MANUAL ID:	**08205**
REV. INDEX:	**00000**
DATE:	**July / 2008**
SUBJECT	**Boot vxWorks from User Flash on PowerEngine7 and PowerNode3 board using TFFS under vxworks 6.2**
Thema:	**This FAQ describes the method to boot vxWorks from PowerEngine7 and PowerNode3 User Flash using the TFFS True Flash File System tools from WindRiver. TrueFFS for Tornado provides a block device interface for flash memory devices**
Report by/ Bericht von:	Technical Support, Kontron Modular Computers Toulon

Question :

How does one boot vxWorks from PowerEngine7 and PowerNode3 User Flash using TrueFFS ?

Answer :

Here is a method which explains how:

Update vxWorks bootrom to include TFFS
Copy a vxWorks into User Flash using TFFS
Configure Boot parameters and boot with TFFS

Software configuration:

Windows or Linux Workbanch 2.4.

BSP ID08088 under vxworks6.2 for PowerEngine7 and PowerNode3 + patch bsp07088-pe7&PN3-Vx6.2-2 id08088 or latest

Hardware:

PowerEngine7 or PowerNode3 board.

1) Update vxWorks bootrom to include TFFS.

First create a user.h file that contains:

```
/* Flash*/
#undef INCLUDE_USER_FLASH
/*tffs*/
#define INCLUDE_TFFS
#define INCLUDE_TFFS_SHOW
#define INCLUDE_TFFS_MOUNT

#define INCLUDE_ERF
```

```
#define INCLUDE_XBD
#define INCLUDE_XBD_BLK_DEV
#define INCLUDE_XBD_RAMDRV
#define INCLUDE_DEVICE_MANAGER
#define INCLUDE_FS_MONITOR
#define INCLUDE_RAW_FS
#define INCLUDE_FS_EVENT_UTIL

#define INCLUDE_DOSFS
#define INCLUDE_DOSFS_MAIN
#define INCLUDE_DOSFS_DIR_VFAT
#define INCLUDE_DOSFS_DIR_FIXED
#define INCLUDE_DOSFS_FAT
#define INCLUDE_DOSFS_CHKDSK
#define INCLUDE_DOSFS_FMT
#define INCLUDE_DOSFS_CACHE
#undef DOSFS_DEFAULT_CACHE_SIZE
#define DOSFS_DEFAULT_CACHE_SIZE 0x20000 /*128K*/
```

Add the following line in the config.h file:

```
#include "user.h"
```

Build bootrom.bin (vxworks 6.2) file in the corresponding BSP directory and write the VMPC system Flash with this new bootrom file.

The method to update a bootrom.bin on VMPCx is explained in the vxWorks BSP documentation, in "Flashing the Boot ROM Using VMPCBug firmware" section.

2) Copy a vxWorks on User Flash using TFFS.

To copy a vxWorks on User Flash using TFFS it is necessary to create a vxWorks kernel with TFFS included and to boot it over the network. In this example, the symbol table is included in the built vxWorks image.

2.1) Compilation by manual command line:

In the vxWorks BSP directory, modify the user.h file and add :

```
#define INCLUDE_STDIO
#define INCLUDE_STDIO_SHOW

#define INCLUDE_SHELL
#define INCLUDE_LOADER



#define INCLUDE_WATCHDOGS_SHOW
```

```
#define INCLUDE_SEM_SHOW
#define INCLUDE_SHOW_ROUTINES


/* Symboles table*/


#define INCLUDE_SYM_TBL
#define INCLUDE_SYM_TBL_INIT
#define INCLUDE_SYM_TBL_SHOW
#define INCLUDE_HOST_TBL
#define INCLUDE_STAT_SYM_TBL
/*for vxWorks.st build*/
#undef LINK_SYMBOLE_TABLE
#ifdef LINK_SYMBOLE_TABLE
#undef INCLUDE_NET_SYM_TBL
#define INCLUDE_STANDALONE_SYM_TBL
#define STANDALONE_NET
#else
#define INCLUDE_NET_SYM_TBL
#endif
```

Rebuilt vxWorks:

make clean

make vxWorks

then boot your target with this new kernel.

2.2) Compilation with Tornado tools

Make sure that the following components are included in the environment:

INCLUDE_TFFS,

and optionally

INCLUDE_STANDALONE_SYM_TBL,

Make a Build 'vxWorks'. The INCLUDE_STANDALONE_SYM_TBL can be set or not depending if you want a separate symbol table from the kernel.

Then boot your target with this new kernel.

2.3) Copy the vxWorks kernel on User Flash by TFFS

Under the vxWorks prompt, type the following commands to create a dos file system on the userFlash, using the TFFS layer :

```
-> sysFlashErase
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++
value = 0 = 0x0
-> sysTffsFormat
value = 0 = 0x0
->
-> usrTffsConfig(0,0,"/tffs")
value = 0 = 0x0
-> copy ("vxWorks","/tffs/vxWorks")
value = 0 = 0x0
-> copy("vxWorks.sym","/tffs/vxWorks.sym")
value = 0 = 0x0
```

3) Configure Boot parameters and boot with TFFS

Modify the vxWorks boot parameters to have:

```
[VxWorks Boot]: c

'.' = clear field; '-' = go to previous field; ^D = quit

boot device : gei1 tffs=0,0
processor number : 0
host name : pcxpsvi
file name : vxWorks /tffs/vxWorks
inet on ethernet (e) : 192.93.161.100
inet on backplane (b): 91.0.0.1
host inet (h) : 192.93.161.143
gateway inet (g) :
user (u) : vxworks
ftp password (pw) (blank = use rsh): azerty
flags (f) : 0x0
target name (tn) : vxworks1
startup script (s) :
other (o) : gei1
```

Notes :

- the boot device parameters "0,0" must be present but will not be used. It is there to keep the same ATA and SCSI boot format.

- this example also initializes giga ethernet network and load a start up script

[VxWorks Boot]: @

boot device : tffs=0,0
unit number : 0
processor number : 0
host name : pcxpsvi
file name : /tffs/vxWorks
inet on ethernet (e) : 192.93.161.100
inet on backplane (b): 91.0.0.1
host inet (h) : 192.93.161.143
user (u) : vxworks
ftp password (pw) : azerty
flags (f) : 0x0
target name (tn) : vxworks1
other (o) : gei1

Attaching to TFFS... done.
Loading /tffs/vxWorks...2388208
Starting at 0x110000...

Set CPU 0 state 0x56585741 VXWA
Set CPU 0 state 0x56585742 VXWB
Set CPU 0 state 0x56585743 VXWC
Set CPU 0 state 0x56585744 VXWD
Set CPU 0 state 0x56585745 VXWE
Set CPU 0 state 0x56585746 VXWF
Attaching interface lo0... done
Attaching to SM net with memory anchor at 0x4100...
SM address: 91.0.0.1
Attached IPv4 interface to gei unit 1
Attached IPv4 interface to sm unit 0
Loading symbol table from /tffs/vxWorks.sym ...done
Set CPU 0 state 0x56585747 VXWG
ftpXfer: Loading PN3_05302_SHML3/P1/default/vxWorks ...
sysSlaveLoad: ftpXfer failed 0x0
File:sysSlave.c Line:354
Loading kernel failed (Slave CPU)
Set CPU 0 state 0x56585753 VXWS

]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]] (R)
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]] Development System
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]] VxWorks 6.2
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]] KERNEL: WIND version 2.8
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]] Copyright Wind River Systems, Inc., 1984-2005

CPU: VEGA-DUAL. Processor #0.
Memory Size: 0x1c00000. BSP version 2.0/07088.
Created: Mar 27 2008, 18:22:27
ED&R Policy Mode: deployed
WDB Comm Type: WDB_COMM_NETWORK
WDB: Ready.

Note: Due to the User Flash structure copy and delete files time can be very variable (from few seconds to several minutes). Some of them will necessitate many sector erase (256K erase) and rewritten depending on what has already been stored on User Flash. DOS/FS is not adapted to optimize copy on Flash memory.

Copyright

Copyright © 2008 Kontron Modular Computers S.A.S.. All rights reserved. This manual may not be copied, photocopied, reproduced, translated or converted to any electronic or machine-readable form in whole or in part without prior written approval of Kontron Modular Computers S.A.S.