The Embedded I/O Company

TEWS TECHNOLOGIES

TIP866-SW-82

Linux Device Driver

8 Channels Serial IPAC

Version 1.3.x

User Manual

Issue 1.3.1

July 2007

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

TEWS TECHNOLOGIES LLC
9190 Double Diamond Parkway,
Suite 127, Reno, NV 89521, USA
www.tews.com
Phone: +1 (775) 850 5830
Fax: +1 (775) 201 0347
e-mail: usasales@tews.com

TIP866-SW-82

Linux Device Driver

8 Channel Serial IPAC

Issue	Description	Date
1.0	First Issue	November 18, 2003
1.1	DEVFS Support	May 28, 2004
1.2.0	Kernel 2.6.x Support	January 13, 2006
1.3.0	Modified Example Applications and UDEV Support	July 11, 2006
1.3.1	New Address TEWS LLC, General Revision	July 26, 2007

Table of Contents

1 Introduction

The TIP866-SW-82 Linux device driver allows the operation of a TIP866 IPAC module on Linux operating systems.

The TIP866-SW-82 device driver is based on standard Linux serial device drivers and supports all standard terminal functions (TERMIOS).

Because the TIP866 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 TIP866 device driver includes the following features:

- Baud rates up to 115200 baud for TIP866-10 and up to 460800 baud for TIP866-11 and TIP866-12.
- Each channel has a 64 Byte transmit and receive FIFO.
- Programmable trigger level for transmit and receive FIFO.
- Hardware (RTS/CTS) and software flow control (XON/XOFF) direct controlled by the serial controller. The advantage of this feature is that the transmission of characters will immediately stop as soon as a complete character is transmitted and not when the transmit FIFO is empty for handshake under software control. This will greatly improve flow control reliability.
- Direct support of different physical interfaces (RS-232, TTL, RS-422,).
- Designed as Linux kernel module with dynamically loading.
- Supports shared IRQ's.
- Creates a *TTY* device *ttySTIP866_...* and dialout device *cuaTIP866_...* with dynamically allocated or fixed major device numbers. (CUA devices for kernel 2.4.x only)
- Support for DEVFS and UDEV
- TEWS TECHNOLOGIES IPAC carrier driver support

The TIP866-SW-82 supports the modules listed below:

TIP866-10	8 channel RS232 serial I/O	IndustryPack® compatible
TIP866-11	8 channel TTL serial I/O	IndustryPack® compatible
TIP866-20	8 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.

TIP866 User manual

TIP866 Engineering Manual

CARRIER-SW-82 IPAC Carrier User Manual

2 Installation

The directory TIP866-SW-82 on the distribution media contains the following files:

TIP866-SW-82-1.3.1.pdf	This manual in PDF format
TIP866-SW-82-SRC.tar.gz	GZIP compressed archive with driver source code
ChangeLog.txt	Release history
Release.txt	Release information

The GZIP compressed archive TIP866-SW-82-SRC.tar.gz contains the following files and directories:

tip866/tip866.c	Driver source code
tip866/tip866.h	Driver include file
tip866/makenode	Script to create device nodes on the file system
tip866/makenodeFM24	Creates CUA devices (Kernel 2.4.x only)
tip866/Makefile	Device driver make file
tip866/example/tip866exa.c	Transfer data between two channels (example)
tip866/example/tip866setspeed.c	Set new baud rates (example)
tip866/example/Makefile	Example application make file
tip866/include/tpmodule.h	Kernel independent library header file
tip866/include/tpmodule.c	Kernel independent library source code file
include/config.h	Driver independent library header file

In order to perform an installation, extract all files of the archive TIP866-SW-82-SRC.tar.gz to the desired target directory. The command 'tar -xzvf TIP866-SW-82-SRC.tar.gz' will extract the files into the local 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/<kernel-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/<kernel-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 tip866drv

The following section is only for kernels without a device file system like DEVFS or UDEV enabled. 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.

 # sh makenode

On success the device driver will create eight minor devices for each TIP866 module found. The first TIP866 (channel 1...8) can be accessed through device node /dev/ttySTIP866_0.../dev/ttySTIP866_7, the second TIP866 (channel 1...8) through device nodes /dev/ttySTIP866_8.../dev/ttySTIP866_15 and so on.

The allocation of device nodes to physical TIP866 modules depends on the search order of the IPAC carrier driver. Please refer to the IPAC carrier user manual.

Loading of the TIP866 device driver will only work if kernel KMOD support is installed, necessary carrier board drivers 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 tip866drv -r

Be sure that the driver isn't opened by any application program. If opened you will get the response *´´tip866drv: 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 following section is only for kernels without a device file system like DEVFS or UDEV enabled. The TIP866 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 *TTY* and *CUA* driver.

To change the major number edit the file tip866.c, change the following symbol to appropriate value and enter **make install** to create a new driver.

TIP866_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.

TIP866_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.

Example:

```
#define TIP866_TTY_MAJOR        122
#define TIP866_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's necessary to create new device nodes if the major number for the TIP866 driver has changed.

2.6 Number of supported devices

By default the TIP866 device driver supports up to 32 TIP866. If this isn't enough the number of supported TIP866 modules respective the number of minor devices can be increased by modifying the macro **TIP866_MAX_NUM_MOD** in tip866.h.

2.7 FIFO Configuration

After installation of the TIP866 Device Driver the trigger level for the transmit and receive FIFO are set to their default values.

Default values for TIP866 are:

Receive FIFO	Transmit FIFO
56	8

The configuration of the FIFO trigger level is used for all TIP866 devices in common.

To change the trigger levels edit the file tip866.c, change the following symbols to appropriate values and enter *make install* to create a new driver.

TIP866_RX_TRG_DEF — Defines the trigger level for the receiver FIFO

Valid trigger levels are:
UART_FCR_R_TRIGGER_8
UART_FCR_R_TRIGGER_16
UART_FCR_R_TRIGGER_56
UART_FCR_R_TRIGGER_60

TIP866_TX_TRG_DEF — Defines the trigger level for the transmitter FIFO

Valid trigger levels are:
UART_FCR_T_TRIGGER_8
UART_FCR_T_TRIGGER_16
UART_FCR_T_TRIGGER_32
UART_FCR_T_TRIGGER_56

Please refer to the User Manual of the ST16C654 controller to get more information how to customize suitable FIFO trigger level.

3 Device Driver Programming

The TIP866-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.

4 Diagnostic

If the TIP866 driver does not work properly it is helpful to get some status information from the driver respective kernel.

The Linux /proc file system provides information about kernel, resources, driver, devices and so on. The following screen dumps display information of a correct running TIP866 driver (see also the proc man pages).

```
# cat /proc/tty/driver/tip866serial
TEWS TECHNOLOGIES - TIP866 serial driver - V1.3.1, date 2006-07-26
0: uart:ST16C654 port:0xEC821000 irq:18 tx:0 rx:0
1: uart:ST16C654 port:0xEC821010 irq:18 tx:0 rx:0
2: uart:ST16C654 port:0xEC821020 irq:18 tx:0 rx:0|DSR|CD|RI
3: uart:ST16C654 port:0xEC821030 irq:18 tx:0 rx:0|DSR|CD|RI
4: uart:ST16C654 port:0xEC821040 irq:18 tx:0 rx:0|DSR|CD|RI
5: uart:ST16C654 port:0xEC821050 irq:18 tx:0 rx:0|DSR|CD|RI
6: uart:ST16C654 port:0xEC821060 irq:18 tx:0 rx:0|DSR|CD|RI
7: uart:ST16C654 port:0xEC821070 irq:18 tx:0 rx:0|DSR|CD|RI

# cat /proc/tty/drivers
/dev/tty             /dev/tty        5       0 system:/dev/tty
/dev/console         /dev/console    5       1 system:console
rfcomm               /dev/rfcomm   216 0-255 serial
tip866serial         /dev/ttySTIP866_ 253 0-255 serial
serial               /dev/ttyS       4 64-95 serial

# cat /proc/tews-ip-carrier

TEWS TECHNOLOGIES - IPAC Carrier Class Driver version 1.3.2 (2007-02-26)

Registered IP slots:

[TEWS TECHNOLOGIES - (Compact)PCI IPAC Carrier - Slot 0]
    Plugged Module     Vendor=0xB3, Modul=0x1D
    Installed Driver   TIP866 - 8 Channel Serial IP
    Slot Setup         INT0_EN | INT1_EN | LEVEL_SENS | CLK_8MHZ |
                       Memory Size = 0x0
    Interrupt Vector   System=17, Module=17
    Interrupt Level    INT0=0, INT1=0
    ID Space Addr      Physical=0xec821080, Virtual=0xe08ac080
    IO Space Addr      Physical=0xec821000, Virtual=0xe08f2000
    MEM8 Space Addr    Physical=0xec000000, Virtual=0x00000000
    MEM16 Space Addr   Physical=0xeb000000, Virtual=0x00000000
```

```
[TEWS TECHNOLOGIES - (Compact)PCI IPAC Carrier - Slot 1]
    Plugged Module     EMPTY

Registered Carrier Drivers:
TEWS TECHNOLOGIES - (Compact)PCI IPAC Carrier V1.3.2
```