



33 South La Patera Lane  
Santa Barbara, CA 93117-3214  
ph (805) 681-3300  
fax (805) 681-3311  
tech@motioneng.com  
www.motioneng.com

# MPI/XMP VxWorks Support Package

Revised 12Jan2004

DCR 739

## 1. Introduction

Welcome to the MPI/XMP-VxWorks BSP. The distribution was built with Tornado II and tested using Tornado II. This document provides an overview of the release and describes the new features and changes from the standard WinNT MPI/XMP software releases.

Each MPI/XMP-VxWorks distribution has a particular revision date and can only be used with a standard, WinNT MPI/XMP release of the identical revision date.

|                 | <i>Current Version</i> |
|-----------------|------------------------|
| VxWorks Version | 5.4                    |
| Tornado II      | 2.0                    |

### 1.1 System Requirements

#### *1.1.1 Host System Requirements*

The MPI/XMP-VxWorks release is designed to operate with either a Win32 or SPARC station host.. The Win32 XMP/MPI software release supports client/server operation via ethernet. Be sure to install the MPI/XMP Win32 Software on the host system.

Tornado version 1.0.1 or Tornado 2.0 for x86 targets must be installed on the host system. Tornado 1.0.1 must include patches and SENS version 1.0 and be installed from the latest revision ZC-07 of the Tornado CD, Part. No. TDK-12217-ZC-07. Otherwise, the files required for PCI initialization and TCP/IP might be missing.

### 1.1.2 Target System Requirements

This MPI/XMP-VxWorks release is built for the following BSP:

- pcPentiumPro BSP
- VxWorks version 5.4 BSP version 1.2/0 for Celeron CPU
- Modified by Motion Engineering, Inc. for X86 architectures.

## 1.2 Installing the Distribution on the Host System

The MPI/XMP-VxWorks installation consists of an InstallShield base distribution. It assumes a default installation directory of **C:\MEI** unless otherwise specified during installation. The MPI/XMP-VxWorks installation is designed to install on top of the standard Win32 MPI/XMP release. Both releases must be installed entirely for the Makefiles to run correctly.

All VxWorks specific files are installed into **VxWorks\x86** subdirectories. Key components of the distribution are:

- VxWorks BSP source files, installed in the **\xmp\mpi\VxWorks\x86** subdirectory.
- The release and debug versions of the MPI/XMP-VxWorks libraries, installed in the **\xmp\lib\VxWorks\x86** subdirectory.
- GNU Makefiles for various sample applications and utilities.

## 2. Building the Board Support Package

After the release has been installed, modifications to the VxWorks system files are necessary to provide support for the XMP-Series Motion Controller. Please follow the steps below to include XMP support in your BSP:

### 1. Modify **sysLib.c**, found in the **icPentium** directory.

**NOTE:** An example source file of **sysLib.c** is included in the **\xmp\mpi\VxWorks\x86** subdirectory. While this can be used as a reference, do not copy this file over your original **sysLib.c**, as the example may not include all of your system specific settings.

- Add this at the end of the **#INCLUDE** section (near line #138)

```
#ifdef      INCLUDE_PCI
#include    "sysXmp.h"
#endif     /* INCLUDE_PCI */
```

- Search for this line:

```
PHYS_MEM_DESC sysPhysMemDesc [] =
```

- Add the following code to the end of the MMU list, before any DUMMY\_MMU\_ENTRY, if one exists. It is possible that there are multiple MMU lists for different cases. Be sure that the following code is included for every ifdef situation:

```
#ifdef INCLUDE_PCI
{
(void *) XMP0_MEMBASE0,
(void *) XMP0_MEMBASE0,
XMP0_MEMSIZE0,
VM_STATE_MASK_VALID | VM_STATE_MASK_WRITABLE | VM_STATE_MASK_CACHEABLE,
VM_STATE_VALID | VM_STATE_WRITABLE | VM_STATE_CACHEABLE_NOT
};
#endif /* INCLUDE_PCI */
```

- After the line:

```
# endif /* INCLUDE_SHOW_ROUTINES */
```

add the following line, near line 358:

```
# include "sysXmp.c" /* XMP support */
```

- At the end of the function:

```
void sysHwInit2 (void)
append:
#ifdef INCLUDE_PCI
/* connect XMP interrupt */
sysXmpHwInit2 ();
#endif /* INCLUDE_PCI */
```

...in the function

```
char *sysPhysMemTop (void),
```

replace the following code segment, near line 629,

```
#ifndef INCLUDE_PCI
{
GDT *pGdt = pSysGdt;
int limit = (((int)memTop) / 0x1000 - 1);

for (ix=1; ix < GDT_ENTRIES; ix++)
{
pGdt++;
pGdt->limit00 = limit & 0x0fff;
pGdt->limit01 = ((limit & 0xf0000) >> 16) | (pGdt->limit01 & 0xf0);
}
}
#endif /* INCLUDE_PCI */
```

...with the following segment:

```
#ifdef      INCLUDE_PCI
            sysXmplnit ();      /* initialize XMP data structure */
#else
{
    GDT *pGdt = pSysGdt;
    int limit = (((int)memTop) / 0x1000 - 1);

    for (ix=1; ix < GDT_ENTRIES; ix++)
    {
        pGdt++;
        pGdt->limit00 = limit & 0x0fff;
        pGdt->limit01 = ((limit & 0xf0000) >> 16) | (pGdt->limit01 & 0xf0);
    }
}
#endif      /* INCLUDE_PCI */
```

**2. Copy *sysXmp.c* and *sysXmp.h* from the `\xmp\mpi\VxWorks\x86` subdirectory to the working directory or modify the source path so they can be found.**

**3. Recompile the *bootrom\_uncmp* and *vxWorks* image to include XMP support. Please refer to your BSP or Tornado documentation for instructions on compiling the boot images.**

## 3. General Operation

### 3.1 Boot the Target System

The first step is to boot the target system using the newly compiled VxWorks and bootrom image containing XMP support. Verify that the target has successfully attached to the host computer and no warning or error messages were logged.

### 3.2 Loading the library object modules

Once the target is booted, is to load the MEI/XMP library modules *mpiXmp.o* and *appUtil.o* to the target:

```
ld < /mei/xmp/lib/VxWorks/x86/debug/mpiXmp.o  
ld < /mei/xmp/lib/VxWorks/x86/debug/appUtil.o
```

These object modules have been precompiled for a VxWorks target and come in a Release or Debug format. The *mpiXmp.o* must be loaded first as it contains definitions for *appUtil.o*.

### 3.3 Host/Client Communication between Win32 and VxWorks Applications

The target system is accessible from a Win32 host through TCP/IP via the server application. The *server.o* program is loaded on the target and then executed by typing '*serverMain*' from the shell or console. This process must be active to receive any commands from a Win32 host. Press any key on the target to exit the server program. On the host side, all Win32 utilities and sample programs must be run with the *-server <target>* flag. Please refer to section 3.4 for each utilities usage.

### 3.4 Running XMP support utilities under VxWorks

Pre-compiled utilities are included in the VxWorks release located at **C:\MEI\XMP\BIN**. These utilities support two modes of communication - they can be loaded and run from the target system or across ethernet from a Win32 host with the *-server* flag:

#### 3.4.1 *version* program

- Load the *version.o* object onto the target system. Type:  
**ver**

**NOTE:** due to a VxWorks module name conflict the *version.o* main routine does not match the module name.

### 3.4.2 *reset* program

- Load the reset.o object onto the target system. Type:  
**reset**
- Running via TCP/IP from a Win32 host, type:  
**reset –server <target>**

### 3.4.3 *flash* program

- Load the flash.o object onto the target system. For example, type:  
**flash “C:/MEI/XMP/bin/xmpNNNxN.bin”**
- Running via TCP/IP from a Win32 host, type:  
**flash –server <target> “C:/MEI/XMP/bin/xmpNNNxN.bin”**

### 3.4.4 *VM3* program

- Operation not supported on the target system or via a sparc station.
- Running via TCP/IP from a Win32 host, type:  
**Vm3 –server <target>**

### 3.4.5 *Motion Console*

- Operation not supported on the target system or via a sparc station.
- Running via TCP/IP from a Win32 host, please refer to the *Motion Console and Motion Scope Users Manual* for operation instructions.

### 3.4.6 *Motion Scope*

- Operation not supported on the target system or via a sparc station.
- Running via TCP/IP from a Win32 host, please refer to the *Motion Console and Motion Scope Users Manual* for operation instructions.

### 3.5 MPI/XMP Sample Applications

This release installs a *makfile* for building the sample applications under *Tornado*. The *makefile* provides an interface to build all the sample apps from the command line using the *make* utility.

## 4. Known Bugs and Issues

There are no known bugs or open issues.

*Copyright © 2004 Motion Engineering, Inc. All rights reserved.  
All trademarks are the property of their respective owners.*