

TPMC700-SW-42

VxWorks Device Driver

32 (16) Digital Outputs

Version 4.1.x

User Manual

Issue 4.1.0

September 2025

TPMC700-SW-42

VxWorks Device Driver

32 (16) Digital Outputs

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.

©1999-2025 by TEWS Technologies GmbH

Issue	Description	Date
1.0	First Issue	1999
1.1	Intel X86 support added	March 11, 2002
1.2	General Revision	November 28, 2003
1.2.1	Filelist changed	August 11, 2005
2.0.0	Functions tpmc700Drv(), tpmc700DevCreate modified Filelist changed	March 8, 2007
2.0.1	Modified parameter description for write()	February 6, 2009
3.0.0	VxBus support and API added, general revision	September 17, 2010
4.0.0	New implementation of API, support of VxWorks 6.9 (SMP and 64-bit)	April 25, 2016
4.1.0	Address of TEWS Technologies changed, VxWorks 7 support added, new installation guide, Support for Legacy VxWorks (before V 6.9) is EOL, descriptions removed	September 12, 2025

Table of Contents

1	INTRODUCTION.....	4
2	API DOCUMENTATION	5
	2.1 General Functions.....	5
	2.1.1 tpmc700Open.....	5
	2.1.2 tpmc700Close	7
	2.1.3 tpmc700GetPciInfo.....	9
	2.2 Output Functions	12
	2.2.1 tpmc700Write	12
	2.2.2 tpmc700WriteMask	14
	2.2.3 tpmc700OutputLineSet	16
	2.2.4 tpmc700OutputLineClear	18
	2.2.5 tpmc700OutputStatus	20
	2.3 Watchdog Functions.....	22
	2.3.1 tpmc700WatchdogEnable	22
	2.3.2 tpmc700WatchdogDisable	24
	2.3.3 tpmc700WatchdogReset.....	26
3	APPENDIX.....	28
	3.1 Enable RTP-Support	28
	3.2 Debugging and Diagnostic	28

1 Introduction

The TPMC700-SW-42 VxWorks device driver software allows the operation of the supported PMC conforming to the VxWorks I/O system specification.

The TPMC700-SW-42 release contains driver sources for the VxBus-enabled (GEN1 and GEN2) driver model and supports later VxWorks 6.9.x and VxWorks 7 releases, including VxWorks 64-bit and SMP systems.

The driver provides an application programming interface (API) for easy access to all functionalities.

The driver invoke synchronization methods to prevent simultaneous requests by multiple tasks from interfering with each other.

The TPMC700-SW-42 device driver supports the following features:

- Set Output Lines
- Start and Stop Output Watchdog
- Reset Watchdog Error Flag
- Read board information

The TPMC700-SW-42 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
TEWS Technologies VxWorks Device Drivers - Installation Guide

2 API Documentation

2.1 General Functions

2.1.1 tpmc700Open

NAME

tpmc700Open – Opens a device

SYNOPSIS

```
TPMC700_HANDLE tpmc700Open  
(  
    char      *deviceName  
)
```

DESCRIPTION

Before I/O can be performed to a device, a file 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 “/tpmc700/0”, the second device is named “/tpmc700/1” and so on.

EXAMPLE

```
#include "tpmc700api.h"  
  
TPMC700_HANDLE    hdl;  
  
/* open file descriptor to device */  
hdl = tpmc700Open( "/tpmc700/0" );  
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.

2.1.2 tpmc700Close

NAME

tpmc700Close – Closes a device

SYNOPSIS

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

DESCRIPTION

This function closes previously opened devices.

PARAMETERS

hdl

This argument 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 file descriptor to 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

2.1.3 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_INVALID	Specified pointer is invalid.

2.2 Output Functions

2.2.1 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.

2.2.2 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.

2.2.3 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_INVALID	Specified output line is invalid.

2.2.4 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_INVALID	Specified output line is invalid.

2.2.5 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.

2.3 Watchdog Functions

2.3.1 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.

2.3.2 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.

2.3.3 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.

3 Appendix

3.1 Enable RTP-Support

Using TPMC700 devices tunneled from Real Time Processes (RTPs) is implemented. For this the "TEWS TPMC700 IOCTL command validation" must be enabled in system configuration.

The API source file "tpmc700api.c" must be added to the RTP-Project directory and built together with the RTP-application.

The definition of TVXB_RTP_CONTEXT must be added to the project, which is used to eliminate kernel headers, values and functions from the used driver files.

Find more detailed information in "TEWS Technologies VxWorks Device Drivers - Installation Guide".

Debugging functions are not usable from RTPs.

3.2 Debugging and Diagnostic

The TPMC700 device driver provides a function and debug statements to display versatile information of the driver installation and status on the debugging console.

The TPMC700 show routine is included in the VxBus driver by default and can be called from the VxWorks shell. If this function is not needed or program space is rare the function can be removed from the code by un-defining the macro INCLUDE_TPMC700_SHOW in tpmc700drv.c

The tpmc700Show function displays detailed information about probed modules, assignment of devices respective device names to probed TPMC700 modules and device statistics.

If TPMC700 modules were probed but no devices were created it may be helpful to enable debugging code inside the driver code by defining the macro TPMC700_DEBUG in tpmc700drv.c.

```
-> tpmc700Show
Probed Modules:
    [0] TPMC700-10: Bus=4, Dev=2, DevId=0x02bc, VenId=0x1498, Init=OK,
    vxDev=0x0xffff800000140bc0

Associated Devices:
    [0] TPMC700: /tpmc700/0

Device Statistics:
    /tpmc700/0:
        open count           = 0
        Current Output Value = 0x00000000
        Watchdog disabled
```