TPMC700-SW-65

Windows Device Driver

32(16) Digital Output PMC

Version 2.0.x

User Manual

Issue 2.0.0

October 2012

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

TPMC700-SW-65

Windows Device Driver

32(16) Digital Output PMC

Supported Modules:
TPMC700

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.

Issue	Description	Date
1.0	First Issue	August 28, 2003
1.0.1	File list changed, general revision	January 25, 2010
2.0.0	Support for Windows 7, API documented	October 29, 2012

Table of Contents

1 INTRODUCTION 4

2 INSTALLATION 5

2.1 Software Installation 5

2.1.1 Windows 2000 5

2.1.2 Windows 7 / XP 6

2.2 Confirming Driver Installation 6

3 API DOCUMENTATION 7

3.1 General Functions 7

3.1.1 tpmc700Open 7

3.1.2 tpmc700Close 9

3.1.3 tpmc700GetPciInfo 11

3.2 Output Functions 14

3.2.1 tpmc700Write 14

3.2.2 tpmc700WriteMask 16

3.2.3 tpmc700OutputLineSet 18

3.2.4 tpmc700OutputLineClear 20

3.2.5 tpmc700OutputStatus 22

3.3 Watchdog Functions 24

3.3.1 tpmc700WatchdogEnable 24

3.3.2 tpmc700WatchdogDisable 26

3.3.3 tpmc700WatchdogReset 28

1 Introduction

The TPMC700-SW-65 Windows device driver is a kernel mode driver which allows the operation of supported hardware modules on an Intel or Intel-compatible Windows operating system. Supported Windows versions are:

- Windows 2000
- Windows XP
- Windows XP Embedded
- Windows 7 (32bit and 64bit)

The TPMC700-SW-65 device driver supports the following features:

- Set Output Lines
- Start and Stop Output Watchdog
- Reset Watchdog Error Flag

The TPMC700-SW-65 supports the modules listed below:

TPMC700-x0	32 Digital Outputs	(PMC)
TPMC700-x1	16 Digital Outputs	(PMC)

To get more information about the features and use of TPMC700 devices it is recommended to read the manuals listed below.

TPMC700 User Manual

TPMC700 Engineering Manual

2 Installation

Following files are located in directory TPMC700-SW-65 on the distribution media:

i386\	Directory containing driver files for 32bit Windows versions
amd64\	Directory containing driver files for 64bit Windows versions
installer_32bit.exe	Installation tool for 32bit systems (Windows XP or later)
installer_64bit.exe	Installation tool for 64bit systems (Windows XP or later)
tpmc700.inf	Windows installation script
tpmc700.h	Header file with IOCTL codes and structure definitions
example\tpmc700exa.c	Example application
api\tpmc700api.c	Application Programming Interface source
api\tpmc700api.h	Application Programming Interface header
TPMC700-SW-65-2.0.0.pdf	This document
Release.txt	Information about the Device Driver Release
ChangeLog.txt	Release history

2.1 Software Installation

2.1.1 Windows 2000

This section describes how to install the TPMC700-SW-65 Device Driver on a Windows 2000 operating system.

After installing the TPMC700 card(s) and boot-up your system, Windows 2000 setup will show a "***New hardware found***" dialog box.

1. The "***Upgrade Device Driver Wizard***" dialog box will appear on your screen.
   Click "***Next***" button to continue.

2. In the following dialog box, choose "***Search for a suitable driver for my device***".
   Click "***Next***" button to continue.

3. Insert the TPMC700 driver media; select "***Disk Drive***" in the dialog box.
   Click "***Next***" button to continue.

4. Now the driver wizard should find a suitable device driver on the media.
   Click "***Next***" button to continue.

5. Complete the upgrade device driver and click "***Finish***" to take all the changes effect.

6. Now copy all needed files (tpmc700.h and API files) to the desired target directories.

After successful installation the TPMC700 device driver will start immediately and creates devices (TPMC700_1, TPMC700_2 ...) for all recognized TPMC700 modules.

2.1.2 Windows 7 / XP

This section describes how to install the TPMC700-SW-65 Device Driver on a Windows 7 (32bit or 64bit) or Windows XP (32-bit) operating system.

Depending on the operating system type, execute the installer binaries for either 32bit or 64bit systems. This will install all required driver files using an installation wizard.

Copy needed files (tpmc700.h and API files) to desired target directory.

After successful installation a device is created for each module found (TPMC700_1, TPMC700_2 ...).

2.2 Confirming Driver Installation

To confirm that the driver has been properly loaded, perform the following steps:

1. Open the Windows Device Manager:

    a. For Windows 2000 / XP, open the "***Control Panel***" from "***My Computer***" and click the "***System***" icon and choose the "***Hardware***" tab, and then click the "***Device Manager***" button.

    b. For Windows 7, open the "***Control Panel***" from "***My Computer***" and then click the "***Device Manager***" entry.

2. Click the "**+**" in front of "***Embedded I/O***".
The driver "***TEWS TECHNOLOGIES – TPMC700 (32/16 Digital Output)***" should appear for each installed device.

3 API Documentation

3.1 General Functions

3.1.1 tpmc700Open

NAME

tpmc700Open – opens a device.

SYNOPSIS

```
TPMC700_HANDLE tpmc700Open
(
        char            *DeviceName
)
```

DESCRIPTION

Before I/O can be performed to a device, a device descriptor must be opened by a call to this function.

PARAMETERS

DeviceName

This parameter points to a null-terminated string that specifies the name of the device. . The first TPMC700 device is named “\\\\.\\TPMC700_1” the second device is named “\\\\.\\TPMC700_2” and so on.

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE     hdl;

/*
** open the specified device
*/
hdl = tpmc700Open(“\\\\.\\TPMC700_1”, 0);
if (hdl == NULL)
{
     /* handle open error */
}
```

RETURNS

A device handle, or NULL if the function fails. An error code will be stored in *errno*.

ERROR CODES

The error codes are stored in *errno.*

The error code is a standard error code set by the I/O system.

3.1.2 tpmc700Close

NAME

tpmc700Close – closes a device.

SYNOPSIS

```
TPMC700_STATUS tpmc700Close
(
    TPMC700_HANDLE      hdl
)
```

DESCRIPTION

This function closes previously opened devices.

PARAMETERS

hdl

This value specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE     hdl;
TPMC700_STATUS     result;

/*
** close the device
*/
result = tpmc700Close(hdl);
if (result != TPMC700_OK)
{
     /* handle close error */
}
```

RETURNS

On success, TPMC700_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TPMC700_ERR_INVALID_HANDLE	The specified device handle is invalid

3.1.3 tpmc700GetPciInfo

NAME

tpmc700GetPciInfo – get PCI information of the module

SYNOPSIS

```
TPMC700_STATUS tpmc700GetPciInfo
(
        TPMC700_HANDLE              hdl,
        TPMC700_PCIINFO_BUF         *pPciInfoBuf
)
```

DESCRIPTION

This function returns information about the module's PCI header as well as the PCI localization.

PARAMETERS

hdl

This argument specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

pPciInfoBuf

This argument is a pointer to the structure TPMC700_PCIINFO_BUF that receives information of the module PCI header.

```
typedef struct
{
        unsigned short      vendorId;
        unsigned short      deviceId;
        unsigned short      subSystemId;
        unsigned short      subSystemVendorId;
        int                 pciBusNo;
        int                 pciDevNo;
        int                 pciFuncNo;
} TPMC700_PCIINFO_BUF;
```

vendorId

PCI module vendor ID.

deviceId

PCI module device ID

subSystemId

PCI module sub system ID

subSystemVendorId

PCI module sub system vendor ID

pciBusNo

Number of the PCI bus, where the module resides.

pciDevNo

PCI device number

pciFuncNo

PCI function number

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE          hdl;
TPMC700_STATUS          result;
TPMC700_PCIINFO_BUF     pciInfoBuf

/*
** get module PCI information
*/
result = tpmc700GetPciInfo( hdl, &pciInfoBuf );

if (result == TPMC700_OK)
{
     printf( “PCI Localization (Bus:Dev.Func): %d:%d.%d\n”,
                    pciInfoBuf.pciBusNo,
                    pciInfoBuf.pciDevNo,
                    pciInfoBuf.pciFuncNo );
} else {
     /* handle error */
}
```

RETURN VALUE

On success, TPMC700_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	**Description**
TPMC700_ERR_INVALID_HANDLE	The specified device handle is invalid
TPMC700_ERR_INVAL	Specified pointer is invalid.

3.2 Output Functions

3.2.1 tpmc700Write

NAME

tpmc700Write – Write Output Value

SYNOPSIS

```
TPMC700_STATUS tpmc700Write
(
        TPMC700_HANDLE          hdl,
        unsigned int            OutputValue
)
```

DESCRIPTION

This function writes the specified output value (32bit) to the specific module.

PARAMETERS

hdl

This value specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

OutputValue

This argument specifies the new output value. Bit 0 of the output word corresponds to the first output line, bit 1 corresponds to the second output line, and so on.

Bit 16 up to 32 will be ignored for TPMC700-x1 (16 output lines).

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE     hdl;
TPMC700_STATUS     result;

/*------------------------------
  Set output lines to 0x12345678
  ------------------------------*/
result = tpmc700Write(  hdl,
                        0x12345678 );
if (result != TPMC700_OK)
{
     /* handle error */
}
```

RETURN VALUE

On success, TPMC700_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TPMC700_ERR_INVALID_HANDLE	The specified device handle is invalid.
TPMC700_ERR_TIMEOUT	Watchdog Timeout error occurred.

3.2.2 tpmc700WriteMask

NAME

tpmc700WriteMask – Write Output Value with Bitmask

SYNOPSIS

```
TPMC700_STATUS tpmc700WriteMask
(
      TPMC700_HANDLE            hdl,
      unsigned int              OutputValue,
      unsigned int              Mask
)
```

DESCRIPTION

This function sets the output lines to the specified value. Only output lines specified by the bitmask are affected.

PARAMETERS

hdl

This value specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

OutputValue

This argument specifies the new output value. Bit 0 of the output word corresponds to the first output line, bit 1 corresponds to the second output line, and so on.

Bit 16 up to 32 will be ignored for TPMC700-x1 (16 output lines).

Mask

This parameter specifies a 32bit mask. '1' means that the corresponding bit in *OutputValue* will be updated. '0' bits will be left unchanged. Bit 0 corresponds to the first output line, bit 1 corresponds to the second output line and so on.

Bit 16 up to 32 will be ignored for TPMC700-x1 (16 output lines).

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE     hdl;
TPMC700_STATUS     result;
unsigned int       OutputValue;
unsigned int       Mask;

/*-------------------------------------------------
  Set output line 1 and 8 (bit 0 and bit 7), and
  clear output line 32 (bit 31)
  -------------------------------------------------*/
OutputValue    = (1 <<  7) | (1 << 0);
Mask           = (1 << 31) | (1 << 7) | (1 << 0);

result = tpmc700WriteMask(  hdl,
                            OutputValue,
                            Mask );
if (result != TPMC700_OK)
{
     /* handle error */
}
```

RETURN VALUE

On success, TPMC700_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TPMC700_ERR_INVALID_HANDLE	The specified device handle is invalid.
TPMC700_ERR_TIMEOUT	Watchdog Timeout error occurred.

3.2.3 tpmc700OutputLineSet

NAME

tpmc700OutputLineSet – Set the specific Output Line

SYNOPSIS

```
TPMC700_STATUS tpmc700OutputLineSet
(
        TPMC700_HANDLE              hdl,
        int                         OutputLine
)
```

DESCRIPTION

This function sets the specified output line to ‘1’.

PARAMETERS

hdl

This value specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

OutputLine

This argument specifies the output line number which shall be set. Valid values are 1 to 32. For TPMC700-x1 modules, values higher than 16 are ignored.

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE     hdl;
TPMC700_STATUS     result;

/*-------------------------------------------------
  Set output line 4
  -------------------------------------------------*/

result = tpmc700OutputLineSet( hdl, 4 );
if (result != TPMC700_OK)
{
     /* handle error */
}
```

RETURN VALUE

On success, TPMC700_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	**Description**
TPMC700_ERR_INVALID_HANDLE	The specified device handle is invalid.
TPMC700_ERR_TIMEOUT	Watchdog Timeout error occurred.
TPMC700_ERR_INVAL	Specified output line is invalid.

3.2.4 tpmc700OutputLineClear

NAME

tpmc700OutputLineClear – Clear the specific Output Line

SYNOPSIS

```
TPMC700_STATUS tpmc700OutputLineClear
(
    TPMC700_HANDLE          hdl,
    int                     OutputLine
)
```

DESCRIPTION

This function clears the specified output line to ‘0’.

PARAMETERS

hdl

This value specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

OutputLine

This argument specifies the output line number which shall be cleared. Valid values are 1 to 32. For TPMC700-x1 modules, values higher than 16 are ignored.

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE     hdl;
TPMC700_STATUS     result;

/*-------------------------------------------------
  Clear output line 32
  -------------------------------------------------*/

result = tpmc700OutputLineClear( hdl, 32 );
if (result != TPMC700_OK)
{
     /* handle error */
}
```

RETURN VALUE

On success, TPMC700_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TPMC700_ERR_INVALID_HANDLE	The specified device handle is invalid.
TPMC700_ERR_TIMEOUT	Watchdog Timeout error occurred.
TPMC700_ERR_INVAL	Specified output line is invalid.

3.2.5 tpmc700OutputStatus

NAME

tpmc700OutputStatus – Read Status of Output Lines and Watchdog

SYNOPSIS

```
TPMC700_STATUS tpmc700OutputStatus
(
      TPMC700_HANDLE             hdl,
      unsigned int               *pOutputValue,
      unsigned int               *pWatchdogStatus
)
```

DESCRIPTION

This function reads the status of the output lines and also the watchdog facility.

PARAMETERS

hdl

This value specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

pOutputValue

This argument is a pointer to an *unsigned int* (32bit) value where the output line status is returned. Bit 0 of the output word corresponds to the first output line, bit 1 corresponds to the second output line, and so on. For TPMC700-x1, the upper 16bits shall be ignored.

pWatchdogStatus

This argument is a pointer to an *unsigned int* (32bit) value where the watchdog status is returned. The following values are possible:

Value	**Description**
TPMC700_WD_ENABLED	The Watchdog is enabled.
TPMC700_WD_DISABLED	The Watchdog is disabled.
TPMC700_WD_FAILURE	The Watchdog has recognized a failure and has disabled all output channels.

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE     hdl;
TPMC700_STATUS     result;
unsigned int       OutputValue;
unsigned int       WatchdogStatus;

/*------------------------------------------------
  Read output status
  ------------------------------------------------*/

result = tpmc700OutputStatus(    hdl,
                                 &OutputValue,
                                 &WatchdogStatus );
if (result == TPMC700_OK)
{
     if (WatchdogStatus != TPMC700_WD_FAILURE)
     {
          printf(“Output Status: 0x%08X\n”, OutputValue);
     } else {
          printf(“Output disabled by Watchdog\n”);
     }
} else {
     /* handle error */
}
```

RETURN VALUE

On success, TPMC700_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TPMC700_ERR_INVALID_HANDLE	The specified device handle is invalid.

3.3 Watchdog Functions

3.3.1 tpmc700WatchdogEnable

NAME

tpmc700WatchdogEnable – Enable Output Watchdog

SYNOPSIS

```
TPMC700_STATUS tpmc700WatchdogEnable
(
    TPMC700_HANDLE          hdl
)
```

DESCRIPTION

This function enables the watchdog timer for the output lines. The watchdog function is activated after the next write operation to the device. Please remember that if the watchdog is enabled and no write access occurs within 120 ms, all outputs go into the OFF state. To unlock the output register and leave the OFF state the function *tpmc700WatchdogReset* must be executed.

PARAMETERS

hdl

This value specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE     hdl;
TPMC700_STATUS     result;

/*-----------------
  Enable Watchdog
  -----------------*/
result = tpmc700WatchdogEnable( hdl );
if (result != TPMC700_OK)
{
     /* handle error */
}
```

RETURN VALUE

On success, TPMC700_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TPMC700_ERR_INVALID_HANDLE	The specified device handle is invalid.

3.3.2 tpmc700WatchdogDisable

NAME

tpmc700WatchdogDisable – Disable Output Watchdog

SYNOPSIS

TPMC700_STATUS tpmc700WatchdogDisable
(
    TPMC700_HANDLE   hdl
)

DESCRIPTION

This function disables the watchdog timer for the output lines.

PARAMETERS

hdl

This value specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE     hdl;
TPMC700_STATUS     result;

/*-----------------
  Disable Watchdog
  -----------------*/
result = tpmc700WatchdogDisable( hdl );
if (result != TPMC700_OK)
{
     /* handle error */
}
```

RETURN VALUE

On success, TPMC700_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TPMC700_ERR_INVALID_HANDLE	The specified device handle is invalid.

3.3.3 tpmc700WatchdogReset

NAME

tpmc700WatchdogReset – Reset Output Watchdog Error

SYNOPSIS

```
TPMC700_STATUS tpmc700WatchdogReset
(
    TPMC700_HANDLE          hdl
)
```

DESCRIPTION

This function resets the watchdog status and clears an occurred error.

PARAMETERS

hdl

This value specifies the device handle to the hardware module retrieved by a call to the corresponding open-function.

EXAMPLE

```
#include “tpmc700api.h”

TPMC700_HANDLE     hdl;
TPMC700_STATUS     result;

/*-----------------
  Reset Watchdog
  -----------------*/
result = tpmc700WatchdogReset( hdl );
if (result != TPMC700_OK)
{
     /* handle error */
}
```

RETURN VALUE

On success, TPMC700_OK is returned. In the case of an error, the appropriate error code is returned by the function.

ERROR CODES

Error Code	Description
TPMC700_ERR_INVALID_HANDLE	The specified device handle is invalid.