

TIP865-SW-82

Linux Device Driver

4 Channel Serial IPAC

Version 1.3.x

User Manual

Issue 1.3.1

May 2010

TIP865-SW-82

Linux Device Driver

4 Channel Serial IPAC

Supported Modules:

TIP865

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.

©2003-2010 by TEWS TECHNOLOGIES GmbH

Issue	Description	Date
1.0	First Issue	February 13, 2003
1.1	DEFFS Support	May 19, 2004
1.1.0	General Revision	November 11, 2004
1.2.0	Kernel 2.6 Revision	August 2, 2005
1.3.0	New Address TEWS TECHNOLOGIES LLC. General Revision	December 14, 2006
1.3.1	Address TEWS LLC removed, description for IPAC Carrier Driver added, file-list updated	May 25, 2010

Table of Contents

1	INTRODUCTION.....	4
	1.1 Device Driver	4
	1.2 IPAC Carrier Driver	5
2	INSTALLATION.....	6
	2.1 Build and install the device driver.....	6
	2.2 Uninstall the device driver	7
	2.3 Install device driver into the running kernel	7
	2.4 Remove device driver from the running kernel	8
	2.5 Change Major Device Number	8
	2.6 Number of supported devices	8
3	DEVICE DRIVER PROGRAMMING	9

1 Introduction

1.1 Device Driver

The TIP865-SW-82 Linux device driver allows the operation of a TIP865 IPAC module on Linux operating systems.

The TIP865-SW-82 device driver is based on standard Linux serial device drivers and supports all standard terminal functions (TERMIOS).

Because the TIP865 device driver is stacked on the TEWS TECHNOLOGIES IPAC carrier driver, it is necessary to install also the appropriate IPAC carrier driver. Please refer to the IPAC carrier driver user manual for further information.

The TIP865-SW-82 device driver supports the following features:

- Baud rates up to 230400 baud
- Direct support of different physical interfaces (RS-232, TTL, RS-422, RS-485)
- Creates *TTY* devices *ttyST865_...* dynamically allocated or fixed major device numbers
- Creates dialout devices *cuaT865* with dynamically allocated or fixed major device numbers (*only for kernel 2.4.x*)
- TEWS TECHNOLOGIES IPAC carrier driver support

The TIP865-SW-82 device driver supports the modules listed below:

TIP865-10	4 channel RS232 serial I/O	IndustryPack® compatible
TIP865-11	4 channel TTL serial I/O	IndustryPack® compatible
TIP865-20	4 channel RS422 serial I/O	IndustryPack® compatible
TIP865-30	4 channel RS485 serial I/O	IndustryPack® compatible
TIP865-51	2 channel RS485 serial I/O 2 channel RS422 serial I/O	IndustryPack® compatible

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

TIP865 User manual
TIP865 Engineering Manual
CARRIER-SW-82 IPAC Carrier User Manual

1.2 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 so on. Also the different byte ordering (big-endian versus little-endian) of CPU boards will cause problems on accessing the IndustryPack I/O and memory spaces.

To simplify the implementation of IPAC device drivers which work with any 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-82 is part of this TIP865-SW-82 distribution. It is located in directory CARRIER-SW-82 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-82 User Manual for a detailed description how to install and setup the CARRIER-SW-82 device driver, and for a description of the TEWS TECHNOLOGIES IPAC Carrier Driver concept.

2 Installation

The directory TIP865-SW-82 on the distribution media contains the following files:

TIP865-SW-82-1.3.1.pdf	This manual in PDF format
TIP865-SW-82-SRC.tar.gz	GZIP compressed archive with driver source code
ChangeLog.txt	Release history
Release.txt	Release information

The GZIP compressed archive TIP865-SW-82-SRC.tar.gz contains the following files and directories:

Directory path '/tip865/':

tip865.c	Driver source code
tip865def.h	Driver include file
makenode	Script to create device nodes on the file system
Makefile	Device driver make file
example/tip865exa.c	Example application
example/Makefile	Example application make file
include/config.h	Driver independent library header file
include/tpmodule.h	Kernel independent library header file
include/tpmodule.c	Kernel independent library source code file

In order to perform an installation, extract all files of the archive TIP865-SW-82-SRC.tar.gz to the desired target directory. The command 'tar -xzf TIP865-SW-82-SRC.tar.gz' will extract the files into the local directory.

- Login as *root* and change to the target directory

Before building a new device driver, the TEWS TECHNOLOGIES IPAC carrier driver must be installed properly, because this driver includes the header file `ipac_carrier.h`, which is part of the IPAC carrier driver distribution. Please refer to the IPAC carrier driver user manual in the directory path CARRIER-SW-82 on the separate distribution media.

2.1 Build and install the device driver

- Login as *root*
- Change to the target directory
- To create and install the driver in the module directory `/lib/modules/<version>/misc` enter:

make install

For Linux kernel 2.6.x, there may be compiler warnings claiming some undefined `ipac_*` symbols. These warnings are caused by the IPAC carrier driver, which is unknown during compilation of this TIP driver. The warnings can be ignored.

- Also after the first build we have to execute *depmod* to create a new dependency description for loadable kernel modules. This dependency file is later used by *modprobe* to automatically load the correct IPAC carrier driver modules.

```
# depmod -aq
```

2.2 Uninstall the device driver

- Login as *root*
- Change to the target directory
- To remove the driver from the module directory */lib/modules/<version>/misc* enter:

```
# make uninstall
```

- Update kernel module dependency description file

```
# depmod -aq
```

2.3 Install device driver into the running kernel

- To load the device driver into the running kernel, login as root and execute the following commands:

```
# modprobe tip865drv
```

- After the first build or if you are using dynamic major device allocation it's necessary to create new device nodes on the file system. Please execute the script file *makenode* to do this. If your kernel has enabled a device file system (*devfs* or *sysfs* with *udev*) then you have to skip running the *makenode* script. Instead of creating device nodes from the script the driver itself takes creating and destroying of device nodes in its responsibility.

```
# sh makenode
```

On success the device driver will create four minor devices for each TIP865 module found. The first TIP865 (channel 1...4) can be accessed through device nodes */dev/ttyST865_0.../dev/ttyST865_3*, the second TIP865 (channel 1...4) through device nodes */dev/ttyST865_4.../dev/ttyST865_7* and so on.

The allocation of device nodes to physical TIP865 modules depends on the search order of the IPAC carrier driver. Please refer to the IPAC carrier user manual.

Loading of the TIP865 device driver will only work if kernel KMOD support is installed, necessary carrier board drivers are already installed and the kernel dependency file is up to date. If KMOD support isn't available you have to build either a new kernel with KMOD installed or you have to install the IPAC carrier kernel modules manually in the correct order (please refer to the IPAC carrier driver user manual).

2.4 Remove device driver from the running kernel

- To remove the device driver from the running kernel login as root and execute the following command:

```
# modprobe -r tip865drv
```

If your kernel has enabled a device file system, all TIP865 device nodes will be automatically removed from your file system after this.

Be sure that the driver isn't opened by any application program. If opened you will get the response "*tip865drv: Device or resource busy*" and the driver will still remain in the system until you close all opened files and execute *modprobe -r* again.

2.5 Change Major Device Number

The TIP865 driver use dynamic allocation of major device numbers by default. If this isn't suitable for the application it's possible to define a major number for the driver.

To change a certain major number edit the file tip865.c, change the following symbol to an appropriate value and enter **make install** to create a new driver.

TIP865_TTY_MAJOR	Defines the value for the terminal device. Valid numbers are in range between 0 and 255. A value of 0 means dynamic number allocation.
TIP865_CUA_MAJOR	Defines the value for the dialout device. Valid numbers are in range between 0 and 255. A value of 0 means dynamic number allocation. <i>(Only used for Kernel 2.4.x)</i>

Example:

```
#define TIP865_TTY_MAJOR          122
#define TIP865_CUA_MAJOR          123
```

Be sure that the desired major number isn't used by other drivers. Please check `/proc/devices` to see which numbers are free.

Keep in mind that it is necessary to create new device nodes if the major number for the TIP865 driver has changed.

2.6 Number of supported devices

By default the TIP865 device driver supports up to 8 minor devices (serial channels). If this isn't enough the number of supported TIP865 modules respective the number of minor devices can be increased by modifying the macro **NUM_TIP865** in file tip865.c.

3 Device Driver Programming

The TIP865-SW-82 driver is based on the standard Linux terminal driver. Due to this way of implementation the driver interface and functionality is compatible to the standard Linux terminal driver.

Please refer to the TERMIOS man page and driver programming related man pages for more information about serial driver programming.