

TPMC861-SW-72

LynxOS Device Driver

4-Channel Isolated Serial RS422/RS485 PMC

Version 1.2.x

User Manual

Issue 1.2.0

July 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

TPMC861-SW-72

LynxOS Device Driver

4-Channel Isolated Serial RS422/RS485 PMC

Supported Modules:
TPMC861

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 | April 24, 2003 |
| 1.1.0 | General Revision, New Address TEWS LLC | May 18, 2009 |
| 1.2.0 | LynxOS 5.0.0 support, address footer changed | July 15, 2009 |

Table of Contents

| | | |
|------------|---|----------|
| 1 | INTRODUCTION..... | 4 |
| 2 | INSTALLATION..... | 5 |
| 2.1 | Device Driver Installation | 6 |
| 2.1.1 | Static Installation | 6 |
| 2.1.1.1 | Build the driver object | 6 |
| 2.1.1.2 | Create Device Information Declaration | 6 |
| 2.1.1.3 | Modify the Device and Driver Configuration File | 6 |
| 2.1.1.4 | Rebuild the Kernel | 7 |
| 2.1.2 | Dynamic Installation | 8 |
| 2.1.2.1 | Build the driver object | 8 |
| 2.1.2.2 | Create Device Information Declaration | 8 |
| 2.1.2.3 | Uninstall dynamic loaded driver | 8 |
| 2.1.3 | Device Information Definition File | 9 |
| 2.1.4 | Configuration File: CONFIG.TBL | 11 |

1 Introduction

The TPMC861-SW-72 LynxOS device driver is a full-duplex tty device driver which allows the operation of a TPMC861 serial PMC on a PowerPC platform with DRM based PCI interface.

The TPMC861-SW-72 device driver is based on the standard LynxOS terminal manager. Due to this way of implementation the driver interface and function is absolute compatible to the standard LynxOS terminal driver.

All standard utility programs for configuration and maintaining terminal interfaces could be used in the same manner. There is only one exception; the TPMC861-SW-72 can't be used as console driver.

The TPMC861-SW-72 device driver supports the following features:

- Extended baud rates up to 460.8 kBaud.
- Each channel has a 128 Byte transmit and receive FIFO
- Programmable trigger level for transmit and receive FIFO.
- Software handshake (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.

The TPMC861-SW-72 device driver supports the modules listed below:

| | | |
|---------|--|-------|
| TPMC861 | 4 channel isolated serial interface RS422/RS485 | (PMC) |
|---------|--|-------|

To get more information about the features and use of TPMC861 devices it is recommended to read the manuals listed below.

- TPMC861 User manual
- TPMC861 Engineering Manual

2 Installation

Following files are located on the distribution media:

Directory path 'TPMC861-SW-72':

| | |
|--------------------------|---|
| TPMC861-SW-72-SRC.tar.gz | GZIP compressed archive with driver source code |
| TPMC861-SW-72-1.2.0.pdf | PDF copy of this manual |
| ChangeLog.txt | Release history |
| Release.txt | Release information |

For installation the files have to be copied to the desired target directory.

The GZIP compressed archive TPMC861-SW-72-SRC.tar.gz contains the following files and directories:

Directory path 'tpmc861':

| | |
|----------------------|--------------------------------------|
| tpmc861.c | TPMC861 device driver source |
| tpmc861_info.c | Device information definition |
| tpmc861_info.h | Device information definition header |
| tpmc861.cfg | Driver configuration file include |
| tpmc861.import | Linker import file |
| Makefile | Device driver make file |
| example/tpmc861exa.c | Example application source |
| example/Makefile | Example application Makefile |

In order to perform an installation, extract all files of the archive TPMC861-SW-72-SRC.tar.gz to the desired target directory and copy the files into the system tree as described below.

- (1) Extract driver files:
`tar -xzf TPMC861-SW-82-SRC.tar.gz tpmc861/`
- (2) Create a new directory in the system drivers directory path `/sys/drivers.xxx`, where `xxx` represents the BSP that supports the target hardware.
For example: `/sys/drivers.pp_drm/tpmc861` or `/sys/drivers.cpci_x86/tpmc861`
- (3) Copy the following files to this directory:
`tpmc861.c, tpmc861.import, Makefile`
- (4) Copy `tpmc861.h` to `/usr/include/`
- (5) Copy `tpmc861_info.c` to `/sys/devices.xxx/` or `/sys/devices` if `/sys/devices.xxx` does not exist (`xxx` represents the BSP).
- (6) Copy `tpmc861_info.h` to `/sys/dheaders/`
- (7) Copy `tpmc861.cfg` to `/sys/cfg.xxx/`, where `xxx` represents the BSP for the target platform. For example: `/sys/cfg.ppc` or `/sys/cfg.x86 ...`

2.1 Device Driver Installation

The two methods of driver installation are as follows:

- Static Installation
- Dynamic Installation

2.1.1 Static Installation

With this method, the driver object code is linked with the kernel routines and is installed during system start-up.

2.1.1.1 Build the driver object

1. Change to the directory `/sys/drivers.xxx/tpmc861`, where xxx represents the BSP that supports the target hardware.
2. To update the library `/sys/lib/libdrivers.a` enter: `make install`

2.1.1.2 Create Device Information Declaration

1. Change to the directory `/sys/devices.xxx` or `/sys/devices` if `/sys/devices.xxx` does not exist (xxx represents the BSP).
2. Add the following dependencies to the Makefile
`DEVICE_FILES_prep = ...tpmc861_info.x`

And at the end of the Makefile

```
...
tpmc861_info.o:$(DHEADERS)/tpmc861_info.h
```

3. To update the library `/sys/lib/libdevices.a` enter: `make install`

2.1.1.3 Modify the Device and Driver Configuration File

In order to insert the driver object code into the kernel image, an appropriate entry in file `CONFIG.TBL` must be created.

1. Change to the directory `/sys/lynx.os/` respective `/sys/bsp.xxx`, where xxx represents the BSP that supports the target hardware.
2. Create an entry in the file `CONFIG.TBL`
Insert the entry after the console driver section

```
# End of console devices
I:tpmc861.cfg
```

2.1.1.4 Rebuild the Kernel

1. Change to the directory `/sys/lynx.os/ (/sys/bsp.xxx)`
2. To rebuild the kernel enter the following command: `make install`
3. Reboot the newly-created operating system by the following command:

```
reboot -aN
```

The N flag instructs init to run `mknod` and create all the nodes mentioned in the new `nodetab`.

4. After reboot you should find the following new devices (depends on the device configuration):
`/dev/tp861a1 ... /dev/tp861a4`

2.1.2 Dynamic Installation

This method allows you to install the driver after the operating system is booted. The driver object code is attached to the end of the kernel image and the operating system dynamically adds this driver to its internal structures. The driver can also be removed dynamically.

2.1.2.1 Build the driver object

- (1) Change to the directory `/sys/drivers.xxx/tpmc861`, where `xxx` represents the BSP that supports the target hardware.
- (2) To make the dynamic link-able driver enter:
`make dldd`

2.1.2.2 Create Device Information Declaration

- (1) Change to the directory `/sys/drivers.xxx/tpmc861`, where `xxx` represents the BSP that supports the target hardware.
- (2) To create a device definition file for the major device (this works only on native system)
`make t861info`
- (3) To install the driver enter:
`drinstall -c tpmc861.obj`
If successful, `drinstall` returns a unique `<driver-ID>`
- (4) To install the major device enter:
`devinstall -c -d <driver-ID> t002info`
The `<driver-ID>` is returned by the `drinstall` command
- (5) To create nodes for the devices enter:
`mknod /dev/tp861a1 c <major_no> 0`
`mknod /dev/tp861a2 c <major_no> 1`
`mknod /dev/tp861a3 c <major_no> 2`
...
The `<major_no>` is returned by the `devinstall` command.

If all steps are successful completed the TPMC861 is ready to use.

2.1.2.3 Uninstall dynamic loaded driver

To uninstall the TPMC861 device enter the following commands:

```
devinstall -u -c <device-ID>
drinstall -u <driver-ID>
```

2.1.3 Device Information Definition File

The device information definition contains information necessary to install the TPMC861 major device.

The implementation of the device information definition is done through a C structure which is defined in the header file `tpmc861_info.h`.

This structure contains following parameters:

| | |
|------------------------|--|
| PCIBusNumber | Contains the PCI bus number at which the TPMC861 compatible device is connected. Valid bus numbers are in range from 0 to 255. |
| PCIDeviceNumber | Contains the device number (slot) at which the TPMC861 compatible device is connected. Valid device numbers are in range from 0 to 31. |

If both PCIBusNumber and PCIDeviceNumber are -1 then the driver will auto scan for the TPMC861 device. The first device found in the scan order will be allocated by the driver for this major device.

Already allocated devices can't be allocated twice. This is important to know if there are more than one TPMC861 major devices.

| | |
|------------------------|--|
| FIFO_Settings[] | Contains the trigger level for receive and transmit FIFO. And the transceiver startup configuration for programmable transceivers. Valid settings for receive and transmit FIFO are between 1 and 127. A trigger level of 1 will disable the receive or transmit FIFO. |
| sg | This structure contains initial tty parameter like baud rate, special characters and so on. Refer also to the tty man pages. |

A device information definition is unique for every TPMC861 major device. The file `tpmc861_info.c` on the distribution disk contains two device information declarations, `tp861A` for the first major device and `tp861B` for the second major device.

If the driver should support more than two major devices it is necessary to copy and paste an existing declaration and rename it with unique name for example **tp861C**, **tp861D** and so on.

It is also necessary to modify the device and driver configuration file, respectively the configuration include file `tpmc861.cfg`.

The following device declaration information uses the auto find method to detect the TPMC861 module on PCI bus.

The Rx trigger level is set to 96 Byte and the Tx trigger level is set to 16 for all minor devices.

```

TP861_INFO tp861A = {
    -1,                /* auto find the TPMC861 on any PCI bus */
    -1,

    { /* Rx, Tx trigger level */
        { 96, 16 },    /* channel 0 ( minor device 0 ) */
        { 96, 16 },    /* channel 1 ( minor device 1 ) */
        { 96, 16 },    /* channel 2 ( minor device 2 ) */
        { 96, 16 },    /* channel 3 ( minor device 3 ) */
    },

    {
        B9600, B9600,    /* input and output speed */
        'H' - '@',      /* erase char */
        -1,              /* 2nd erase char */
        'U' - '@',      /* kill char */
        ECHO | CRMOD,    /* mode */
        'C' - '@',      /* interrupt character */
        '\\\' - '@',    /* quit char */
        'Q' - '@',      /* start char */
        'S' - '@',      /* stop char */
        'D' - '@',      /* EOF */
        -1,              /* brk */
        (LCRTBS | LCRTERA | LCRTKIL | LCTLECH), /* local mode word */
        'Z' - '@',      /* process stop */
        'Y' - '@',      /* delayed stop */
        'R' - '@',      /* reprint line */
        'O' - '@',      /* flush output */
        'W' - '@',      /* word erase */
        'V' - '@'       /* literal next char */
    }
};

```

2.1.4 Configuration File: CONFIG.TBL

The device and driver configuration file CONFIG.TBL contains entries for device drivers and its major and minor device declarations. Each time the system is rebuilt, the config utility read this file and produces a new set of driver and device configuration tables and a corresponding nodetab.

To install the TPMC861 driver and devices into the LynxOS system, the configuration include file tpmc861.cfg must be included in the CONFIG.TBL (see also 2.1.1.3).

The file tpmc861.cfg on the distribution disk contains the driver entry (*C:tpmc861:\...*) and one enabled major device entry (*D:TPMC861 1-4:tp861A::*) with 4 minor device entries (*N: tp861a1:0 ... N:tp861a4:0*).

If the driver should support more than one major device the following entries for major and minor devices must be enabled by removing the comment character (#). By copy and paste an existing major and minor entry and renaming the new entries, it is possible to add any number of additional TPMC861 devices.

This example shows a driver entry with one major device and 4 minor devices:

```
#   Format:
#   C:driver-name:open:close:read:write:select:control:install:uninstall
#   D:device-name:info-block-name:raw-partner-name
#   N:node-name:minor-dev
C:tpmc861:\
    :tp861open:tp861close:tp861read:tp861write:\
    :tp861select:tp861ioctl:tp861install:tp861uninstall
D:TPMC861 1-8:tp861A::
N:tp861a1:0
N:tp861a2:1
N:tp861a3:2
N:tp861a4:3
```

The configuration above creates the following nodes in the /dev directory.

```
/dev/tp861a1
/dev/tp861a2
/dev/tp861a3
/dev/tp861a4
```