

Tutorial for the serial Com-Server:

TTY-Redirector/Pseudo-TTYs under Linux

Product overview

Application overview

Linux applications whose communication with serial devices is limited to local TTYs are made network-capable using a Com-Server and the GPL2-licensed open source tool *socat* (e.g. for bridging long distances, defecient or absent local TTYs, etc.). Whereas the Com-Server provides for the hardware side connection of the serial device to the network, *socat* provides virtual TTYs and pseudo TTYs on the Linux system and connects them to the Com-Server bi-directionally over the network using TCP.

1. Download and installation

- In many Linux distributions *socat* can be installed directly using the respective software management. Simply search for *socat*.
- The homepage of the *socat* project with download links, information about supported Linux distributions and comprehensive documentation can be found at <http://www.dest-unreach.org/socat/>. Suitable Com-Servers can be found [here](#).

2. Create pseudo-TTYs

- The following command creates the pseudo-TTY *csTTY1* and redirects it to the Com-Server 10.40.21.18 located in the network:

```
socat PTY,link=/dev/csTTY1,echo=0,wait-slave TCP:10.40.21.18:8000,retry=5
```

- The keyword *PTY* first creates the pseudo-TTY with the name *csTTY1* in the directory */dev*. The associated option *echo=0* deactivates the local echo of the TTY.
- By using *wait-slave* the TCP connection to the 2nd data channel is not established until the created pseudo-TTY is opened by an application, so that if a TTY is closed, other stations in the network can access the Com-Server and the connected serial device.
- The keyword *TCP* specifies the 2nd data channel, which is connected to the previously created pseudo-TTY. The IP address and the TCP port number of the Com-Server are specified as address parameters. *retry=5* specifies the number of connection attempts in case the Com-Server is currently being used by another station.

3. Notes and special features

- The required serial transmission parameters (baud rate, parity etc.) must be first configured manually in the Com-Server setup. Automatic configuration coupled to the parameters of the opening application is not possible.
- Via *socat* created pseudo-TTYs are useful for all applications which communicate with the terminal device only on the level of the serial data lines Rx and Tx. Transmission on the RS232 control lines (RTS, CTS, etc.) is not possible.
- The *wait-slave* option used in the example above provides for a need-based network connection opening. At the same time, closing the TTY and the network connection to the Com-Server also ends the *socat* process, so that a script-based restart must follow. When the *socat* command is executed as part of the system start, the procedure differs depending on the Init system of whichever Linux distribution is used.
- You can also perform a character-oriented test of the virtual TTY using *socat*. The following command creates a transparent connection between *STDIN/STDOUT* of the Linux system and the Com-Server through the pseudo-TTY *csTTY1*:

```
socat STDIO,raw,echo=0,escape=0x03 /dev/csTTY1,raw,echo=0
```

Do you not yet have a Com-Server but would like to try out the example shown?

No problem: we will be glad to provide a unit at no cost for 30 days. Order a Com-Server as a sample and we will ship you the device for testing on open account. If you return the unit within 30 days, we will fully credit the invoice amount.

[to the Com-Servers](#) 

Still unsure which Com-Server is the right one for you?

Our qualified team of engineers will be glad to assist you.
You can reach us by telephone at +49 202/2680-110 or by email at info@wut.de.

none of our statements should be applied without verification. Please let us know of any errors or misunderstandings you find so that we can become aware of and eliminate them.

[Data Privacy](#)