

*The Embedded I/O Company*



---

# TPCI868-SW-95

## QNX-Neutrino Device Driver

16 Channel Asynchronous Serial Interface

Version 1.0.x

## User Manual

Issue 1.0.0

December 2006

---

**TEWS TECHNOLOGIES GmbH**

Am Bahnhof 7  
25469 Halstenbek, Germany  
[www.tews.com](http://www.tews.com)

Phone: +49 (0) 4101 4058 0  
Fax: +49 (0) 4101 4058 19  
e-mail: [info@tews.com](mailto:info@tews.com)

**TEWS TECHNOLOGIES LLC**

9190 Double Diamond Parkway,  
Suite 127, Reno, NV 89521, USA  
[www.tews.com](http://www.tews.com)

Phone: +1 (775) 850 5830  
Fax: +1 (775) 201 0347  
e-mail: [usasales@tews.com](mailto:usasales@tews.com)

## TPCI868-SW-95

QNX-Neutrino Device Driver

16 Channel Asynchronous Serial Interface

Supported Modules:

TPCI868-10

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.

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

©2006 by TEWS TECHNOLOGIES GmbH

Issue	Description	Date
1.0.0	First Issue	December 18, 2006

---

## Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>4</b>
<b>2</b>	<b>INSTALLATION.....</b>	<b>5</b>
	2.1 Build the device driver .....	6
	2.2 Start the Driver Process .....	6

---

# 1 Introduction

The TPCI868-SW-95 QNX6 Neutrino device driver allows the operation of the TPCI868 PCI module conforming to the QNX6 Neutrino I/O system specification.

The TPCI868-SW-95 device driver is based on the standard QNX6 Neutrino 8250 serial communication manager. Due to this way of implementation the driver interface and function is compatible to the standard QNX6 Neutrino serial device manager.

The TPCI868-SW-95 device driver supports the following features:

- supporting standard terminal interface functions like stty()
- baudrates up to 921600 Baud (hardware limits)

The TPCI868-SW-95 device driver supports the modules listed below:

TPCI868	16 channel serial interface (RS232)	PCI module
---------	-------------------------------------	------------

To get more information about the features and use of TPCI868 devices it is recommended to read the manuals listed below.

- TPCI868 User manual
- TPCI868 Engineering Manual

## 2 Installation

Following files are located on the distribution media:

Directory path 'TPCI868-SW-95':

TPCI868-SW-95-SRC.tar	TAR archive with driver source code
TPCI868-SW-95-1.0.0.pdf	PDF copy of this manual
ChangeLog.txt	Release history
Release.txt	Release information

The TAR archive TPCI868-SW-95-SRC.tar contains the following files and directories:

Directory path './TPCI868/':

driver/main.c	TPCI868 device driver source (driver start up)
driver/options.c	TPCI868 device driver source (configurations)
driver/init.c	TPCI868 device driver source (device initialization)
driver/intr.c	TPCI868 device driver source (interrupt handling)
driver/tedit.c	TPCI868 device driver source
driver/tto.c	TPCI868 device driver source (I/O access)
driver/externs.c	external includes
driver/externs.h	external prototypes and includes
driver/proto.h	internal prototypes
driver/tcpi868.h	device and driver include file
driver/common.mk	Build parameters (QNX-Neutrino >= V6.3)
driver/common.mk-nopm	Build parameters (QNX-Neutrino < V6.3)
driver/nto/Makefile	recursive Makefile
driver/nto/x86/Makefile	recursive Makefile
driver/nto/x86/sys_ttyinit.c	TPCI868 device and system parameters
driver/nto/x86/nto/Makefile	recursive Makefile
example/tpci868exa.c	Example application
example/common.mk	Build parameters
example/Makefile	recursive Makefile
example/nto/Makefile	recursive Makefile
example/nto/x86/Makefile	recursive Makefile
example/nto/x86/o/Makefile	recursive Makefile

In order to perform an installation, extract all files of the archive TPCI868-SW-95.tar to the desired target directory.

**Its absolute important to create the TPCI868 project directory in the /usr/src directory otherwise the automatic build with make will fail.**

**For building the device driver it is necessary that the QNX serial DDK is installed.**

## 2.1 Build the device driver

- (1) Change to the `/usr/src/TPCI868/driver/nto` directory
- (2) Execute the Makefile

```
make install
```
- (3) After successful completion the driver binary will be installed in the `/bin` directory.

## 2.2 Start the Driver Process

To start the TPCI868 device driver respective the TPCI868 serial communications manager you have to enter the process name with optional parameters from the command shell or in the startup script.

```
devc-tpci868 [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).

**-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), its 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 TPCI868 device is */dev/ser3*; additional device are given increasing numbers.

**-v**

Print out debug information.

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

**DESCRIPTION**

The devc-tpci868 manager is based on the standard QNX6 devc-ser8250 serial communications manager and can support any number of serial ports and TPCI868 PMC modules.

The devc-tpci868 manager searches the entire PCI bus for TPCI868 devices and creates devices for each serial channel (16 per TPCI868 module). The first device created depends on the **-u** option. If the **-u** option is omitted the first TPCI868 serial device is */dev/ser3*. In this configuration the devices */dev/ser3*, */dev/ser4*, ...*/dev/ser18* will be created for the first TPCI868, */dev/ser19* ... */dev/ser34* will be created for the second TPCI868 and so on.

Usually the device names */dev/ser1* and */dev/ser2* are assigned to the default PC serial ports, therefore the TPCM868 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 TPCI868 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). This can be disabled by the **-F** option.

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()*, *tcsetattr()*, *readcond()* and *TimerTimeout()* in the Library Reference).

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

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

## EXAMPLES

Start the device driver with default parameters (first created device is /dev/ser3, 9600 baud, see also options above ...):

```
devc-tpci868 &
```

Start the device driver with default parameters and change baud rate to 38400

```
devc-tpci868 -b 38400 &
```

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

```
devc-tpci868 -u 5 &
```