ein FTDI-USB-Kabel das problemlos erkannt wurde:
```
DS212> dmesg |grep ttyUSB
[158612.320000] usb 1-1.3: FTDI USB Serial Device converter now attached to
ttyUSB0
```

Wenn das nicht klappt kann es sein das die Module für diesen USB-Chip nicht geladen sind. Das lässt sich nach dieser Anleitung aber leicht beheben:
```
insmod /lib/modules/usbserial.ko
insmod /lib/modules/ftdi_sio.ko
```

Neuen USB-Port auf 2400 Baud einstellen:
```
stty -F /dev/ttyUSB0 speed 2400
```

Motor mit ID 0xC hochfahren lassen:
```
echo -e '\x5c\x02\xA1' > /dev/ttyUSB0
```

wer will kann auch ankommende Daten mitlesen. Dazu gibt es gleich mehrere Möglichkeiten: Die HEX-ausgabe funktioniert allerdings bei mir nicht weil `OD` bei mir zu alt ist…
```
/opt/bin/screen /dev/ttyUSB0 2400
cat < /dev/ttyS0
cat /dev/ttyS0
od -x < /dev/ttyS0
```

Feste Portzuordnungen

Die USB-Seriell-Adapter können unter Linux nach einem Neustart anders zugeordnet werden. Wenn mehrere USB-Adpater verwendet werden kann nicht garantiert werden dass der selbe Adapter wieder unter /dev/ttyUSB0 angesprochen werden kann.

Als erstes alle USB-Seriell-Adapter einstecken und nachsehen ob und unter wechem Namen sie erkannt werden:
```
pi@raspberrypi:~ $ dmesg | grep ttyUSB
[ 4.769458] usb 1-1.2: FTDI USB Serial Device converter now attached to
ttyUSB0
[ 4.776933] usb 1-1.4: FTDI USB Serial Device converter now attached to
ttyUSB1
```
Dann muss noch festgestellt was die beiden Adapter unterscheidet:
```
udevadm info --name=/dev/ttyUSB0 --attribute-walk
```
und
```
udevadm info --name=/dev/ttyUSB1 --attribute-walk
```

Dort dann durch die Einträge zu den Einträgen der USB-Adpater gehen und dort die PID,VID und Seriennummer suchen. Bei mir waren es diese Werte:

```
ATTRS{idVendor}=="0403"
ATTRS{idProduct}=="6001"
ATTRS{serial}=="AL0123MH"
```

jetzt muss noch die Datei **/etc/udev/rules.d/99-usb-serial.rules** erstellt werden. Für jeden USB-

Adapter muss dann eine entsprechende Zeile eingefügt werden. Bei mir waren es diese zwei Zeilen

```
SUBSYSTEM=="tty", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001",
ATTRS{serial}=="AL0123MH" , SYMLINK+="ttySWB0"
SUBSYSTEM=="tty", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001",
ATTRS{serial}=="FTVB965T" , SYMLINK+="ttySMI0"
```

Eventuell kann man auch an die Seriennummern auf anderen Wege kommen:

```
ls -la /dev/serial/by-id|grep ttyUSB
...
lrwxrwxrwx 1 root root 13 Jan 14 22:09 usb-FTDI_FT232R_USB_UART_AL0123MH-
if00-port0 -> ../../ttyUSB0
lrwxrwxrwx 1 root root 13 Jan 14 22:09 usb-FTDI_TTL232RG-VSW5V0_FTVB965T-
if00-port0 -> ../../ttyUSB1
```

Jetzt können die Einstellungen auch getestet werden:
`sudo udevadm trigger`

Bei Erfolg sieht es dann so aus:

```
ls -la /dev |grep ttyUSB
lrwxrwxrwx  1 root root           7 Jan 14 22:09 ttySMI0 -> ttyUSB1
lrwxrwxrwx  1 root root           7 Jan 14 22:09 ttySWB0 -> ttyUSB0
crw-rw----  1 root dialout 188,   0 Jan 14 22:09 ttyUSB0
crw-rw----  1 root dialout 188,   1 Jan 14 22:15 ttyUSB1
```

Jetzt haben wir zwei neue Links auf die ttyUSB0 und ttyUSB1. Wenn sich jetzt nach dem Neustart diese Zuordnungen ändern bleiben die Links die selben.

Pakete zum kompiliern

```
ipkg install wget
ipkg install bzip2
ipkg install zlib
ipkg install tar
ipkg install autoconf
ipkg install automake
ipkg install binutils
ipkg install gcc
...
```

SMI-Tools kompilieren

Erste Erfolge sind hier zu finden: https://github.com/ingof/smi-tools

SMI-Server kompilieren

Der SMI-Server ist ein Paket für die Synology Diskstation. Meine erste Test-Version gibt es hier: https://github.com/ingof/smi-server

From:
https://smiwiki.thefischer.net/ -

Permanent link:
https://smiwiki.thefischer.net/doku.php?id=wiki:smi:diskstation

Last update: **2018/05/20 16:42**