

TIP866-SW-95

QNX-Neutrino Device Driver

8 Channel Serial Interface IP

Version 2.1.x

User Manual

Issue 2.1.1

October 2009

TEWS TECHNOLOGIES GmbH

Am Bahnhof 7 25469 Halstenbek, Germany
Phone: +49 (0) 4101 4058 0 Fax: +49 (0) 4101 4058 19
e-mail: info@tews.com www.tews.com

TIP866-SW-95

QNX-Neutrino Device Driver

8 Channel Serial Interface IP

Supported Modules:

TIP866

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.

©2002-2009 by TEWS TECHNOLOGIES GmbH

Issue	Description	Date
1.0	First Issue	February 11, 2002
2.0	Complete revised manual for new driver concept	October 14, 2004
2.0.1	Installation process description corrected	February 15, 2005
2.0.2	File list corrected	April 13, 2005
2.0.3	File list corrected	May 19, 2005
2.1.0	General revision	October 9, 2009
2.1.1	Description of configuration file parameters corrected	October 29, 2009

Table of Contents

1	INTRODUCTION	4
	1.1 IPAC Carrier Driver	5
2	INSTALLATION	6
	2.1 Build the device driver	6
	2.2 Start the driver process.....	7
3	CHANNEL CONFIGURATION FILES	8
	3.1.1 Configuration Parameters	8

1 Introduction

The TIP866-SW-95 QNX-Neutrino device driver allows the operation of a TIP866 8 Channel Serial Interface IP on QNX-Neutrino operating systems and requires the TEWS QNX-Neutrino IPAC Carrier Driver Software (CARRIER-SW-95).

The TIP866-SW-95 device driver is based on the standard QNX-Neutrino (ser8250) serial communication manager. Due to this way of implementation the driver interface and function is compatible to the standard QNX-Neutrino serial device manager.

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

The TIP866-SW-95 device driver supports the following features:

- Extended baud rates up to 460.8 kBaud. (TIP866-11/-20)
- 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, TTL, RS-422)
- Individual startup configuration for each channel

The TIP866-SW-95 device driver supports the modules listed below:

TIP866-10(-ET)	8 channel RS232 serial I/O	(IndustryPack®)
TIP866-11(-ET)	8 channel TTL serial I/O	(IndustryPack®)
TIP866-20(-ET)	8 channel RS422 serial I/O	(IndustryPack®)

To get more information about the features and use of TIP866 devices it is recommended to read the manuals listed below.

TIP866 User manual
TIP866 Engineering Manual
CARRIER-SW-95 User Manual

1.1 IPAC Carrier Driver

IndustryPack (IPAC) carrier boards have different implementations of the system to IndustryPack bus bridge logic, different implementations of interrupt and error handling and other differences. Also, the varying byte ordering (big-endian versus little-endian) of CPU boards will cause problems when accessing the IndustryPack I/O and memory spaces.

To simplify the implementation of IPAC device drivers which should work with every supported carrier board, TEWS TECHNOLOGIES has designed a so called Carrier Driver that hides all differences of different carrier boards under a well defined interface.

The TEWS TECHNOLOGIES IPAC Carrier Driver CARRIER-SW-95 is part of this TIP866-SW-95 distribution. It is located in the directory CARRIER-SW-95 on the corresponding distribution media.

This IPAC Device Driver requires a properly installed IPAC Carrier Driver. Due to the design of the Carrier Driver, it is sufficient to install the IPAC Carrier Driver once, even if multiple IPAC Device Drivers are used.

Please refer to the CARRIER-SW-95 User Manual for a detailed description on how to install and setup the CARRIER-SW-95 device driver, and for a description of the TEWS TECHNOLOGIES IPAC Carrier Driver concept.

2 Installation

Following files are located on the distribution media:

Directory path 'TIP866-SW-95':

TIP866-SW-95-SRC.tar.gz	GZIP compressed archive with driver source code
TIP866-SW-95-2.1.1.pdf	PDF copy of this manual
ChangeLog.txt	Release history
Release.txt	Release information

The GZIP compressed archive TIP866-SW-95-SRC.tar.gz contains the following files and directories:

Directory path 'tip866':

driver/tip866.c	Device driver source
driver/tip866def.h	Device driver include file
driver/t866_default.txt	Channel configuration file example
driver/Makefile	Recursive multiplatform build tree
driver/common.mk	
driver/nto/Makefile	
driver/nto/x86/Makefile	
driver/nto/x86/dll/Makefile	

For installation, copy the tar-archive TIP866-SW-95-SRC.tar.gz to /usr/src and extract all files (e.g tar -xvzf TIP866-SW-95-SRC.tar.gz). Afterwards, the necessary directory structure for the automatic build and the source files are available underneath the new directory called tip866.

Before building a new device driver, the TEWS TECHNOLOGIES IPAC carrier driver must be installed properly, because this driver includes the header files ipac_*.h, which are part of the IPAC carrier driver distribution. Please refer to the IPAC carrier driver user manual in the directory path CARRIER-SW-95 on the distribution media.

It is very important to extract the TIP866-SW-95-SRC.tar.gz in the /usr/src directory, because otherwise the automatic build with make will fail.

2.1 Build the device driver

Change to the directory `/usr/src/tip866/driver` and execute the Makefile

```
# make install
```

After successful completion the driver dynamic library `tip866.so` will be installed in the directory `/lib/dll`.

2.2 Start the driver process

To start the TIP866 resource manager you have to start the TEWS TECHNOLOGIES IPAC carrier driver. The IPAC carrier driver detects installed TEWS IPAC modules automatically and loads the appropriate driver dynamic libraries.

```
# ipac_class &
```

The TIP866 resource manager registers a device for each TIP866 channel (8 channels per module) in the QNX-Neutrinos pathname space under the following name, where n specifies the overall channel number of detected TIP866 modules. The devices `/dev/t866_1` to `/dev/t866_8` belongs the first found TIP866 the devices `/dev/t866_9` to `/dev/t866_16` belongs to second found TIP866 and so forth (please refer to the IPAC carrier driver manual for detailed information of the module search order):

```
/dev/t866_n
```

This pathname must be used in the application program to open a path to the desired TIP866 device.

For debugging purposes, you can start the IPAC carrier driver with the `-V` (verbose) option. Now the resource manager will print versatile information about TIP866 configuration and command execution on the terminal window. For further details about debugging, please see the IPAC carrier driver manual.

3 Channel Configuration Files

Setting up channel defaults is possible with individual configuration files. The relationship between TIP866 channel (device) and configuration file is established by the file name. The configuration file name has the format *t866_n.txt*, where *n* specifies the overall channel number (e.g. the configuration file *t866_8.txt* belongs to the device */dev/t866_8*). An example configuration file is part of this distribution (*driver/t866_default.txt*).

All configuration files must be copied into the */etc/IPAC_CARRIER* path.

3.1.1 Configuration Parameters

It is not necessary to specify all parameters, unspecified parameters will be configured to the default values (see example configuration file).

Each parameter must start with a '*' followed by an identifier. The value must be separated with a space or a tab character.

**NAME XYZ*

This parameter specifies the device name.

**BAUD n*

This parameter specifies the default baud rate.

**DATABITS n*

This parameter specifies the number of data bits. Supported values are 5, 6, 7, and 8.

**STOPBITS*

This parameter specifies the number of stop bits. Supported values are 1 and 2. (If 5 data bits are specified, 2 will select 1½ stop bits).

**PARITY*

This parameter specifies the parity mode. Allowed values are:

NO	No parity supported
EVEN	Even parity bit supported
ODD	Odd parity bit supported

**FIFORX*

This parameter specifies the trigger level of receive hardware FIFO. Supported values are 8, 16, 56, and 60.

**FIFOTX*

This parameter specifies the trigger level of transmit hardware FIFO. Supported values are 8, 16, 32, and 56.

**INBUF*

This parameter specifies the size to allocate for the software input buffer.

**OUTBUF*

This parameter specifies the size to allocate for the software output buffer.

***CANBUF**

This parameter specifies the size to allocate for the software canonical buffer.

***EDIT**

This parameter specifies if the edit mode is used or not (SWFLOW will be enabled). Valid values are *ON* and *OFF*.

***HWFLOW**

This parameter enables hardware handshake. Valid values are *ON* and *OFF*.

***SWFLOW**

This parameter enables software handshake. Valid values are *ON* and *OFF*.

Example

This example will setup the serial device `/dev/t866_1`. The communication parameter will be set to 9600 baud, 8 data bits, 1 stop bit, and no parity. The input FIFO trigger level will be set to 56 characters and the output FIFO trigger level will be set to 8 characters. 2048 will be allocated for input and output buffer; 128 byte will be allocated for the canonical buffer. The terminal driver will use the interface in raw mode without handshaking.

File: `/etc/IPAC_CARRIER/t866_1.txt`

```
*NAME          t866_  
*BAUD          9600  
*DATABITS      8  
*STOPBITS      1  
*PARITY        NO  
*FIFORX        56  
*FIFOTX        8  
*INBUF         2048  
*OUTBUF        2048  
*CANBUF        128  
*EDIT          OFF  
*HWFLOW        OFF  
*SWFLOW        OFF
```