



TCP866-SW-95
QNX6 – Neutrino Device Driver
8-Channel Serial cPCI Module

Version 1.0

Reference Manual
Issue 1.0

April 9, 2002

TEWS TECHNOLOGIES GmbH
Am Bahnhof 7
D-25469 Halstenbek
Germany
Tel.: +49 (0)4101 4058-0
Fax.: +49 (0)4101 4058-19
<http://www.tews.com>
e-mail: info@tews.com

TCP866-SW-95

8 Channel Serial CompactPCI Module QNX6 -Neutrino Device Driver

This document contains information, which is proprietary to TEWS TECHNOLOGIES GmbH. Any reproduction without written permission is forbidden.

TEWS TECHNOLOGIES GmbH has made any effort to ensure that this manual is accurate and complete. However TEWS TECHNOLOGIES GmbH reserves the right to change the product described in this document at any time without notice.

This product has been designed to operate in a CompactPCI system. Connection to incompatible hardware is likely to cause serious damage.

TEWS TECHNOLOGIES GmbH is not liable for any damage arising out of the application or use of the device described herein.

©2002 by TEWS TECHNOLOGIES GmbH

Issue	Description	Date
1.0	First Issue	April 9, 2002

Table of Contents

1	INTRODUCTION	4
2	INSTALLATION.....	5
2.1	Build the device driver	5
2.2	Start the driver process	6

1 Introduction

The TCP866-SW-95 QNX-Neutrino device driver is a full-duplex serial device driver which allows the operation of a TCP866 serial PMC on QNX6-Neutrino operating systems.

The TCP866-SW-95 device driver based on the standard QNX6 8250 serial communication manager. Due to this way of implementation the driver interface and function is compatible to the standard QNX6 serial device manager.

All standard utility programs for configuration (e.g. *stty*) and maintaining terminal interfaces could be used in the same manner.

Additional supported features:

- ✓ Extended baud rates up to 921.6 kBaud.
- ✓ Each channel has a 64 Byte transmit and receive FIFO
- ✓ Programmable trigger level for transmit and receive FIFO.
- ✓ Direct support of different physical interfaces (RS-232, RS-422).

2 Installation

The software is delivered on a DOS formatted 3½" diskette.

Following driver specific files are located on the diskette:

```
/driver/common.mk
/driver/externs.h
/driver/externs.c
/driver/init.c
/driver/intr.c
/driver/main.c
/driver/options.c
/driver/proto.h
/driver/teedit.c
/driver/tcp866.h
/driver/tto.c
/driver/nto/Makefile
/driver/nto/x86/Makefile
/driver/nto/x86/sys_ttyinit.c
/driver/nto/x86/o/Makefile
tcp866-sw-95.pdf
```

For installation create a new directory (e.g. `../tcp866`) in the `/usr/src` directory and copy (e.g. `mount -t dos /dev/fd0 /floppy`) the complete `/driver` directory (with sub-directories and all files) from the distribution diskette into the new created project directory.

Note

It's absolute important to create the `tcp866` project directory in the `/usr/src` directory otherwise the automatic build with `make` will fail.

Note

For building the device driver it is necessary that the serial DDK is installed. (Installer: "/QNX Realtime Platform/Software Development/Device Driver Kits/Character (Serial) DDK targeting x86")

2.1 Build the device driver

1. Change to the `/usr/src/tcp866/driver/nto` directory
2. Execute the Makefile

```
# make install
```

After successful completion the driver binary will be installed in the `/bin` directory.

2.2 Start the driver process

To start the TCP866 device driver respective the TCP866 serial communications manager you have to enter the process name with optional parameter from the command shell or in the startup script.

```
devc-tcp866 [options] &
```

OPTIONS

- b** *number* Initial baud rate (default 9600).
- C** *size* The size of the canonical buffer in bytes (default 256).
- E** Start in raw mode (the default). Software flow control is disabled by default.
- e** Start in edit mode (default raw). Software flow control is enabled by default.
- F** Disable hardware flow control (default to hardware flow control enabled).
- f** Enable hardware flow control (default). Only valid for TCP866-1x modules. Must be disabled for TCP866-2x modules.
- I** *number* The size of the interrupt input buffer in bytes (default 2048).
- O** *number* The size of the interrupt output buffer in bytes (default 2048).
- S|s** Disable / enable software flow control. The default depends on the mode: in raw mode (**-E**, the default), it's disabled; in edited mode (**-e**), it's enabled.
The order in which you specify the **-E** or **-e**, and **-S** or **-s** options matters:

Options	Mode	Software flow control
-e	Edited	Enabled
-S -e	Edited	Enabled
-e -S	Edited	Disabled
-E	Raw	Disabled
-s -E	Raw	Disabled
-E -s	Raw	Enabled

- r** *number* Set the receive FIFO trigger level. Valid settings for the receiver FIFO are: 1, 8, 16, 56 and 60 (default 56).
- t** *number* Set the transmit FIFO trigger level. Valid settings for the transmitter FIFO are: 1, 8, 16, 32 and 56 (default 8).
- u** *number* Append number to the device name prefix (*/dev/ser*). The default is 3, which mean the first TCP866 device is */dev/ser3*; additional device are given increasing numbers.
- v** Print out debug information.

Note

Most of the options above are standard options for serial communications manager. Please refer also to related QNX6 documentation if necessary.

DESCRIPTION

The devc-tcp866 manager is based on the standard QNX6 devc-ser8250 serial communications manager and can support any number of serial ports of the TCP866-1x modules.

The devc-tcp866 manager searches the entire PCIbus for TCP866-1x devices and creates devices for each serial channel (8 per module). The first device created depends on the `-u` option. If the `-u` option is omitted the first TPCM866 serial device is `/dev/ser3`. In this configuration the devices `/dev/ser3`, `/dev/ser4`, ...`/dev/ser10` will be created for the first TCP866, `/dev/ser11` ... `/dev/ser18` will be created for the second TCP866 and so on.

Usually the device names `/dev/ser1` and `/dev/ser2` are assigned to the default PC serial ports, therefore the TPCM866 devices can start with `/dev/ser3` (default). If there are additional onboard serial devices you have to start with a higher device number for the TCP866 devices by defining an appropriate number with the `-u` option (please check also the `/dev` directory).

All devices are fully interrupt driven and by default support standard hardware flow control on input and output (RTS/CTS) for TCP866-10. This can be disabled by the `-F` option for the TCP866-1x and **must** be disabled for TCP866-2x modules.

A read request by default returns when at least 1 character is available. To increase efficiency, you can control three parameters to control when a read is satisfied:

Time Return after a specified amount of time has elapsed (`c_cc[VTIME]`).
Min Return when this number of characters is in the input buffer (`c_cc[VMIN]`).
Char Return if the forwarding character is in the input buffer (`c_cc[VEND]`).

These parameters, and other, are set using library routines (see `tcgetattr()`, `txsetattr()`, `readcond()` and `TimerTimeout()` in the Library Reference).

The following fields and flags are supported in the *termios* structure.

Field	Supported fields and flags
<code>c_cc</code>	All characters
<code>c_iflag</code>	BRKINT ICRNL IGNBRK IXON
<code>c_oflag</code>	OPOST
<code>c_cflag</code>	CLOCAL CSIZE CSTOPB PARENB PARODD
<code>c_lflag</code>	ECHO ECHOE ECHOK ECHONL ICANON IEXTEN ISIG NOFLSH

EXAMPLES

Start device with default parameters:

- first created device is */dev/ser3*
- 9600 baud
- see also options above...

```
devc-tcp866 &
```

Start device with default parameters and change baud rate to 38400

```
devc-tcp866 -b 38400 &
```

Start device with default parameters. The first created device is */dev/ser5*.

```
devc-tcp866 -u 5 &
```