



TPMC880-SW-42
VxWorks Device Driver
TPMC880 – Ethernet Adapter

Version 1.3

Reference Manual
Issue 1.5

March 2004

TEWS TECHNOLOGIES GmbH
Am Bahnhof 7
D-25469 Halstenbek
Germany
Tel.: +49 (0)4101 4058-0
Fax.: +49 (0)4101 4058-19
<http://www.tews.com>
e-mail: info@tews.com

TPMC880-SW-42

Ethernet Adapter

VxWorks Device Driver

This document contains information, which is proprietary to TEWS TECHNOLOGIES. Any reproduction without written permission is forbidden.

TEWS TECHNOLOGIES has made any effort to ensure that this manual is accurate and complete. However TEWS TECHNOLOGIES reserves the right to change the product described in this document at any time without notice.

This product has been designed to operate with PCI Mezzanine Card (PMC) compatible carriers. Connection to incompatible hardware is likely to cause serious damage.

TEWS TECHNOLOGIES is not liable for any damage arising out of the application or use of the device described herein.

©2001 by TEWS TECHNOLOGIES GmbH

Issue	Description	Date
1.0	First Issue	April 17, 2001
1.1	Installation Changed	November 15, 2001
1.2	BNC/TP Selection Added	April 16, 2002
1.3	Support for MV5100 BSP	August 22, 2003
1.4	Support for MV5100 BSP	August 22, 2003
1.5	Description errors fixed	March 4. 2004

Table of Contents

1	GENERAL INFORMATION.....	4
2	INSTALLATION	5
2.1	Install the driver to the VxWorks-System (1).....	5
2.1.1	Install Driver File.....	5
2.1.2	Modify local BSP	5
2.1.3	Modify global BSP files.....	8
2.1.4	Change Project Configuration	8
2.2	Install the driver to the VxWorks-System (2).....	9
2.2.1	Install Driver File.....	9
2.2.2	Modify local BSP	9
2.2.3	Change Project Configuration	11
2.3	Other BSPs	12
3	STARTING NETWORK NODE	13

1 General Information

The TPMC880-SW-42 VxWorks device driver is based on the standard VxWorks LAN driver dec21X4xEnd.c. Different changes were made to support the medias supported by TPMC880 variants.

The driver should be installed additional to the BSP driver.

2 Installation

The TPMC880-SW-42 device driver software is delivered on a 3½" HD PC diskette.

Following files are located on the diskette:

tp880End.o	Linkable driver object for TPMC880
example*	changed files from BSP (example)
TPMC880-SW-42.pdf	This manual

2.1 Install the driver to the VxWorks-System (1)

This installation shall be used for MVME2300/2600/3600 BSPs.

The following changes will install the driver and enable the TPMC880 on the first PMC-slot of a MVME2300/2600/3600 CPU as a secondary network controller.

2.1.1 Install Driver File

Copy the driver file from the MV2600 path into your project directory and include the file into the project.

The file must be copied into the project path where the objects will be placed. Then you have to add the file into the project. First select the 'builds' tab from the project, and change to your target configuration, open properties using the right click and select the 'MACROS' tab. Select 'EXTRA_MODULES' in 'Macros' and add tp880End.o in 'Value'.

2.1.2 Modify local BSP

The VxWorks documentation describes the dec21x4xEnd driver, which is configuration compatible to the tp880End driver.

Attention

After modifying a file of the BSP, the complete project must be recompiled, if just an update compile is made, the BSP-files will not be compiled!

a) File: \$(BSP)/config.h

i) Enable define to enable a secondary network controller:

(there are some *#undef INCLUDE_SECONDARY_ENET* statements in config.h, the define should be done behind it)

```
/* added for TPMC880 */  
#define          INCLUDE_SECONDARY_ENET /* TPMC880 */
```

ii) Change memory pool parameters for the second 'dc' controller:

(the `DC_POOL_SIZE` and `DC1_POOL_SIZE` should be assigned a value power of 2, and should not exceed 0x10000)

```
#ifndef INCLUDE_PRIMARY_ENET
#define INCLUDE_DC
#define DC_POOL_ADRS      (RAM_LOW_ADRS - DC_POOL_SIZE)
                        /* memory pool adrs */
#define DC_POOL_SIZE     0x00008000
                        /* memory pool size */
#define DC_MODE          0x08
                        /* Dec21140 */
#endif /* INCLUDE_PRIMARY_ENET */

/* Secondary Ethernet = DEC21143 = device unit #1 (auto size) */

#ifndef INCLUDE_SECONDARY_ENET
#define DC1_POOL_ADRS    -1
                        /* memory pool adrs */
#define DC1_POOL_SIZE   -1
                        /* memory pool size */
#define DC1_MODE        0x04
                        /* Dec21143 */
#endif /* INCLUDE_SECONDARY_ENET */
```

b) File: \$(BSP)/mv2600.h

i) Add configuration values for secondary network controller:

```
/* Dec 21140 (unit 0) vector and level */
#define INT_VEC_DC      LN_INT_VEC
                        /* interrupt vector */
#define INT_LVL_DC     LN_INT_LVL
                        /* interrupt level */
#define DC_DATA_WIDTH  NONE
                        /* all data widths */
#define IO_ADRS_DC     LAN_BASE_ADRS
                        /* Base adrs */
#define DC_RAM_PCI_ADRS PCI2DRAM_BASE_ADRS
                        /* RAM seen from PCI */

/* Dec 21143 (unit 1) vector and level */
#define INT_VEC_DC1    0x19
                        /* interrupt vector */
#define INT_LVL_DC1   0x19
                        /* interrupt level */
#define DC1_DATA_WIDTH NONE
                        /* all data widths */
#define IO_ADRS_DC1   LAN2_BASE_ADRS
                        /* Base adrs */
#define DC1_RAM_PCI_ADRS PCI2DRAM_BASE_ADRS
                        /* RAM seen from PCI */
```

ii) Add and advice Device and Vendor ID:

```
/* PCI Device/Vendor IDs */

#define PCI_ID_RAVEN      0x48011057 /* Id for RAVEN ASIC */
#define PCI_ID_IBC       0x056510ad /* Id for W83C553 PIB */
#define PCI_ID_UNIVERSE  0x000010e3 /* Id for Universe VME chip */
#define PCI_ID_LN_DEC21140 0x00091011 /* Id DEC chip 21140 */
#define PCI_ID_SCSI      0x00031000 /* Id for SYM53C825/75 Chips*/
#define PCI_ID_SEC_SCSI  0x00011000 /* Id for SYM53C810A Chip */
#define PCI_ID_NCR810    0x00011000 /* Id for SYM53C810A Chip */
#define PCI_ID_NCR860    0x00061000 /* Id for SYM53C860 Chip */
#define PCI_ID_NCR825    0x00031000 /* Id for SYM53C825/75 Chips*/
#define PCI_ID_LN_DEC21040 0x00021011 /* Id DEC chip 21040 */
#define PCI_ID_5434      0x00a81013 /* Id for CL-GD534 chip */
#define PCI_ID_5436      0x00ac1013 /* Id for CL-GD536 chip */
#define PCI_ID_BR_DEC21150 0x00221011 /* Id DEC 21150 PCI bridge */
#define PCI_ID_LN_DEC21143 0x00191011 /* Id DEC chip 21143 */

#define PCI_ID_PRI_LAN   PCI_ID_LN_DEC21140 /* Id for Primary LAN */
#define PCI_ID_SEC_LAN   PCI_ID_LN_DEC21143 /* Id for Secondary LAN */
```

c) File: \$(BSP)/configNet.h

Add device entry for 2nd network controller:

The BNC media can be selected explicit with setting a bit in the user flags. If the flag is set the driver will search for a BNC media or fail. If the BNC selection is not set, the driver will search for the first valid media it can find, normally this will be the TP-interface, if you are using the TPMC880-12, it will use BNC. The selection is predefined and can be changed by moving the comment brackets.

```
#define DEC_LOAD_FUNC    dec21x40EndLoad
#define DEC_LOAD_FUNC1  tpmc880EndLoad
#define DEC_BUFF_LOAN   1

/*    <devAdrs>:<PCIAdrs>:<ivec>:<ilevel>:
        <numRds>:<numTds>:<memBase>:<memSize>:<userFlags>
or    (for tpmc880)
        <devAdrs>:<PCIAdrs>:<ivec>:<ilevel>:
        <memBase>:<memSize>:<userFlags>:<align> */
#ifdef EXTENDED_VME
#define    DEC_LOAD_STRING    "0xfe020000:0x0:0x12:0x12:-1:-1:
-1:0:0x80800000"
#define    DEC_LOAD_STRING2  "0xfe060000:0x0:0x19:0x19:-
1:0:0x40400000:4"          /* Use first valid media (select for TP) */
/* #define DEC_LOAD_STRING2  "0xfe060000:0x0:0x19:0x19:-
1:0:0x40C00000:4" */      /* Use BNC interface (only -10/-12) */
#else
#define    DEC_LOAD_STRING    "0x81020000:0x80000000:0x12:0x12:-1:-1:-
1:0:0x80800000"
#define    DEC_LOAD_STRING2  "0x81060000:0x80000000:0x19:0x19:
-1:0:0x40400000:4"          /* Use first valid media (select for TP) */
/* #define DEC_LOAD_STRING2  "0x81060000:0x80000000:0x19:0x19:
-1:0:0x40C00000:4" */      /* Use BNC interface (only -10/-12) */
#endif

IMPORT END_OBJ* DEC_LOAD_FUNC (char*, void*);
IMPORT END_OBJ* DEC_LOAD_FUNC1 (char*, void*);

END_TBL_ENTRY endDevTbl [] =
{
{ 0, DEC_LOAD_FUNC, DEC_LOAD_STRING, DEC_BUFF_LOAN, NULL, FALSE},
{ 1, DEC_LOAD_FUNC1, DEC_LOAD_STRING2, DEC_BUFF_LOAN, NULL, FALSE},
{ 0, END_TBL_END, NULL, 0, NULL, FALSE},
};
#endif /* INCnetConfigh */
```

The **devAdrs**, **ivec**, **ilevel** can be read with executing the `pciHeaderShow(bus,device,0)` command on the system. The **devAdrs** is returned for base address 1, **ivec** and **ilevel** are shown in interrupt line.

d) File: \$(BSP)/sysLib.c

Modify initialization of the secondary network controller:

```
#ifndef INCLUDE_SECONDARY_ENET
/*
 * Initialize the Standard PCI Header of the dec21143 device
 * (Secondary Ethernet) if present
 */
if (pciFindDevice (    PCI_ID_SEC_LAN & 0xFFFF,
                    (PCI_ID_SEC_LAN >> 16) & 0xFFFF, 0,
                    &pciBusNo, &pciDevNo, &pciFuncNo) != ERROR)
{
    (void)pciDevConfig (pciBusNo, pciDevNo, pciFuncNo,
                      PCI_IO_LN2_ADRS,
                      NULL,
                      (PCI_CMD_MASTER_ENABLE | PCI_CMD_IO_ENABLE));
    {
        UINT temp;
        pciConfigInLong(pciBusNo, pciDevNo, pciFuncNo,0x40,&temp);
        temp &= ~(1 << 31);
        pciConfigOutLong(pciBusNo, pciDevNo, pciFuncNo,0x40,temp);
    }
}
#endif /* INCLUDE_SECONDARY_ENET */
```

2.1.3 Modify global BSP files

File: ../target/config/comps/src/net/usrNetBsd.c

Add a 2nd "dc" entry into *usrNetIfTbl*

```
#ifndef INCLUDE_DC
{ "dc", dcattach, (char*)IO_ADRS_DC, INT_VEC_DC, INT_LVL_DC,
  DC_POOL_ADRS, DC_POOL_SIZE, DC_DATA_WIDTH, DC_RAM_PCI_ADRS,
  DC_MODE },
{ "dc", dcattach, (char*)IO_ADRS_DC1, INT_VEC_DC1, INT_LVL_DC1,
  DC1_POOL_ADRS, DC1_POOL_SIZE, DC1_DATA_WIDTH, DC1_RAM_PCI_ADRS,
  DC1_MODE },
#endif /* INCLUDE_DC */
```

2.1.4 Change Project Configuration

The number of network interfaces must be increased. Start Tornado2 and open the VxWorks configuration tree of your project. Find 'network components' and open 'properties ...' with a right-click. Now change the value of 'IP_MAX_UNITS' to the number of installed network interfaces.

2.2 Install the driver to the VxWorks-System (2)

This installation shall be used for MVME5100 BSPs.

The following changes will install the driver and enable the TPMC880 on the first PMC-slot of a MVME5100 CPU as a secondary network controller.

2.2.1 Install Driver File

Copy the driver file from the MV5100 path into your project directory and include the file into the project.

The file must be copied into the project path where the objects will be placed. Then you have to add the file into the project. First select the 'builds' tab from the project, and change to your target configuration, open properties using the right click and select the 'MACROS' tab. Select 'EXTRA_MODULES' in 'Macros' and add tp880End.o in 'Value'.

2.2.2 Modify local BSP

The VxWorks documentation describes the dec21x4xEnd driver, which is configuration compatible to the tp880End driver.

Attention

After modifying a file of the BSP, the complete project must be recompiled, if just an update compile is made, the BSP-files will not be compiled!

a) File: \$(BSP)/configNet.h

Add device entry for 3rd network controller:

The BNC media can be selected explicit with setting a bit in the user flags. If the flag is set the driver will search for a BNC media or fail. If the BNC selection is not set, the driver will search for the first valid media it can find, normally this will be the TP-interface, if you are using the TPMC880-12, it will use BNC. The selection is predefined and can be changed by moving the comment brackets.

```
#ifdef INCLUDE_DEC_END
#define END_LOAD_FUNC    sysEndLoad
#endif

#define DEC_LOAD_FUNC    tpmc880EndLoad

#define END_BUFF_LOAN    1
#define DEC_BUFF_LOAN    1

#define END_LOAD_STRING ""
/* xxx_LOAD_STRING
    <devAdrs>:<PCIAdrs>:<ivec>:<ilevel>:<memBase>:<memSize>:
    <userFlags>:<align> */
*/
/* Use first valid media (select for TP) */
#define DEC_LOAD_STRING "0xfd040000:0:0x19:0x19:-1:0:0x40400000:4"

/* BNC interface (only -10/-12) */
/*#define DEC_LOAD_STRING    "0xfd040000:0:0x19:0x19:-
1:0:0x40C00000:4"*/

IMPORT END_OBJ* END_LOAD_FUNC (char*, void*);
IMPORT END_OBJ* DEC_LOAD_FUNC (char*, void*);

END_TBL_ENTRY endDevTbl [] =
{
{ 0, END_LOAD_FUNC, END_LOAD_STRING, END_BUFF_LOAN, NULL, FALSE},
#ifdef INCLUDE_SECONDARY_ENET
{ 1, END_LOAD_FUNC, END_LOAD_STRING, END_BUFF_LOAN, NULL, FALSE},
#endif
{ 2, DEC_LOAD_FUNC, DEC_LOAD_STRING, DEC_BUFF_LOAN, NULL, FALSE},
{ 0, END_TBL_END, NULL, 0, NULL, FALSE},
};
```

The **devAdrs**, **ivec**, **ilevel** can be read with executing the `pciHeaderShow(bus,device,0)` command on the system. The **devAdrs** is returned for base address 1, **ivec** and **ilevel** are shown in interrupt line.

b) File: \$(BSP)/sysLib.c

Modify initialization TPMC880 network controller:

```
if (pciFindDevice (    PCI_ID_IBC & 0xFFFF,
                    (PCI_ID_IBC >> 16) & 0xFFFF, 0,
                    &pciBusNo, &pciDevNo, &pciFuncNo) != ERROR)
{
    ...
}

/*
 * Modify PCI Header of the dec21143 device on TPMC880 if present
 */
if (pciFindDevice (    PCI_ID_LN_DEC21143 & 0xFFFF,
                    (PCI_ID_LN_DEC21143 >> 16) & 0xFFFF, 0,
                    &pciBusNo, &pciDevNo, &pciFuncNo) != ERROR)
{
    unsigned long temp;
    pciConfigInLong(pciBusNo, pciDevNo, pciFuncNo,0x40,&temp);
    temp &= ~(1 << 31);
    pciConfigOutLong(pciBusNo, pciDevNo, pciFuncNo,0x40,temp);
}
```

2.2.3 Change Project Configuration

The number of network interfaces must be increased. Start Tornado2 and open the VxWorks configuration tree of your project. Find 'network components' and open 'properties ...' with a right-click. Now change the value of 'IP_MAX_UNITS' to the maximum number of installed network interfaces.

2.3 Other BSPs

The driver has been tested on a MVME2605-900. If other BSPs are used, adaptations in the BSP configuration may be necessary and the installation example may be not the same.

The driver will not work with non PowerPC systems.

3 Starting network node

The following script can be used for a simple startup of the network node after booting. The device 'dc1' will be installed with the IP-Address of 89.0.20.111, the mask will be set to FF000000h (This should work with the MV2600 example). For a detailed description of the commands, please refer to the VxWorks manuals.

```
ipAttach 1,"dc"  
ifMaskSet "dc1",0xff000000  
ifAddrSet "dc1","89.0.20.111"
```

The device 'dc2' will be installed with the IP-Address of 89.0.20.111, the mask will be set to FF000000h (This should work with the MV5100 example). For a detailed description of the commands, please refer to the VxWorks manuals.

```
ipAttach 2,"dc"  
ifMaskSet "dc2",0xff000000  
ifAddrSet "dc2","89.0.20.111"
```

Have also a look to the following VxWorks commands, which will help you to route the IP-addresses:

```
routeAdd()  
routeDelete()  
routeShow()
```